

**PENGEMBANGAN APLIKASI TATA TERTIB PELANGGARAN
(SiTalang) SEKOLAH BERBASIS WEBSITE MENGGUNAKAN
SDLC WATERFALL PADA SMKN 1 PASURUAN**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Pendidikan

Disusun oleh:

Muhammad Sodik

NIM: 175150601111016



**PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2021**



PENGESAHAN

PENGEMBANGAN APLIKASI TATA TERTIB PELANGGARAN (SiTalang) SEKOLAH
BERBASIS *WEBSITE* MENGGUNAKAN SDLC *WATERFALL* PADA SMKN 1 PASURUAN

SKRIPSI

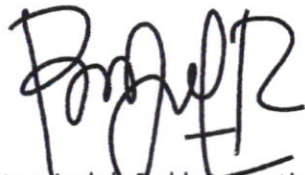
Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Pendidikan

Disusun Oleh :
Muhammad Sodik
NIM: 175150601111016


Skripsi ini telah diuji dan dinyatakan lulus pada
02 Juli 2021
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2




Retno Indah Rokhmawati, S.Pd., M.Pd.
NIK. 2016099009172001



Tri Afirianto S.T., M.T.
NIK. 2013098512131001

Mengetahui
Ketua Jurusan **Sistem Informasi**



Issa Arwani, S.Kom., M.Sc.
NIK: 198309222012121003

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain. Kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 02 Juli 2021



Muhammad Sodik

NIM: 175150601111016

PRAKATA

Segala puji bagi Tuhan Yang Maha Esa atas limpahan rahmat dan hidayah yang diberikan sehingga penulis dapat menyelesaikan tugas akhir atau skripsi yang berjudul "PENGEMBANGAN APLIKASI TATA TERTIB PELANGGARAN (SiTalang) SEKOLAH BERBASIS WEBSITE MENGGUNAKAN SDLC WATERFALL PADA SMKN 1 PASURUAN". Penulis menyadari bahwa penulisan skripsi ini dapat diselesaikan dengan bantuan dan dukungan dari pihak terkait dan orang-orang terdekat, sehingga saya ingin menyampaikan rasa terima kasih saya kepada:

1. Ibu saya Sudeyah, Ayah saya Untung Suseno, serta seluruh keluarga saya yang telah memberikan dukungan secara material maupun doa serta senantiasa memberikan semangat agar dapat menyelesaikan penulisan skripsi ini.
2. Ibu Retno Indah Rokhmawati, S.Pd, M.Pd. selaku Dosen Pembimbing satu yang telah membimbing dalam menyelesaikan penulisan skripsi ini.
3. Bapak Tri Afirianto, S.T., M.T. selaku Dosen Pembimbing dua yang telah membimbing dalam menyelesaikan penulisan skripsi ini.
4. Bapak Admaja Dwi Herlambang, S.Pd, M.Pd selaku Ketua Program Studi Pendidikan Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Brawijaya yang telah mengayomi seluruh mahasiswa Pendidikan Teknologi Informasi.
5. Pihak SMK Negeri 1 Pasuruan yang telah membantu dalam penelitian.
6. Saudara Tio Rizky Bachtiar yang telah membantu dalam penyelesaian program dalam penulisan skripsi ini.
7. Seluruh teman-teman mahasiswa Pendidikan Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Brawijaya yang telah menemani perjuangan selama masa perkuliahan.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran maupun kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi seluruh pihak yang menggunakannya.

Malang, 02 Juli 2021

Muhammad Sodik

muesodiq@student.ub.ac.id

ABSTRAK

Muhammad Sodik, Pengembangan Aplikasi Tata Tertib Pelanggaran (SiTalang) Sekolah Berbasis Website Menggunakan SDLC Waterfall pada SMKN 1 Pasuruan**Pembimbing: Retno Indah Rokhmawati, S.Pd., M.Pd. dan Tri Afirianto, S.T., M.T.**

Sistem poin pelanggaran merupakan salah satu solusi yang digunakan oleh sekolah dalam menangani pelanggaran siswa di sekolah, salah satunya SMK Negeri 1 Pasuruan. Akan tetapi, dalam penerapan sistem poin pelanggaran di SMK Negeri 1 Pasuruan masih menggunakan media kertas sehingga penerapan sistem poin pelanggaran menjadi kurang efektif. Tujuan dari penelitian ini yaitu untuk mempermudah penerapan sistem poin di SMK Negeri 1 Pasuruan dengan dikembangkannya aplikasi pengolah poin pelanggaran bernama SiTalang (Sistem Tata Tertib Pelanggaran). Proses pengembangan aplikasi SiTalang yaitu menggunakan SDLC (*System Development Life Cycle*) *Waterfall Model* dengan tahapan seperti analisis kebutuhan dengan proses pengambilan data menggunakan metode wawancara, perancangan sistem, implementasi, dan pengujian. Penelitian ini menghasilkan sebuah sistem dengan dua pengguna yaitu guru dan siswa. Fungsional pada sistem yang dikembangkan menghasilkan beberapa fitur seperti masuk, reset *password*, ubah *password*, lihat daftar siswa, lihat pelanggaran siswa, tambah poin, edit poin, hapus poin, cetak poin, tambah akun siswa, tambah data siswa, keluar, dan lihat pelanggaran. Proses pengujian dari sistem yang dikembangkan menggunakan tiga metode pengujian yaitu *unit testing*, *validation testing*, dan *user acceptance testing* (UAT). Hasil dari *unit testing* terhadap 3 fungsi utama sistem mendapatkan total 8 kasus uji. Hasil *validation testing* yang menguji fungsional sistem menunjukkan bahwa seluruh fungsional sistem telah berjalan sesuai dengan fungsinya. Hasil pengujian *user acceptance testing* yang menguji penerimaan pengguna terhadap sistem juga menunjukkan nilai rata-rata penerimaan sebesar 98,68% yang didapat dari tiga responden guru dan 100% yang didapat dari empat responden siswa dengan kesimpulan yaitu sistem dapat diterima oleh pengguna. Hasil pengujian *user acceptance testing* juga menghasilkan *feedback* secara kualitatif untuk memberikan masukan terhadap pengembangan sistem sebagai saran pada penelitian selanjutnya.

Kata kunci: Sistem Poin, *Waterfall Model*, *Validation Testing*, *User Acceptance Testing*, Sekolah

ABSTRACT

Muhammad Sodik, Application Development Code of Conduct for Violations (SiTalang) School with Website Based Using SDLC Waterfall on SMKN 1 Pasuruan

Supervisors: Retno Indah Rokhmawati, S.Pd., M.Pd. and Tri Afirianto, S.T., M.T.

The violation point system is one of the solutions used by schools in dealing with student violations in schools, one of which is SMK Negeri 1 Pasuruan. However, in implementing the violation point system at SMK Negeri 1 Pasuruan, it still written using paper media, so the application of the violation point system is less effective.

The purpose of this study is to facilitate the application of the point system at SMK Negeri 1 Pasuruan by developing a website-based violation point processing application called SiTalang (Sistem Tata Tertib Pelanggaran). The process of developing the SiTalang application is using the SDLC (System Development Life Cycle) Waterfall Model with stages such as needs analysis with the data collection process using the interview method, system planning, implementation, and testing. This study produces a system with two users, teachers and students.

Functional in the system developed produce several features such as logging in, resetting passwords, changing passwords, viewing student lists, viewing student violations, adding points, editing points, removing points, printing points, adding student accounts, adding student data, leaving, and see violation. The testing process of the system developed uses three testing methods, namely unit testing, validation testing, and user acceptance testing (UAT). The results of unit testing on the 3 main functions of the system get a total of 8 test cases. The results of validation testing that tests the system's functionality show that all functional systems have been running according to their functions. The results of user acceptance testing which tests user acceptance of the system also show an average acceptance value of 98.43% obtained from three teacher respondents and 100% obtained from four student respondents with the conclusion that the system can be accepted by users. The results of user acceptance testing also produce qualitative feedback to provide input on system development as suggestions for further research.

Keywords: Point System, Waterfall Model, Validation Testing, User Acceptance Testing, School

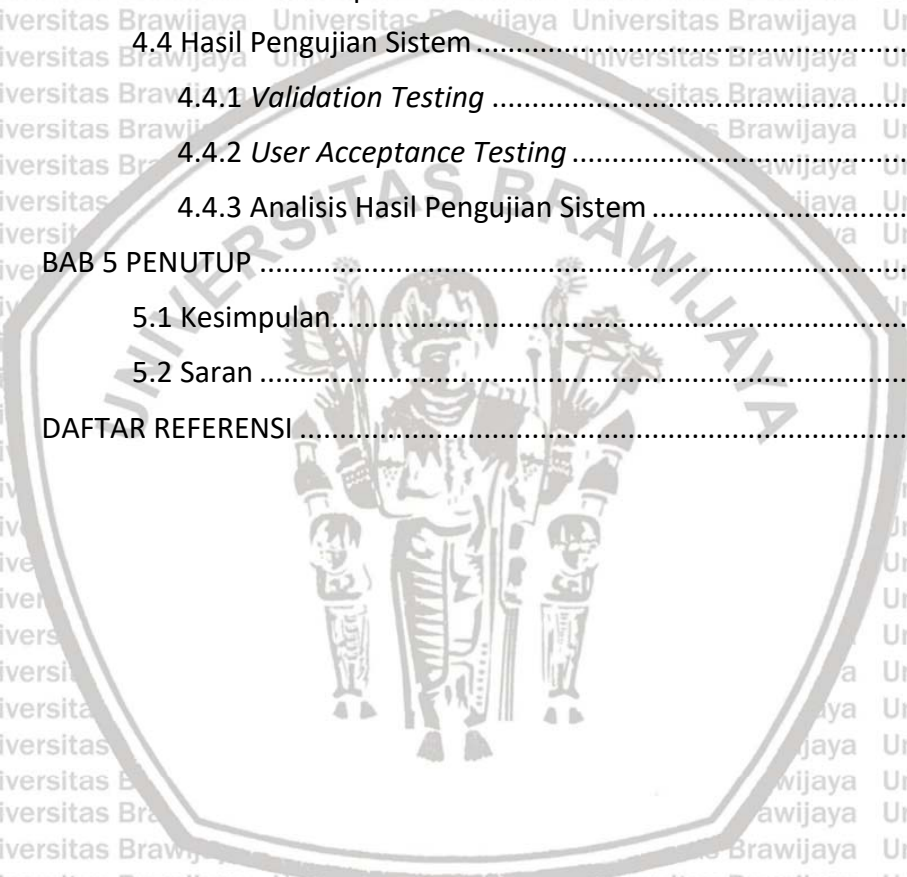
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Penelitian Relevan	4
2.2 Profil Sekolah	5
2.2.1 Profil SMK Negeri 1 Pasuruan	5
2.2.2 Visi Misi SMK Negeri 1 Pasuruan	5
2.3 Poin Pelanggaran Siswa	5
2.4 Sistem Informasi	7
2.5 <i>Website</i>	7
2.6 <i>Waterfall Model</i>	7
2.6.1 Analisis Kebutuhan/ <i>Requirement Definition</i>	8
2.6.2 Desain Sistem dan Perangkat Lunak / <i>System and Software Design</i>	12



2.6.3 Pengkodean dan Pengujian Unit/ <i>Implementation and Unit Testing</i>	19
2.6.4 Pengujian/ <i>Integration and System Testing</i>	22
2.6.5 Pemeliharaan/ <i>Operation and Maintenance</i>	26
2.7 <i>Database</i>	26
2.8 <i>Framework</i> Laravel.....	26
2.9 XAMPP.....	29
2.10 Bootstrap.....	29
BAB 3 METODOLOGI PENELITIAN	30
3.1 Alur Penelitian.....	30
3.2 Studi Literatur.....	30
3.3 Identifikasi Masalah.....	31
3.4 Analisis Kebutuhan.....	31
3.5 Perancangan Sistem.....	31
3.6 Implementasi Sistem dan Pengujian Unit.....	31
3.7 Pengujian Sistem.....	32
3.8 Penutup.....	32
BAB 4 HASIL DAN PEMBAHASAN	33
4.1 Hasil Analisis Kebutuhan.....	33
4.1.1 Analisis Permasalahan.....	33
4.1.2 Identifikasi Kebutuhan.....	33
4.1.3 Identifikasi Aktor.....	34
4.1.4 Kebutuhan Fungsional dan Non-Fungsional.....	34
4.1.5 <i>Use Case Diagram</i>	40
4.1.6 <i>Use Case Scenario</i>	41
4.1.7 <i>Activity Diagram</i>	53
4.2 Hasil Perancangan Sistem.....	67
4.2.1 <i>Sequence Diagram</i>	67
4.2.2 <i>Entity Relationship Diagram (ERD)</i>	84
4.2.3 <i>Class Diagram</i>	84
4.2.4 <i>Pseudocode</i>	85
4.2.5 Desain <i>Wireframe</i>	95

4.3 Hasil Implementasi Sistem.....	110
4.3.1 Hasil Implementasi <i>Physical Data Model</i> (PDM)	110
4.3.2 Batasan Implementasi Sistem.....	110
4.3.3 Spesifikasi Implementasi Sistem.....	111
4.3.4 Hasil Implementasi <i>Model</i>	111
4.3.5 Hasil Implementasi Controller.....	116
4.3.6 <i>Unit Testing</i>	122
4.3.7 Hasil Implementasi Antarmuka.....	127
4.4 Hasil Pengujian Sistem	136
4.4.1 <i>Validation Testing</i>	136
4.4.2 <i>User Acceptance Testing</i>	160
4.4.3 Analisis Hasil Pengujian Sistem	165
BAB 5 PENUTUP	167
5.1 Kesimpulan.....	167
5.2 Saran	167
DAFTAR REFERENSI.....	168



DAFTAR TABEL

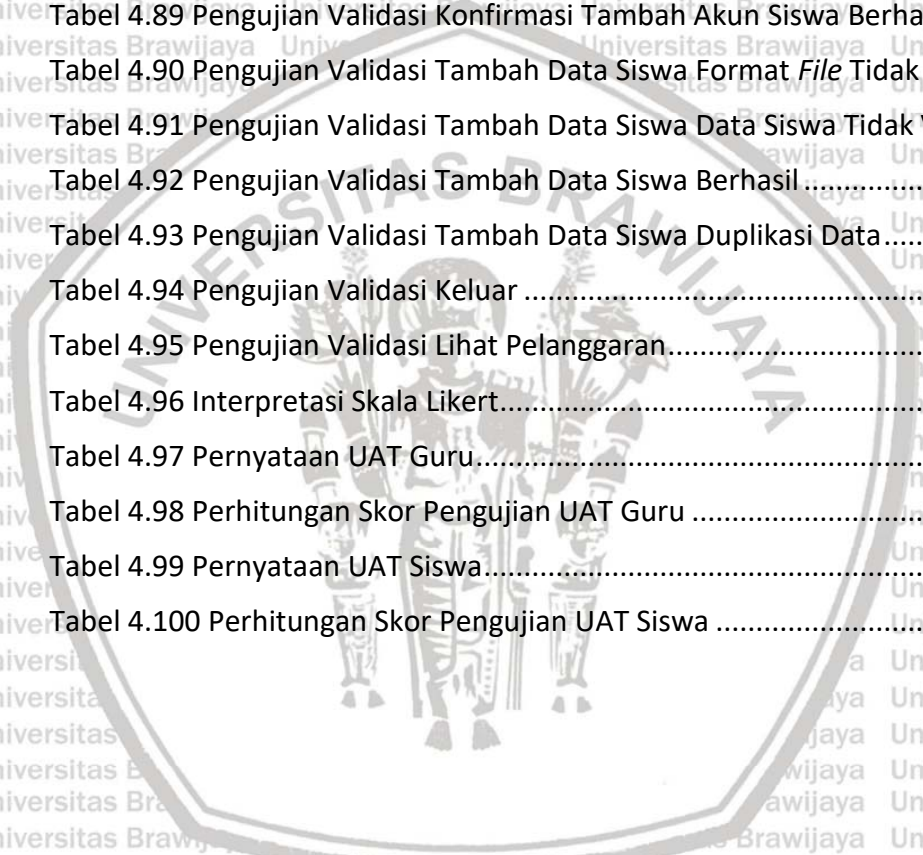
Tabel 2.1 Simbol Diagram <i>Use Case</i>	9
Tabel 2.2 Format Tabel <i>Use Case Scenario</i>	10
Tabel 2.3 Simbol <i>Activity Diagram</i>	11
Tabel 2.4 Simbol <i>Sequence Diagram</i>	13
Tabel 2.5 Simbol <i>Entity Relational Database (ERD)</i>	15
Tabel 2.6 Simbol <i>Class Diagram</i>	17
Tabel 2.7 Contoh Penulisan <i>Pseudocode</i>	18
Tabel 2.8 Contoh Tabel Kasus Uji.....	21
Tabel 2.9 Contoh Tabel <i>Validation Testing</i>	22
Tabel 2.10 Contoh Kuesioner <i>User Acceptance Testing (UAT)</i>	24
Tabel 2.11 Interpretasi Skala Likert.....	24
Tabel 2.12 Format Kuesioner Skala Likert.....	25
Tabel 2.13 Contoh Penulisan Kode Program <i>Framework Laravel</i>	28
Tabel 4.1 Identifikasi Kebutuhan.....	33
Tabel 4.2 Identifikasi Aktor.....	34
Tabel 4.3 Identifikasi Kebutuhan Fungsional SiTalang.....	34
Tabel 4.4 Penjabaran Kebutuhan Fungsional Pengguna.....	36
Tabel 4.5 Penjabaran Kebutuhan Fungsional Guru.....	36
Tabel 4.6 Penjabaran Kebutuhan Fungsional Siswa.....	38
Tabel 4.7 Penjabaran Kebutuhan Non-Fungsional SiTalang.....	39
Tabel 4.8 <i>Use Case Scenario</i> Masuk.....	41
Tabel 4.9 <i>Use Case Scenario</i> Reset <i>Password</i>	42
Tabel 4.10 <i>Use Case Scenario</i> Ubah <i>Password</i>	43
Tabel 4.11 <i>Use Case Scenario</i> Lihat Daftar Siswa.....	44
Tabel 4.12 <i>Use Case Scenario</i> Lihat Pelanggaran Siswa.....	45
Tabel 4.13 <i>Use Case Scenario</i> Tambah Poin Pelanggaran.....	46
Tabel 4.14 <i>Use Case Scenario</i> Edit Poin Pelanggaran.....	46
Tabel 4.15 <i>Use Case Scenario</i> Hapus Poin Pelanggaran.....	48
Tabel 4.16 <i>Use Case Scenario</i> Cetak Poin Pelanggaran.....	48
Tabel 4.17 <i>Use Case Scenario</i> Tambah Motivasi.....	49



Tabel 4.18 <i>Use Case Scenario</i> Tambah Akun Siswa	50
Tabel 4.19 <i>Use Case Scenario</i> Tambah Data Siswa	51
Tabel 4.20 <i>Use Case Scenario</i> Keluar	52
Tabel 4.21 <i>Use Case Scenario</i> Lihat Pelanggaran	52
Tabel 4.22 <i>Pseudocode</i> Masuk	86
Tabel 4.23 <i>Pseudocode</i> Verifikasi Email	86
Tabel 4.24 <i>Pseudocode</i> Verifikasi Kode	87
Tabel 4.25 <i>Pseudocode</i> Reset Password	87
Tabel 4.26 <i>Pseudocode</i> Ubah Password	88
Tabel 4.27 <i>Pseudocode</i> Lihat Daftar Siswa	88
Tabel 4.28 <i>Pseudocode</i> Profil Siswa	89
Tabel 4.29 <i>Pseudocode</i> Pelanggaran Siswa	89
Tabel 4.30 <i>Pseudocode</i> Tambah Poin Pelanggaran	90
Tabel 4.31 <i>Pseudocode</i> Edit Poin Pelanggaran	90
Tabel 4.32 <i>Pseudocode</i> Hapus Poin Pelanggaran	90
Tabel 4.33 <i>Pseudocode</i> Cetak Poin Pelanggaran	91
Tabel 4.34 <i>Pseudocode</i> Tambah Motivasi	91
Tabel 4.35 <i>Pseudocode</i> Konfirmasi Data Siswa	92
Tabel 4.36 <i>Pseudocode</i> Konfirmasi Tambah Akun Siswa	92
Tabel 4.37 <i>Pseudocode</i> Tambah Data Siswa	93
Tabel 4.38 <i>Pseudocode</i> Keluar	93
Tabel 4.39 <i>Pseudocode</i> Lihat Profil Siswa	94
Tabel 4.40 <i>Pseudocode</i> Lihat Pelanggaran Siswa	94
Tabel 4.41 Spesifikasi Perangkat Keras	111
Tabel 4.42 Spesifikasi Perangkat Lunak	111
Tabel 4.43 <i>Source Code Model</i> Pengguna.php	112
Tabel 4.44 <i>Source Code Model</i> Siswa	112
Tabel 4.45 <i>Source Code Model</i> Poin	113
Tabel 4.46 <i>Source Code Model</i> Status	113
Tabel 4.47 <i>Source Code Model</i> Nilai	114
Tabel 4.48 <i>Source Code Model</i> Motivasi	114
Tabel 4.49 <i>Source Code Model</i> Kelas	115

Tabel 4.50 <i>Source Code Model</i> KelasGrup	115
Tabel 4.51 <i>Source Code Model</i> Kategori.....	115
Tabel 4.52 <i>Source Code Controller</i> AuthController.....	116
Tabel 4.53 <i>Source Code Controller</i> SiswaController.....	119
Tabel 4.54 <i>Source Code Controller</i> PoinController.....	121
Tabel 4.55 Kasus Uji Fungsi Tambah Poin Pelanggaran.....	124
Tabel 4.56 Kasus Uji Fungsi Edit Poin Pelanggaran.....	125
Tabel 4.57 Kasus Uji Fungsi Tambah Poin Pelanggaran.....	127
Tabel 4.58 Pengujian Validasi Fungsional Masuk Gagal	136
Tabel 4.59 Pengujian Validasi Fungsional Masuk Berhasil.....	137
Tabel 4.60 Pengujian Validasi Verifikasi <i>Email</i> Salah	138
Tabel 4.61 Pengujian Validasi Verifikasi <i>Email</i> Benar	139
Tabel 4.62 Pengujian Validasi Konfirmasi Kode Salah	139
Tabel 4.63 Pengujian Validasi Konfirmasi Kode Benar.....	140
Tabel 4.64 Pengujian Validasi Reset <i>Password</i> Tidak Valid	140
Tabel 4.65 Pengujian Validasi Reset <i>Password</i> Tidak Sesuai	141
Tabel 4.66 Pengujian Validasi Reset <i>Password</i> 6 Karakter	141
Tabel 4.67 Pengujian Validasi Reset <i>Password</i> Berhasil	142
Tabel 4.68 Pengujian Validasi Ubah <i>Password</i> Tidak Valid.....	143
Tabel 4.69 Pengujian Validasi Ubah <i>Password</i> Tidak Sesuai.....	144
Tabel 4.70 Pengujian Validasi Ubah <i>Password</i> 6 Karakter.....	144
Tabel 4.71 Pengujian Validasi Ubah <i>Password</i> Berhasil.....	145
Tabel 4.72 Pengujian Validasi Lihat Daftar Siswa	146
Tabel 4.73 Pengujian Validasi Lihat Pelanggaran Siswa.....	146
Tabel 4.74 Pengujian Validasi Tambah Poin Tidak Valid.....	147
Tabel 4.75 Pengujian Validasi Tambah Poin Kosong.....	148
Tabel 4.76 Pengujian Validasi Tambah Poin Berhasil	148
Tabel 4.77 Pengujian Validasi Edit Poin Tidak Valid.....	149
Tabel 4.78 Pengujian Validasi Edit Poin Kosong	150
Tabel 4.79 Pengujian Validasi Edit Poin Berhasil	151
Tabel 4.80 Pengujian Validasi Hapus Poin Gagal	152
Tabel 4.81 Pengujian Validasi Hapus Poin Berhasil	153

Tabel 4.82 Pengujian Validasi Cetak Poin Pelanggaran.....	153
Tabel 4.83 Pengujian Validasi Tambah Motivasi Gagal.....	154
Tabel 4.84 Pengujian Validasi Tambah Motivasi Berhasil.....	154
Tabel 4.85 Pengujian Validasi Tambah Akun Siswa <i>Username</i> atau <i>Email</i> Tidak Valid.....	155
Tabel 4.86 Pengujian Validasi Tambah Akun Siswa <i>Password</i> Tidak Valid.....	155
Tabel 4.87 Pengujian Validasi Tambah Akun Siswa Informasi Valid.....	156
Tabel 4.88 Pengujian Validasi Konfirmasi Tambah Akun Siswa Gagal.....	157
Tabel 4.89 Pengujian Validasi Konfirmasi Tambah Akun Siswa Berhasil.....	157
Tabel 4.90 Pengujian Validasi Tambah Data Siswa Format <i>File</i> Tidak Valid.....	158
Tabel 4.91 Pengujian Validasi Tambah Data Siswa Data Siswa Tidak Valid.....	158
Tabel 4.92 Pengujian Validasi Tambah Data Siswa Berhasil.....	159
Tabel 4.93 Pengujian Validasi Tambah Data Siswa Duplikasi Data.....	159
Tabel 4.94 Pengujian Validasi Keluar.....	160
Tabel 4.95 Pengujian Validasi Lihat Pelanggaran.....	160
Tabel 4.96 Interpretasi Skala Likert.....	161
Tabel 4.97 Pernyataan UAT Guru.....	161
Tabel 4.98 Perhitungan Skor Pengujian UAT Guru.....	162
Tabel 4.99 Pernyataan UAT Siswa.....	164
Tabel 4.100 Perhitungan Skor Pengujian UAT Siswa.....	164



DAFTAR GAMBAR

Gambar 2.1 Alur Kerja <i>Waterfall Model</i>	8
Gambar 2.2 Contoh <i>Use Case Diagram</i>	10
Gambar 2.3 Contoh <i>Activity Diagram</i>	12
Gambar 2.4 Contoh <i>Sequence Diagram</i>	15
Gambar 2.5 Contoh <i>Entity Relational Database (ERD)</i>	17
Gambar 2.6 Contoh <i>Class Diagram</i>	18
Gambar 2.7 Contoh <i>Wireframe</i>	19
Gambar 2.8 Contoh <i>Physical Data Model (PDM)</i>	20
Gambar 2.9 Notasi <i>Flow Graph</i>	21
Gambar 3.1 Alur Penelitian	30
Gambar 4.1 Aturan Penomoran Kebutuhan	35
Gambar 4.2 <i>Use Case Diagram</i> SiTalang	40
Gambar 4.3 <i>Activity Diagram</i> Masuk	53
Gambar 4.4 <i>Activity Diagram</i> Reset Password	54
Gambar 4.5 <i>Activity Diagram</i> Ubah Password	56
Gambar 4.6 <i>Activity Diagram</i> Lihat Daftar Siswa	57
Gambar 4.7 <i>Activity Diagram</i> Lihat Pelanggaran Siswa	57
Gambar 4.8 <i>Activity Diagram</i> Tambah Poin Pelanggaran	58
Gambar 4.9 <i>Activity Diagram</i> Edit Poin Pelanggaran	59
Gambar 4.10 <i>Activity Diagram</i> Hapus Poin Pelanggaran	60
Gambar 4.11 <i>Activity Diagram</i> Cetak Poin Pelanggaran	61
Gambar 4.12 <i>Activity Diagram</i> Tambah Motivasi	62
Gambar 4.13 <i>Activity Diagram</i> Tambah Akun Siswa	63
Gambar 4.14 <i>Activity Diagram</i> Tambah Data Siswa	64
Gambar 4.15 Rancangan <i>Activity Diagram</i> Keluar	65
Gambar 4.16 Rancangan <i>Activity Diagram</i> Lihat Pelanggaran	66
Gambar 4.17 <i>Sequence Diagram</i> Masuk	67
Gambar 4.18 <i>Sequence Diagram</i> Reset Password	68
Gambar 4.19 <i>Sequence Diagram</i> Ubah Password	70
Gambar 4.20 <i>Sequence Diagram</i> Lihat Daftar Siswa	71

Gambar 4.21 <i>Sequence Diagram</i> Lihat Pelanggaran Siswa.....	72
Gambar 4.22 <i>Sequence Diagram</i> Tambah Poin Pelanggaran	73
Gambar 4.23 <i>Sequence Diagram</i> Edit Poin Pelanggaran	75
Gambar 4.24 <i>Sequence Diagram</i> Hapus Poin Pelanggaran	76
Gambar 4.25 <i>Sequence Diagram</i> Cetak Poin Pelanggaran	77
Gambar 4.26 <i>Sequence Diagram</i> Tambah Motivasi	78
Gambar 4.27 <i>Sequence Diagram</i> Tambah Akun Siswa	79
Gambar 4.28 <i>Sequence Diagram</i> Tambah Data Siswa	81
Gambar 4.29 <i>Sequence Diagram</i> Keluar	83
Gambar 4.30 <i>Sequence Diagram</i> Lihat Pelanggaran.....	83
Gambar 4.31 <i>Entity Relationship Diagram</i> (ERD).....	84
Gambar 4.32 <i>Class Diagram</i>	85
Gambar 4.33 <i>Wireframe</i> Masuk.....	95
Gambar 4.34 <i>Wireframe</i> Verifikasi <i>Email</i>	96
Gambar 4.35 <i>Wireframe</i> Verifikasi Kode	97
Gambar 4.36 <i>Wireframe</i> Reset <i>Password</i>	98
Gambar 4.37 <i>Wireframe</i> Daftar Siswa	99
Gambar 4.38 <i>Wireframe</i> Detail Poin Pelanggaran Siswa.....	100
Gambar 4.39 <i>Wireframe</i> Tambah Poin	101
Gambar 4.40 <i>Wireframe</i> Edit Poin Pelanggaran	102
Gambar 4.41 <i>Wireframe</i> Ubah <i>Password</i>	103
Gambar 4.42 <i>Wireframe</i> Halaman Kelola Siswa	104
Gambar 4.43 <i>Wireframe</i> Tambah Akun Siswa	105
Gambar 4.44 <i>Wireframe</i> Konfirmasi Tambah Akun Siswa.....	106
Gambar 4.45 <i>Wireframe</i> Tambah Data Siswa.....	107
Gambar 4.46 <i>Wireframe</i> Lihat Pelanggaran	108
Gambar 4.47 <i>Wireframe</i> Hasil Cetak Poin Pelanggaran	109
Gambar 4.48 <i>Physical Data Model</i> (PDM)	110
Gambar 4.49 <i>Flow graph</i> Fungsi Tambah Poin Pelanggaran	123
Gambar 4.50 <i>Flow graph</i> Fungsi Edit Poin Pelanggaran	124
Gambar 4.51 <i>Flow graph</i> Fungsi Hapus Poin Pelanggaran	126
Gambar 4.52 Implementasi Antarmuka Masuk.....	128

Gambar 4.53 Implementasi Antarmuka Verifikasi Email 128

Gambar 4.54 Implementasi Antarmuka Verifikasi Kode..... 129

Gambar 4.55 Implementasi Antarmuka Reset *Password* 129

Gambar 4.56 Implementasi Antarmuka Daftar Siswa 130

Gambar 4.57 Implementasi Antarmuka Lihat Pelanggaran Siswa..... 130

Gambar 4.58 Implementasi Antarmuka Tambah Poin Pelanggaran 131

Gambar 4.59 Implementasi Antarmuka Edit Poin Pelanggaran 131

Gambar 4.60 Implementasi Antarmuka Ubah *Password* 132

Gambar 4.61 Implementasi Antarmuka Halaman Kelola Siswa 132

Gambar 4.62 Implementasi Antarmuka Tambah Akun Siswa 133

Gambar 4.63 Implementasi Antarmuka Konfirmasi Tambah Akun Siswa 133

Gambar 4.64 Implementasi Antarmuka Tambah Data Siswa 134

Gambar 4.65 Implementasi Antarmuka Lihat Pelanggaran..... 134

Gambar 4.66 Implementasi Antarmuka Hasil Cetak Poin Pelanggaran..... 135



DAFTAR LAMPIRAN

LAMPIRAN A HASIL PENGGALIAN KEBUTUHAN DENGAN PIHAK SEKOLAH.....	171
LAMPIRAN B PANDUAN POIN PENANGANAN SISWA.....	176
LAMPIRAN C HASIL PENGUJIAN VALIDASI OLEH GURU.....	180
LAMPIRAN D HASIL PENGUJIAN VALIDASI OLEH SISWA.....	226
LAMPIRAN E HASIL PENGUJIAN UAT OLEH GURU.....	243
LAMPIRAN F HASIL PENGUJIAN UAT OLEH SISWA.....	245



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pendidikan, tingkah laku merupakan suatu hal yang penting mengingat dalam proses pelaksanaan pendidikan terdapat aturan-aturan yang berfungsi untuk melancarkan proses pembelajaran dalam pendidikan. Pembelajaran merupakan sebuah proses timbal balik antara pelajar dengan pendidik. Hal ini juga dikatakan bahwa pembelajaran merupakan proses dalam membantu siswa atau peserta didik dalam mendapatkan ilmu dengan baik. Jadi dengan adanya kebutuhan tingkah laku yang baik tersebut, beberapa instansi pendidikan di Indonesia menerapkan sistem poin untuk memfasilitasi jaminan perilaku baik bagi siswa selaku pelaku utama dalam pendidikan. Sistem poin merupakan salah satu kebijakan sekolah dalam usaha mengurangi tingkat pelanggaran yang dilakukan oleh siswa. Siswa yang melanggar aturan akan diberikan sanksi tergantung dari berapa banyak poin yang didapatkan siswa melalui proses akumulasi.

Akan tetapi dalam penerapannya masih banyak sekolah-sekolah yang menerapkan proses sistem poin tersebut secara manual menggunakan media kertas maupun sejenisnya. Dengan menggunakan media seperti ini dan dilakukan secara manual, proses pengolahan poin pelanggaran siswa menjadi tidak cukup efektif mengingat dengan cara tersebut pengimplementasian sistem poin dapat memakan waktu yang tidak sedikit ketika akan melakukan proses rekapitulasi. Padahal di era sekarang teknologi dapat menjangkau permasalahan tersebut, sehingga akan lebih membantu dalam penerapan sistem poin secara lebih efektif.

Berdasarkan penelitian sebelumnya yang dilakukan oleh Wahyudi dan Aristantia (2017) pada SMK Yayasan Pendidikan Teknologi 1 Purbalingga dengan aplikasi bernama Aplikasi Pengolahan Pelanggaran Siswa Menggunakan SMS Gateway, sekolah tempat dilakukannya penelitian menggunakan sistem poin dalam mendata pelanggaran siswanya. Ketika poin sudah terkumpul, nantinya poin akan dikalkulasikan untuk mendapatkan jumlah poin yang diterima siswa agar pihak sekolah dapat menentukan sanksi apa yang akan diperoleh siswa berdasarkan panduan poin dari sekolah tersebut. Sistem poin ini dinilai lebih mudah untuk diimplementasikan dalam lingkup sekolah karena dari segi prosesnya sekolah bisa memberikan sanksi serta pengambilan keputusan menjadi lebih sederhana. Akan tetapi dalam implementasinya masih menggunakan cara manual yang kurang efektif dari segi waktu maupun biaya.

Untuk itu peneliti membuat sebuah sistem informasi untuk memudahkan dalam proses pendataan pelanggaran siswa di sekolah tersebut menggunakan aplikasi berbasis desktop dengan menggunakan SMS (Short Message Service) gateway. SMS (Short Message Service) Gateway merupakan sebuah proses untuk mengirim dan menerima pesan melalui SMS (Wahyudi & Aristantia, 2017). Ketika siswa melakukan pelanggaran, sistem akan mengirim pesan ke nomor telepon orang tua melalui SMS sehingga orang tua bisa dengan mudah memantau perilaku

siswa di sekolah. Tentunya dengan menggunakan sistem ini, proses pengolahan poin dan pendataan pelanggaran siswa menjadi lebih akurat, cepat, dan mudah.

SMK Negeri 1 Pasuruan merupakan objek penelitian dalam penelitian yang akan dilakukan. Alasan pemilihan SMK Negeri 1 Pasuruan sebagai objek penelitian dikarenakan pada saat proses analisis permasalahan, pihak SMK Negeri 1 Pasuruan memberikan informasi bahwa sekolah menggunakan sistem poin dalam menangani tingkah laku siswa di sekolah. Akan tetapi penerapan sistem poin pada SMK Negeri 1 Pasuruan tersebut masih menggunakan cara manual seperti ditulis pada media kertas dalam penerapannya sehingga menjadikan penerapan sistem poin seperti perekapan poin pelanggaran siswa di SMK Negeri 1 Pasuruan memakan waktu yang tidak sedikit. Pihak sekolah juga memberikan informasi bahwa sekolah membutuhkan aplikasi yang dapat membantu untuk memudahkan dalam melakukan rekapitulasi poin pelanggaran siswa serta dapat melaporkan poin pelanggaran tersebut kepada orang tua siswa pada saat rapor akhir semester. Informasi tersebut didapatkan melalui hasil wawancara yang dilakukan di SMK Negeri 1 Pasuruan pada tanggal 16 Oktober 2020 dengan pihak sekolah yang diwakili oleh Venus, S.Pd. dan Setyo Wahyu Wicaksono, M.Pd. selaku tim kedisiplinan sekolah. Hasil wawancara tersebut dapat dilihat pada Lampiran A.

Berdasarkan uraian tersebut, maka diperlukan sebuah sistem yang dapat membantu untuk mengolah data poin pelanggaran siswa serta dapat melaporkan poin pelanggaran tersebut kepada orang tua siswa di SMK Negeri 1 Pasuruan. Untuk itu peneliti akan melakukan penelitian yang berjudul "PENGEMBANGAN APLIKASI TATA TERTIB PELANGGARAN (SiTalang) SEKOLAH BERBASIS *WEBSITE* MENGGUNAKAN SDLC *WATERFALL* PADA SMKN 1 PASURUAN". Penelitian ini dilakukan untuk mengembangkan suatu sistem informasi agar sekolah dapat melakukan rekapitulasi poin pelanggaran siswa secara lebih efektif dan dapat melaporkan poin pelanggaran yang telah didapatkan oleh siswa kepada orang tua siswa pada rapor akhir semester.

1.2 Rumusan Masalah

1. Bagaimana cara untuk mempermudah pengolahan pelanggaran siswa di SMK Negeri 1 Pasuruan?
2. Apakah aplikasi tata tertib pelanggaran dapat membantu untuk mempermudah pengolahan poin pelanggaran di sekolah?

1.3 Tujuan

1. Mengembangkan sistem yang dapat membantu SMK Negeri 1 Pasuruan dalam mengolah poin pelanggaran siswa di sekolah.
2. Mengetahui apakah sistem pengolah poin pelanggaran siswa dapat membantu dalam memudahkan pengolahan poin pelanggaran siswa di SMK Negeri 1 Pasuruan.

1.4 Manfaat

Manfaat dari penelitian ini yaitu untuk memudahkan SMK Negeri 1 Pasuruan dalam mengolah serta melaporkan poin pelanggaran siswa agar menjadi lebih efektif serta untuk memberikan informasi yang akurat terhadap orang tua siswa terhadap perilaku siswa di sekolah.

1.5 Batasan Masalah

1. Metode pengembangan aplikasi pengolahan poin pelanggaran siswa di SMKN 1 Pasuruan yang digunakan adalah *Waterfall Model* tanpa adanya pemeliharaan sistem (*maintenance*).
2. Pengujian sistem yang dilakukan menggunakan *user acceptance testing* dan *validation testing* untuk menguji apakah program telah sesuai dengan sistem yang dibangun dapat diterima oleh pengguna atau tidak serta untuk menguji fungsional sistem sebelum dapat digunakan oleh pengguna.

1.6 Sistematika Pembahasan

Sistematika pembahasan dalam penulisan skripsi ini meliputi beberapa bagian di antaranya adalah sebagai berikut.

BAB 1 PENDAHULUAN

Bab pendahuluan memuat tentang latar belakang terlaksananya penelitian, rumusan masalah, tujuan dilakukannya penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab landasan kepastakaan menjelaskan mengenai landasan pustaka serta dasar teori yang menunjang proses penelitian.

BAB 3 METODOLOGI PENELITIAN

Pada bab metodologi memuat alur kerja serta metode yang digunakan oleh peneliti dalam menyelesaikan masalah penelitian.

BAB 4 HASIL & PEMBAHASAN

Bab hasil dan pembahasan membahas terkait bagaimana hasil yang diperoleh dari metodologi yang digunakan serta hasil dari proses pengembangan aplikasi Sistem Tata Tertib Pelanggaran (SiTalang).

BAB 5 PENUTUP

Pada bab terakhir atau penutup, terdapat sebuah kesimpulan dari pemaparan penelitian yang sudah dilakukan pada bab sebelumnya serta terdapat saran yang diharapkan dapat membantu penelitian berikutnya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Penelitian Relevan

Penelitian ini relevan dengan penelitian yang telah dilakukan oleh Wahyudi dan Aristantia (2017) dengan judul penelitian “APLIKASI PENGOLAHAN DATA PELANGGARAN SISWA PADA SMK YAYASAN PENDIDIKAN TEKNOLOGI 1 PURBALINGGA TERINTEGRASI DENGAN SMS GATEWAY”. Pada penelitian tersebut menjelaskan bahwa sekolah tempat penelitian dilakukan yaitu SMK Yayasan Pendidikan Teknologi 1 Purbalingga menggunakan sistem poin dalam pengelolaan pelanggaran siswa. Untuk itu Wahyudi dan Aristantia (2017) membuat sebuah aplikasi untuk mengelola data pelanggaran siswa yang terintegrasi dengan *SMS Gateway*. Pada aplikasi tersebut ketika admin telah memasukkan poin pelanggaran yang didapatkan oleh siswa, maka aplikasi akan mengirimkan sebuah pemberitahuan melalui SMS (*Short Message Service*) yang diterima oleh orang tua. Dengan adanya aplikasi tersebut dapat mengurangi adanya kesalahan informasi karena data disimpan di dalam database yang terintegrasi dengan aplikasi. Sehingga membuat proses pengolahan data pelanggaran siswa menjadi lebih efektif. Metode pengembangan sistem yang digunakan oleh penelitian tersebut dalam merancang aplikasi juga sesuai dengan penelitian yang akan dilaksanakan yaitu dengan menggunakan metode *waterfall model*. Metode ini digunakan karena alur proses penerapannya mudah dipahami serta berurutan.

Peneliti juga menemukan penelitian lain yang relevan dengan penelitian yang akan dilakukan yaitu penelitian yang berjudul “SISTEM PENGOLAHAN DATA PERILAKU SISWA DENGAN PENERAPAN KUM DI SMAN 2 DUMAI BERBASIS WEB” yang dilakukan oleh Habisal, Sinaga, dan Saputra (2018). Pada penelitian tersebut peneliti mengembangkan sebuah aplikasi berbasis *website* untuk mengelola data poin pelanggaran siswa. Pada sistem tersebut Admin adalah bagian tata usaha sekolah yang bertugas untuk memasukkan data-data yang terkait dengan siswa ke dalam *database*. Kemudian guru BP (Bimbingan Penyuluhan) bertugas untuk memasukkan poin pelanggaran yang diperoleh siswa. Kemudian ketika tahun ajaran akan selesai, pihak tata usaha dan guru BP menyerahkan data-data poin pelanggaran siswa untuk divalidasi sebelum diserahkan kepada orang tua siswa. Dengan demikian proses pengolahan data pelanggaran siswa menjadi lebih efektif dikarenakan semua proses dilakukan melalui sistem yang terintegrasi.

Berdasarkan kedua penelitian yang telah dilakukan, terdapat persamaan dan perbedaan yang terdapat pada penelitian yang akan dilakukan. Persamaan dari kedua penelitian tersebut dengan penelitian yang akan dilakukan yaitu sama-sama menggunakan metode SDLC *waterfall* dalam pengembangannya serta berbasis *website*. Untuk perbedaan dari kedua penelitian tersebut dengan penelitian yang akan dilakukan yaitu pengembangan *website* pada penelitian yang akan dilakukan menggunakan *framework* Laravel dan menggunakan *user acceptance testing* (UAT) serta *validation testing* dalam pengujian sistemnya.

2.2 Profil Sekolah

2.2.1 Profil SMK Negeri 1 Pasuruan

SMK Negeri 1 Pasuruan terletak di Jalan Veteran, Bugul Lor, Panggungrejo, Kota Pasuruan, Jawa Timur 67122. Menurut *website* SMK Negeri 1 Pasuruan (2017), SMK Negeri 1 Pasuruan memiliki beberapa kompetensi keahlian diantaranya adalah sebagai berikut.

1. Kompetensi Keahlian Otomatisasi dan Tata Kelola Perkantoran.
2. Kompetensi Keahlian Akuntansi dan Keuangan Lembaga
3. Kompetensi Keahlian Bisnis Daring dan Pemasaran
4. Kompetensi Keahlian Kimia Industri.
5. Kompetensi Keahlian Kimia Analisis.
6. Kompetensi Keahlian Teknik Komputer dan Jaringan.
7. Kompetensi Keahlian Multimedia.
8. Kompetensi Keahlian Rekayasa Perangkat Lunak.

2.2.2 Visi Misi SMK Negeri 1 Pasuruan

Visi dan Misi dari SMK Negeri 1 Pasuruan menurut *website* SMK Negeri 1 Pasuruan (2017) adalah sebagai berikut.

2.2.2.1 Visi

Terwujudnya insan yang berakhlak mulia, kreatif, inovatif, mandiri, dan peduli lingkungan.

2.2.2.2 Misi

1. Meningkatkan Nilai Keimanan dan Ketaqwaan kepada Tuhan Yang Maha Esa.
2. Menumbuhkembangkan Jiwa Nasionalisme.
3. Meningkatkan Prestasi dalam Ilmu Pengetahuan, Teknologi, Seni Budaya dan Olahraga.
4. Menumbuhkembangkan Kreatifitas, Inovatif dan Produktifitas dalam Peningkatan Mutu Pendidikan.
5. Menumbuhkembangkan Kemandirian.
6. Menanamkan sikap pelestarian Lingkungan, Pencegahan Terjadinya Pencemaran dan Kerusakan Lingkungan.

2.3 Poin Pelanggaran Siswa

Dalam upaya untuk meningkatkan kedisiplinan siswa selama proses pembelajaran di sekolah, beberapa cara dilakukan untuk membantu meningkatkan kedisiplinan tersebut. Salah satunya yaitu penerapan sistem poin untuk setiap pelanggaran yang dilakukan oleh siswa. Menurut Wahyudi dan

Aristantia (2017), poin pelanggaran tata tertib siswa pada sekolah sangat diperlukan untuk mengetahui tingkat ketidakdisiplinan siswa dalam lingkungan sekolah. Selain memberikan efek jera kepada siswa melalui proses akumulasi poin yang didapat, sistem poin yang diterapkan dapat membantu siswa untuk meningkatkan kepribadiannya dengan kebijakan yang diberikan oleh sekolah dari poin yang didapatkan. Dengan demikian penerapan sistem poin merupakan suatu metode yang cukup efektif dalam menangani permasalahan kedisiplinan siswa di sekolah.

SMK Negeri 1 Pasuruan sebagai objek penelitian menerapkan sistem poin pelanggaran dalam penanganan pelanggaran siswa di sekolah. Sistem poin tersebut digunakan sebagai pedoman dalam memberikan sanksi kepada siswa sesuai dengan pelanggaran yang dilakukan oleh siswa. Terdapat kategori dalam pemberian poin pelanggaran seperti ringan, sedang, dan berat. Beberapa keterangan dari masing-masing kategori pelanggaran adalah sebagai berikut.

1. Kategori Ringan (Kerapian)

- a. Atribut sekolah tidak lengkap (Poin 5).
- b. Seragam tidak sesuai dengan aturan (Poin 5).
- c. sepatu tidak sesuai dengan ketentuan (Poin 5).
- d. Rambut panjang atau diwarnai (Poin 5).
- e. Kuku panjang dan aksesoris berlebihan (Poin 5).
- f. Siswa perempuan menggunakan make up berlebihan (Poin 5).

2. Kategori Sedang (Kedisiplinan)

- a. Alpha (Poin 10).
- b. Bolos/tidak mengikuti pelajaran pada saat jam pelajaran (Poin 10).
- c. Terlambat (Poin 10).
- d. Membuat surat keterangan palsu (Poin 10).

3. Kategori Berat

- a. Obat terlarang/narkoba (Poin 100).
- b. Minuman keras atau sejenisnya (Poin 75).
- c. Mencuri (Poin 75 – 100).
- d. Tindakan asusila, pencemaran nama baik, hamil di luar nikah (Poin 100).
- e. Perkelahian di lingkungan sekolah (Poin 50).
- f. Membawa barang terlarang (senjata tajam, CD pornografi) (Poin 50).
- g. Membawa rokok di lingkungan sekolah (Poin 50).
- h. Merusak fasilitas sekolah (Poin 25).
- i. Menghabiskan uang bukan miliknya (Poin 25).

j. Mengganggu pihak lain (Poin 20).

Untuk keterangan lebih lengkap dapat dilihat pada panduan poin penanganan siswa yang dapat dilihat pada Lampiran B. Dari masing-masing kategori tersebut, pihak sekolah dapat menentukan aksi yang akan diberikan kepada siswa seperti peringatan dan pengarahan, pemanggilan orang tua, dan dipondokkan. Akan tetapi, hasil poin pelanggaran siswa tersebut masih tidak dilaporkan kepada orang tua siswa pada saat rapor akhir semester.

2.4 Sistem Informasi

Pada era perkembangan teknologi yang begitu cepat seperti saat ini, penggunaan teknologi informasi sudah hampir digunakan di berbagai sektor kehidupan dikarenakan penerapannya yang dapat membantu manusia untuk memudahkan pekerjaannya. Proses pemanfaatan terhadap teknologi informasi oleh manusia ini disebut dengan sistem informasi. Sistem informasi menurut Firman (2016) merupakan kombinasi yang teratur dari pengguna, perangkat keras (*hardware*), perangkat lunak (*software*), dan sumber data yang mengubah, mengumpulkan, dan menyebarkan informasi dalam suatu organisasi. Secara lebih singkat, sistem informasi merupakan hubungan antara beberapa komponen dalam mengelola data hingga menjadikan data tersebut sebagai sebuah informasi. Tentunya dengan adanya sistem informasi, informasi dapat lebih cepat tersalurkan dan diterima oleh pengguna.

2.5 Website

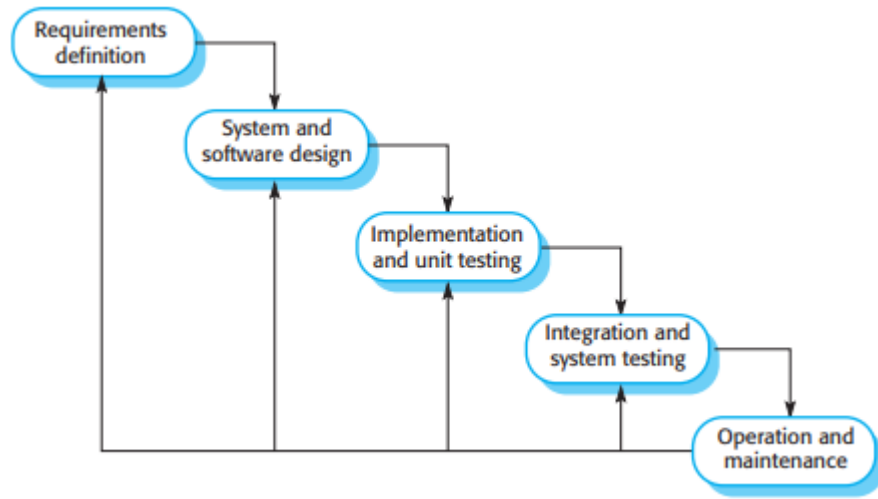
Website merupakan sebuah aplikasi yang didalamnya berisikan elemen multimedia seperti teks, gambar, suara, animasi, dan video serta menggunakan protokol HTTP (*Hypertext Transfer Protocol*) yang dapat diakses melalui perangkat lunak bernama browser (Erinton, Negara and Sanjoyo, 2017). Dengan berbasis *website*, aplikasi dapat mudah diakses di mana saja asalkan terdapat koneksi internet. Untuk dapat mengakses aplikasi berbasis *website* tidak diperlukan sebuah perangkat dengan spesifikasi yang tinggi sehingga dapat memudahkan dalam mengakses aplikasi yang dibangun dengan basis *website* tersebut.

2.6 Waterfall Model

Metode pengembangan yang digunakan dalam penelitian ini yaitu menggunakan metode *waterfall model*. *Waterfall model* dalam penerapannya menerapkan sebuah pengembangan sistem yang terdiri dari tahapan-tahapan dengan arah yang mengalir ke bawah (Sommerville, 2016). *Waterfall model* juga dapat dikatakan sebagai metode yang paling sering digunakan dikarenakan alur pada *model* tersebut mudah dipahami dan berurutan sehingga dapat memudahkan pengembang dalam melakukan pengembangan suatu aplikasi atau sistem.

Tahapan-tahapan yang terdapat pada *waterfall model* ini diantaranya yaitu analisis kebutuhan (*requirement definition*), desain sistem dan perangkat lunak

(*system and software design*), implementasi dan pengujian unit (*implementation and unit testing*), integrasi dan pengujian sistem (*integration and system testing*), dan pemeliharaan (*operation and maintenance*). Fase-fase tersebut harus dilaksanakan secara urut agar proses pengembangan sistem atau aplikasi menjadi lebih efektif. Contoh tahapan-tahapan atau alur kerja pada *waterfall model* menurut Sommerville (2016) ditunjukkan pada Gambar 2.1.



Gambar 2.1 Alur Kerja Waterfall Model

Sumber: Sommerville (2016)

2.6.1 Analisis Kebutuhan/*Requirement Definition*

Analisis atau analisis kebutuhan merupakan proses penggalian kebutuhan yang dilakukan kepada pengguna atau klien. Penggalian kebutuhan bisa dilakukan dengan cara wawancara kepada klien mengenai fungsi-fungsi yang ada pada sistem. Sehingga dalam proses pengembangan aplikasi dapat berjalan sesuai dengan kebutuhan dari klien tersebut. Tahapan ini mendefinisikan fitur dari sistem yang nantinya akan dibangun, kendala yang mungkin terjadi, serta capaian dari tujuan sistem yang akan dibangun dengan berkonsultasi bersama pengguna yang kemudian akan menghasilkan spesifikasi sistem (Sommerville, 2016). Dari spesifikasi sistem tersebut akan dikonversi ke dalam bentuk visual yang akan dilakukan pada tahap berikutnya atau tahap desain sistem.







2.6.1.1 Use Case

Use Case merupakan cara untuk mendeskripsikan interaksi antara pengguna dengan sistem yang menggunakan model grafis dan teks yang terstruktur (Sommerville, 2016). Di dalam *Use case* terdapat sebuah aktor dan sistem, di mana keduanya merupakan gambaran entitas dari manusia dan sistem yang dapat berinteraksi melakukan pekerjaan pada sistem. *Use case* dapat dibagi menjadi dua jenis yaitu *use case diagram* dan *use case scenario*. Di mana keduanya memiliki perbedaan sebagai berikut.

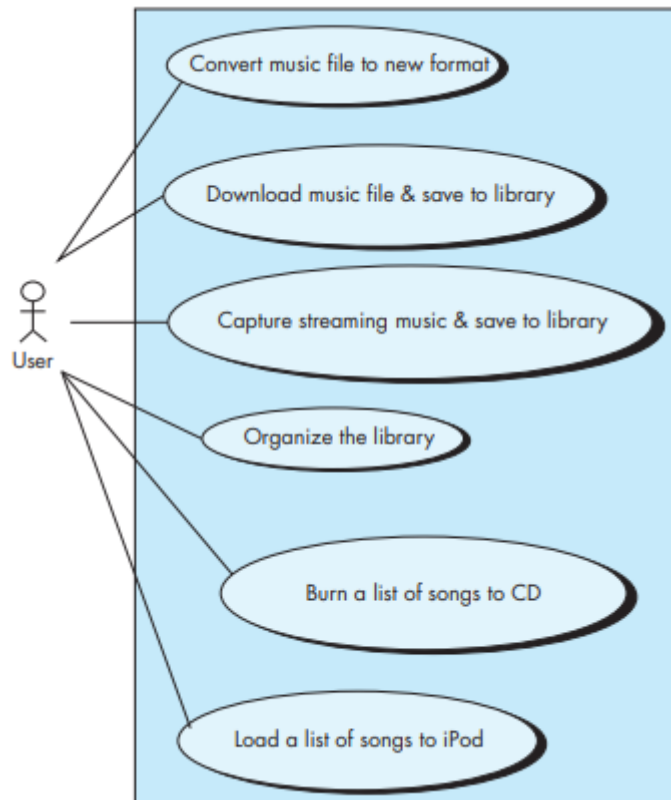
1. *Use Case Diagram*

Use case diagram merupakan representasi dari hubungan antara manusia atau pengguna dengan sistem yang digambarkan dalam bentuk sebuah diagram (Sommerville, 2016). Tabel 2.1 merupakan contoh dari simbol-simbol yang dapat digunakan saat membuat *use case diagram* menurut Aprianti dan Maliha (2016).

Tabel 2.1 Simbol Diagram Use Case

Simbol	Nama	Keterangan
	<i>Actor</i>	Suatu gambaran dari <i>user</i> atau orang yang berinteraksi dengan sistem.
	<i>Use Case</i>	Fungsionalitas yang terdapat pada sistem dan saling bertukar pesan antar unit <i>use case</i> yang lain maupun <i>actor</i> .
	<i>Association</i>	Jembatan atau jalan komunikasi antar <i>actor</i> dengan <i>use case</i> .
	<i>Generalization</i>	Hubungan spesial antara dua <i>use case</i> di mana fungsi dari salah satu <i>use case</i> tersebut lebih umum daripada yang lainnya.
	<i>Include</i>	Hubungan atau relasi <i>use case</i> utama yang hanya bisa dipenuhi dengan bantuan dari <i>use case</i> tambahan (Kurniawan, 2018).
	<i>Extend</i>	Hubungan <i>use case</i> di mana <i>use case</i> utama dapat diperluas oleh <i>use case</i> tambahan (Kurniawan, 2018).

Contoh dari penggunaan *use case diagram* menurut Pressman dan Maxim (2015) ditunjukkan pada Gambar 2.2.



Gambar 2.2 Contoh Use Case Diagram

Sumber: Pressman dan Maxim (2015)

2. Use Case Scenario

Use case scenario merupakan suatu penjelasan mengenai alur dari hubungan manusia atau pengguna dengan sistem yang dijelaskan dalam bentuk kalimat atau teks (Pressman dan Maxim, 2015). Use case scenario juga dapat dikatakan sebagai urutan proses tentang bagaimana suatu fitur dalam sebuah sistem tersebut berjalan. Contoh dari format use case scenario ditunjukkan pada Tabel 2.2.

Tabel 2.2 Format Tabel Use Case Scenario

Nama Fungsi	
Kode Fungsi	
Objektif	
Aktor	
Pre-Condition	
Main Flow	





Tabel 2.2 Format Tabel Use Case Scenario (lanjutan)

Alternative Flow	
Post-Condition	



2.6.1.2 Activity Diagram

Activity diagram menggambarkan perilaku bagian-bagian sistem yang dinamis melalui suatu aliran kontrol tindakan yang dilakukan oleh sistem (Pressman dan Maxim, 2015). *Activity diagram* menunjukkan aktivitas yang melibatkan proses (Sommerville, 2016). *Activity diagram* berbentuk seperti *flowchart* akan tetapi *activity diagram* dapat menjalankan proses secara paralel, sedangkan hal tersebut tidak dapat dilakukan di *flowchart*. Perbedaan yang paling menonjol antara *use case* dan *activity diagram* yaitu pada bagian prosesnya. Jika pada *use case* lebih fokus pada berjalannya proses, sedangkan *activity diagram* menggambarkan sesuatu yang dilakukan pengguna atau *user* kepada sistem sehingga dapat disebut dengan aktivitas atau *activity*. Tabel 2.3 merupakan contoh simbol-simbol dari *activity diagram* pada proses perancangan desain sistem menurut Aprianti dan Maliha (2016).

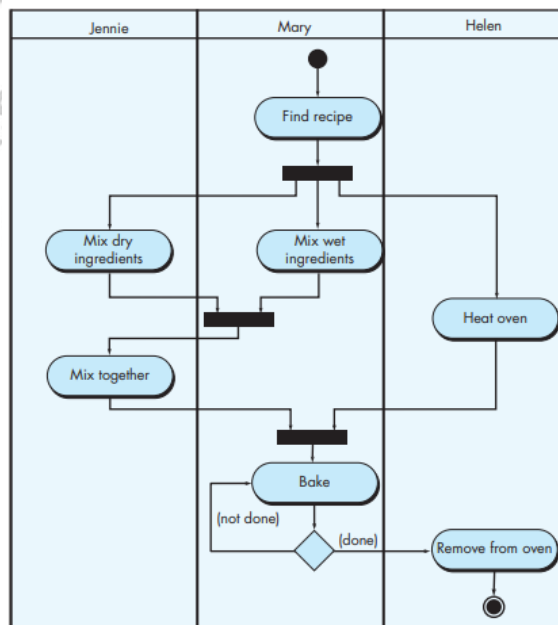
Tabel 2.3 Simbol Activity Diagram

Simbol	Nama	Keterangan
	<i>Start Point</i>	Merupakan tanda dimulainya <i>activity diagram</i> .
	<i>Activity/Action</i>	Merupakan simbol dari aksi maupun aktivitas yang sedang dijalankan oleh sistem.
	<i>Action Flow</i>	Merupakan simbol yang menjadi tempat transisi atau perpindahan dari satu aktivitas ke aktivitas yang lain.
	<i>Decision Symbol</i>	Simbol yang digunakan pada saat terdapat lebih dari satu kondisi yang akan dijalankan oleh sistem.

Tabel 2.3 Simbol Activity Diagram (lanjutan)

Simbol	Nama	Keterangan
	Join Node	Join Node berfungsi untuk menggabungkan beberapa aliran kembali menjadi aliran tunggal.
	End Point	Merupakan tanda untuk mengakhiri activity diagram.

Contoh dari penerapan activity diagram menurut Pressman dan Maxim (2015) ditunjukkan pada Gambar 2.3.



Gambar 2.3 Contoh Activity Diagram

Sumber: Pressman dan Maxim (2015)

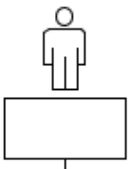

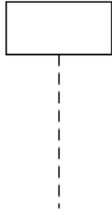
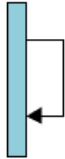
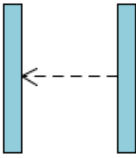
2.6.2 Desain Sistem dan Perangkat Lunak /System and Software Design

Tahap desain akan menerjemahkan hasil dari pendefinisian kebutuhan pada tahap sebelumnya yang dialokasikan terhadap persyaratan perangkat lunak yang akan dibangun dengan mengidentifikasi hubungan antar fitur pada sistem (Sommerville, 2016). Proses desain sistem dijelaskan pada pada sub bab-sub bab berikut.

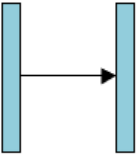
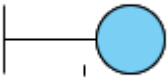


2.6.2.1 Sequence Diagram

Sequence diagram menunjukkan interaksi antara aktor dengan sistem serta dengan komponen-komponen sistem (Sommerville, 2016). Misalnya ketika sistem ingin mengakses data dari sebuah *database*, *sequence diagram* akan memberikan urutan alur proses mulai dari permintaan akses data hingga bagaimana data tersebut dapat ditampilkan pada aplikasi. Dengan adanya *sequence diagram* ini akan lebih memudahkan pengembang dalam menyusun urutan langkah-langkah pada tahap implementasi. Tabel 2.4 merupakan contoh dari simbol-simbol yang digunakan ketika membuat *sequence diagram* menurut Hendini (2016).

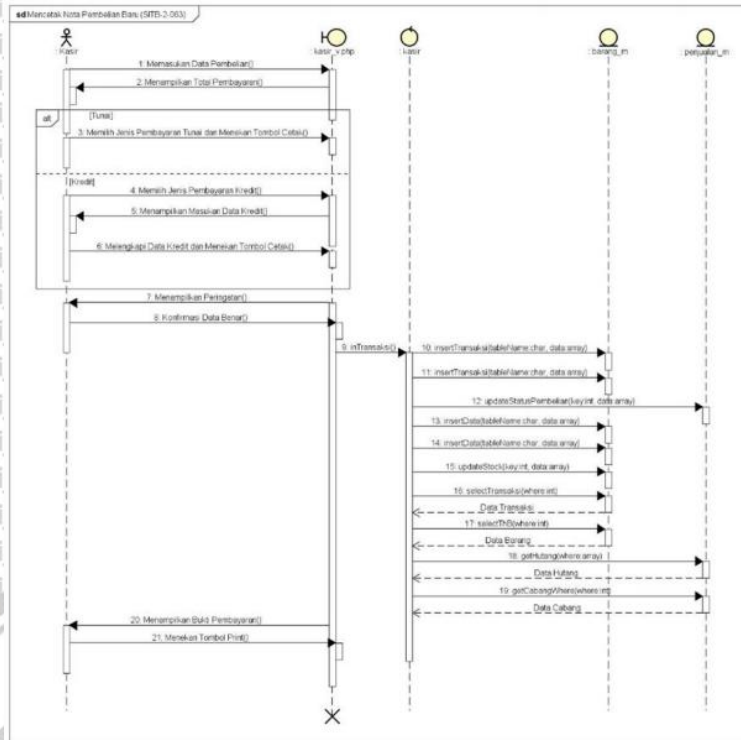
Tabel 2.4 Simbol Sequence Diagram

Simbol	Nama	Keterangan
	<i>Actor</i>	Gambaran dari <i>user</i> yang melakukan interaksi dengan sistem.
	<i>Activation/Time Active</i>	Menyatakan keaktifan suatu objek yang ketika sedang melakukan interaksi.
	<i>Lifeline</i>	Menyatakan keberadaan suatu objek. Di dalamnya terdapat <i>time active</i> ketika objek tersebut berinteraksi pada sistem.
	<i>Self Message/Recursive</i>	Relasi yang menyatakan adanya aktivitas atau sebuah pesan yang ditujukan untuk objek itu sendiri.
	<i>Return Message</i>	Sebuah relasi yang menggambarkan sebuah nilai kembalian dari suatu objek.

Tabel 2.4 Simbol *Sequence Diagram* (lanjutan)

Simbol	Nama	Keterangan
	<i>Message</i>	Menggambarkan terkait interaksi antar objek yang memuat tentang informasi dari aktivitas yang terjadi.
	<i>Boundary Class</i>	Menunjukkan <i>class</i> atau objek yang menjadi <i>user interface</i> pada sistem. Dengan kata lain komponen ini berfungsi untuk menjembatani interaksi dengan sistem yang lain.
	<i>Control Class</i>	Menggambarkan sebuah objek yang berisi tentang logika dari sistem tersebut serta tidak memiliki tanggung jawab kepada suatu entitas.
	<i>Entity Class</i>	Merupakan suatu objek atau <i>class</i> yang berfungsi untuk menyimpan data atau informasi.

Contoh dari penerapan *sequence diagram* dalam proses perancangan desain sistem menurut Sarjono (2018) ditunjukkan pada Gambar 2.4.



Gambar 2.4 Contoh Sequence Diagram

Sumber: Sarjono (2018)



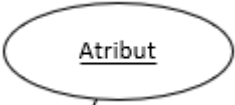
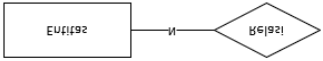
2.6.2.2 Entity Relational Database (ERD)

Entity Relational Database atau ERD merupakan model yang digunakan untuk menggambarkan relasi yang terjadi antar tabel pada suatu basis data atau database (Sulistiyarini, et al., 2018). ERD dapat memudahkan dalam memvisualisasikan struktur database dan hubungan antar tabel pada suatu database. Pada ERD juga dijelaskan hubungan antar tabel pada suatu database apakah termasuk one to many, many to many, atau one to one. Simbol-simbol dari pembuatan ERD menurut Noor (2018) ditunjukkan pada Tabel 2.5.

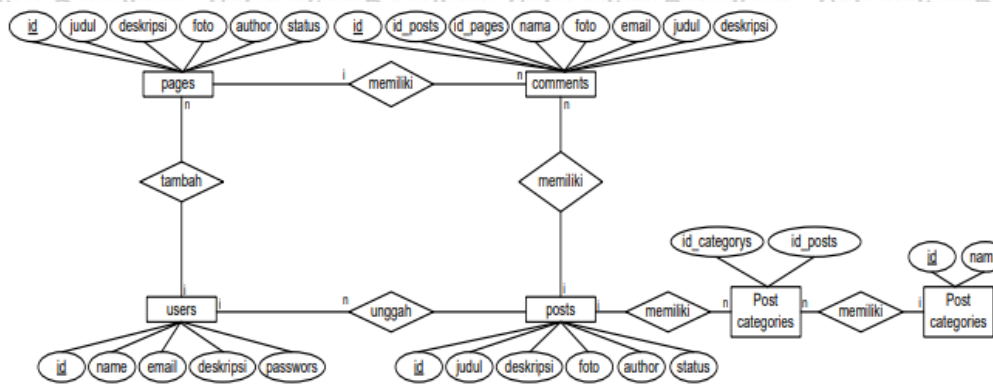
Tabel 2.5 Simbol Entity Relational Database (ERD)

Simbol	Nama	Keterangan
	Entitas/Entity	Merupakan gambaran dari data inti yang akan disimpan pada database atau suatu entitas yang disimpan agar dapat diakses oleh sistem.

Tabel 2.5 Simbol Entity Relational Database (ERD) (lanjutan)

Simbol	Nama	Keterangan
	Relasi/ <i>Relationship</i>	Merupakan sebuah notasi yang menghubungkan antar entitas. Biasanya diawali dengan kata kerja.
	Atribut/ <i>Attribute</i>	Merupakan <i>field</i> atau kolom data yang akan disimpan pada entitas.
	Atribut Kunci/ <i>Key Attribute</i>	Merupakan <i>field</i> atau kolom data yang akan disimpan pada entitas dan digunakan sebagai kunci akses seperti id. Atribut kunci dapat lebih dari satu kolom asalkan kolom yang lain tersebut bersifat unik (berbeda).
	Asosiasi/ <i>Association</i>	Merupakan penghubung antara entitas dengan relasi yang biasanya mempunyai <i>multiplicity</i> dengan keterangan keterhubungan antar entitas yang disebut dengan kardinalitas seperti <i>one to many</i> , <i>many to many</i> , atau <i>one to one</i> .

Contoh dari penerapan ERD menurut Sulistiyarini (2018) ditunjukkan pada Gambar 2.5.



Gambar 2.5 Contoh Entity Relational Database (ERD)

Sumber: Sulistiyarini (2018)

2.6.2.3 Class Diagram

Class diagram merupakan gambaran dari struktur statis pada sistem yang terdapat beberapa elemen seperti class dengan bentuk kotak yang didalamnya terdapat attribute dan operations (Pressman dan Maxim, 2015). Class diagram memudahkan dalam memvisualisasikan struktur kelas dari suatu sistem serta berperan untuk menangkap seluruh arsitektur yang terbentuk dari kelas-kelas sistem. Contoh dari simbol-simbol pada class diagram menurut Aprianti dan Maliha (2016) ditunjukkan pada Tabel 2.6.

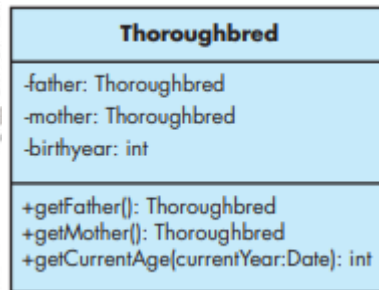
Tabel 2.6 Simbol Class Diagram

Simbol	Nama	Keterangan
	Class	Menggambarkan sebuah class dalam sebuah struktur sistem.
	Association	Merupakan sebuah gambaran dari relasi yang terjadi antar kelas. Biasanya pada relasi association ini disertai dengan multiplicity.
	Directed Association	Merupakan bentuk dari relasi yang terjadi antar kelas yang memiliki makna kelas yang satu digunakan oleh kelas yang lainnya.
	Dependency	Merupakan suatu relasi yang terjadi antar kelas dengan memiliki makna kebergantungan antar kelas.

Tabel 2.6 Simbol Class Diagram (lanjutan)

Simbol	Nama	Keterangan
	Generalization	Merupakan suatu relasi yang menyatakan hubungan inheritance.

Contoh dari *class diagram* dalam proses perancangan desain sistem menurut Pressman dan Maxim (2015) ditunjukkan pada Gambar 2.6.



Gambar 2.6 Contoh Class Diagram

Sumber: Pressman dan Maxim (2015)

2.6.2.4 Pseudocode

Pseudocode merupakan deskripsi dari suatu pemrograman komputer yang menggunakan struktur sederhana agar maksud dari pemrograman tersebut dapat dibaca oleh manusia (Ardiyanto & Purwoto, 2014). Dengan kata lain, *pseudocode* dijelaskan dengan bahasa yang mudah dimengerti oleh manusia untuk mengetahui maksud atau urutan algoritma suatu program dari sistem yang dikembangkan. Tabel 2.7 merupakan contoh dari penulisan *pseudocode* menurut Bella (2019).

Tabel 2.7 Contoh Penulisan Pseudocode

```

Begin
  TM= Time_Manager
  TW1=Time_Worker1
  NW1=Node_Worker1
  TW2=Time_Worker2
  NW2=Node_Worker2
  TR= Time_Tolerance
  If ((TM< TW1+TR) && (TM<TW2+TR))
    Do Loadbalancing()
  If ((TM< TW1+TR) && (TM>TW2+TR))
    Do Web Service route to
  NW1;
  If ((TM> TW1+TR) && (TM<TW2+TR))
    Do Web Service route to
  NW2;
  If ((TM> TW1+TR) && (TM>TW2+TR))
    Print ('All Web Server
  Down')

```

Sumber: Bella (2019)

2.6.2.5 Wireframe

Wireframe merupakan sebuah tampilan yang memfokuskan pada konten dan penggunaan ruang kosong (Junus et al., 2015). Penggambaran dari *wireframe* tidak memerlukan banyak warna, font, maupun gambar. *Wireframe* juga dapat dikatakan sebagai rancangan tampilan yang bertujuan untuk memudahkan dalam penyusunan *interface* dari program atau aplikasi yang akan dibangun. Dengan disusunnya *wireframe* pada tahap desain sistem, proses pengembangan *interface* program atau aplikasi akan menjadi lebih efektif. Gambar 2.7 merupakan contoh dari pembuatan *wireframe* menurut Idris (2018).

Gambar 2.7 Contoh *Wireframe*

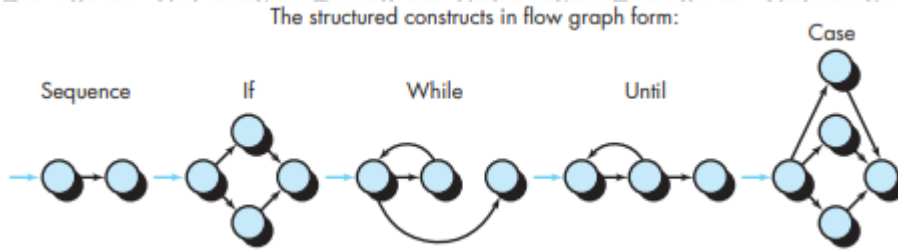
Sumber: Idris (2018)

2.6.3 Pengkodean dan Pengujian Unit/*Implementation and Unit Testing*

Selama tahap ini, tahapan dari desain sistem direalisasikan sebagai suatu program atau bahasa yang dikenali oleh komputer (Sommerville, 2016). Proses mengubah hasil dari desain sistem ke dalam bahasa yang dikenali oleh komputer biasa disebut dengan pengkodean. Beberapa bahasa pemrograman dikenali oleh komputer dan biasa digunakan oleh pengembang suatu sistem yaitu seperti PHP, Python, C++, dan lain sebagainya.

2.6.3.1 *Physical Data Model (PDM)*

Physical Data Model atau PDM merupakan gambaran dari implementasi hasil ERD yang telah dibuat sehingga dapat digunakan oleh sistem sebagai tempat penyimpanan data yang saling berhubungan (Sulistiyarini, et al., 2018). PDM memudahkan dalam memvisualisasikan hasil dari implementasi penyusunan tabel dalam suatu *database* sehingga hubungan relasi antar tabel dapat dijelaskan.



Gambar 2.9 Notasi Flow Graph

Sumber: Pressman dan Maxim (2015)

Berdasarkan Gambar 2.9, anak panah disebut dengan *edge* yang dilambangkan dengan huruf *E* mempresentasikan aliran kontrol, lingkaran disebut dengan simpul atau *node* yang dilambangkan dengan huruf *N* mempresentasikan aksi, daerah yang dibatasi dengan sampul disebut dengan area atau *region* yang ditandai dengan dengan huruf *R*. Simpul yang menggambarkan suatu percabangan disebut dengan *predicate node* yang ditandai dengan huruf *P*.

Cyclomatic complexity merupakan sebuah nilai atau besaran suatu perangkat lunak yang menyatakan ukuran tingkat kompleksitas sebuah program (Pressman dan Maxim, 2015). Menurut Pressman dan Maxim (2015), *cyclomatic complexity* $V(G)$ untuk grafik alir didefinisikan dengan persamaan yang ditunjukkan pada Persamaan 2.1, Persamaan 2.2, dan Persamaan 2.3.

$$V(G) = R \tag{2.1}$$

$$V(G) = E - N + 2 \tag{2.2}$$

$$V(G) = P + 1 \tag{2.3}$$

Persamaan 2.1 menjelaskan bahwa penentuan jumlah jalur yaitu sama dengan jumlah *region* (*R*) yang dimiliki oleh *flowgraph*. Penentuan jalur dasar juga dapat dilakukan dengan melakukan perhitungan aritmatika terhadap jumlah *edge* (*E*) dan *node* (*N*) seperti yang dijelaskan pada Persamaan 2.2. Cara penentuan jalur dasar yang terakhir yaitu dengan melakukan perhitungan aritmatika terhadap jumlah *predicate node* yang ditunjukkan pada Persamaan 2.3.

Penentuan jalur dasar atau *basis path* harus sesuai dengan *cyclomatic complexity* yang telah diperoleh pada perhitungan langkah sebelumnya dan penentuan kasus uji disesuaikan dengan jumlah jalur dasar yang telah diperoleh. Berikut merupakan contoh format penulisan kasus uji yang ditunjukkan pada Tabel 2.8.

Tabel 2.8 Contoh Tabel Kasus Uji

No.	Kode Fungsional	Kasus Uji	Hasil yang Diharapkan

2.6.4 Pengujian/*Integration and System Testing*

Tahap pengujian dapat dikatakan sebagai tahap terakhir dari metode *Waterfall Model* ini. Tahap pengujian merupakan sebuah tahap di mana seluruh fitur akan diuji sebagai sebuah sistem yang utuh untuk memastikan apakah sistem yang telah dibangun sudah memenuhi kebutuhan atau tidak sebelum dapat digunakan oleh pengguna (Sommerville, 2016). Tahapan ini diperlukan karena sebelum aplikasi atau sistem dapat digunakan oleh pengguna, diperlukan cara untuk mendeteksi lebih awal terkait kesalahan-kesalahan yang mungkin terjadi pada saat proses pengembangan sistem yaitu dengan melakukan pengujian. Dengan begitu kesalahan-kesalahan tersebut dapat diatasi lebih awal sehingga pada saat sudah digunakan oleh pengguna dapat berjalan dengan baik.

Pengujian *black box* atau *black box Testing* merupakan pengujian bagian dari sistem (fungsional) dengan menanggapi rangsangan tertentu yang menerapkan beberapa aturan agar rangsangan dapat menjadi tanggapan. Secara sederhana, *black box testing* merupakan suatu pengujian yang berfokus kepada fungsional sistem. Responden atau *tester* dapat menguji aplikasi dengan cara mencoba fitur-fitur yang terdapat pada aplikasi tersebut. Penguji tidak perlu mengetahui bagaimana alur atau proses dalam mengembangkan aplikasi yang diuji.

2.6.4.1 *Validation Testing*

Pengujian validasi atau *validation testing* merupakan sebuah pengujian akhir yang menguji sebuah sistem yang telah menjadi sebuah kesatuan dan kesalahan-kesalahan tampilan yang telah diperbaiki serta berfokus pada tindakan yang dilihat oleh pengguna serta keluaran yang dapat dikenali oleh pengguna juga (Pressman dan Maxim, 2015). Pengujian ini berhasil ketika sistem yang dibangun dapat berjalan sesuai harapan penggunanya. Semua fungsionalitas yang terdapat pada sistem yang telah didefinisikan akan diuji untuk memastikan bahwa sudah tidak ada permasalahan pada fungsional sistem. Tabel 2.9 merupakan contoh dari format pengujian validasi.

Tabel 2.9 Contoh Tabel *Validation Testing*

<i>Test Step</i>	<i>Test Description</i>	<i>Expected Result</i>	Sukses/Gagal
<i>Pre-Condition:</i>			
<i>Scenario Test:</i>			
<i>Expected Result:</i>			
<i>Test Data:</i>			

2.6.4.2 *User Acceptance Testing*

UAT atau yang biasa disebut dengan *user acceptance testing* merupakan suatu pengujian yang bertujuan untuk memastikan bahwa sistem yang telah

dibangun dapat melakukan sesuai apa yang diinginkan serta sebagai bentuk validasi akan pemenuhan persyaratan yang dimiliki oleh bisnis yang telah ditetapkan (Hambling dan van Goethem, 2014). Dengan digunakannya UAT sebagai metode untuk melakukan pengujian, pengembang dapat mengetahui apakah sistem yang sudah dibangun sesuai dengan kebutuhan yang sudah ditetapkan sebelumnya atau tidak. Pengujian UAT dipilih dikarenakan pengujian UAT dapat mengetahui apakah sistem yang dibangun dapat digunakan oleh pengguna atau tidak serta untuk mendapatkan *feedback* dari pengguna agar pengembang dapat mengembangkan aplikasi atau sistem menjadi lebih baik nantinya.

Persyaratan bisnis memberikan informasi terkait hal yang akan dilakukan oleh sistem sesuai dengan persyaratan bisnis. Berdasarkan hal tersebut peneliti dapat melakukan pengujian sesuai dengan persyaratan bisnis yang sudah disepakati sebelumnya. Di sisi lain, terdapat beberapa kriteria penerimaan (*acceptance criteria*) yang dapat menginformasikan kepada pengembang apakah sistem yang dibangun sudah sesuai dan dapat dirilis ke pengguna atau belum. Kriteria penerimaan tersebut diantaranya menurut Hambling dan van Goethem (2014) yaitu:

1. Semua kebutuhan (*requirement*) harus sudah dites sebelum dirilis.
2. Tidak seharusnya terdapat cacat yang serius pada sistem serta tidak lebih dari dua kecacatan pada fitur utama sistem.
3. Tidak boleh terdapat masalah yang tidak teridentifikasi dalam laporan.

Beberapa persyaratan (*requirement*) bisnis harus dites atau dimasukkan dalam pengujian UAT. Biasanya persyaratan bisnis yang digunakan dalam pengujian UAT hanya dua, yaitu kebutuhan fungsional dan non fungsional. Tabel 2.10 merupakan contoh kuesioner dari UAT menurut Sarjono (2018).

Tabel 2.10 Contoh Kuesioner *User Acceptance Testing* (UAT)

No	Nama Pengujian	Penilaian User				
		Sangat Baik	Baik	Netral	Kurang	Sangat Kurang
1	Pengujian Mencetak Nota Pembelian Baru					
2	Pengujian Mengurangi tagihan hutang pelanggan					

Sumber: Sarjono (2018)

Skala Likert merupakan suatu skala yang digunakan untuk mengukur persepsi, sikap, atau pendapat dari individu atau kelompok mengenai peristiwa atau fenomena sosial (Pranatawijaya, et al., 2019). Skala Likert juga merupakan skala yang paling banyak digunakan dalam penelitian yang menggunakan kuesioner kepada responden untuk menentukan tingkat penilaian terhadap kegunaan dari aplikasi yang telah dikembangkan (Wardhono, et al., 2015). Interpretasi dari skala Likert menurut Wardhono (2015) ditunjukkan pada Tabel 2.11.

Tabel 2.11 Interpretasi Skala Likert

Skor Likert	Interpretasi skor dengan interval = 20	Pilihan
1	0% - 19.99%	Sangat Tidak Setuju
2	20% - 39.99%	Tidak Setuju
3	40% - 59.99%	Netral
4	60% - 79.99%	Setuju
5	80% - 100%	Sangat Setuju

Keterangan: Interval = 20 didapatkan dari pembagian nilai 100 dengan jumlah skor Likert

Sumber: Wardhono (2015)

Perhitungan skala Likert dilakukan sesuai dengan jawaban responden dari setiap pernyataan pada kuesioner yang diberikan. Jawaban responden tersebut akan dihitung sesuai skor Likert yang ditentukan dengan rumus seperti keterangan berikut.

1. Skor Ideal

Skor ideal merupakan skor maksimal dari skor pernyataan yang diajukan. Penentuan skor ideal ditentukan oleh banyaknya jumlah responden dengan skor

Likert tertinggi. Rumus perhitungan dari skor ideal menurut Pranatawijaya (2019) ditunjukkan pada Persamaan 2.4.

$$skor\ ideal = jumlah\ responden \times skor\ likert\ tertinggi \quad (2.4)$$

2. Skor Aktual

Skor aktual merupakan skor asli dari jawaban yang diberikan oleh responden sesuai pernyataan yang diajukan. Skor jawaban responden ditentukan melalui skor Likert seperti yang ditunjukkan pada Tabel 2.11. Rumus perhitungan dari skor aktual menurut Pranatawijaya (2019) ditunjukkan pada Persamaan 2.5.

$$skor\ aktual = jumlah\ responden \times skor\ likert \quad (2.5)$$

3. Persentase Skor Aktual

Setelah skor ideal dan skor aktual diketahui, langkah selanjutnya yaitu menghitung persentase dari pernyataan yang telah diajukan. Dari persentase tersebut dapat diketahui apakah pernyataan yang diajukan kepada responden termasuk dalam pilihan sangat setuju, setuju, netral, tidak setuju, atau sangat tidak setuju sesuai dengan interval persentase yang dijelaskan pada Tabel 2.11. Rumus perhitungan dari persentase skor aktual menurut Pranatawijaya (2019) ditunjukkan pada Persamaan 2.6.

$$\% \text{ skor aktual} = \frac{skor\ aktual}{skor\ ideal} \times 100 \quad (2.6)$$

4. Persentase Rata-Rata

Ketika persentase dari skor aktual diketahui dari masing-masing pernyataan, langkah selanjutnya yaitu dengan menghitung rata-rata dari hasil persentase keseluruhan skor aktual pernyataan yang telah diajukan. Hasil dari persentase rata-rata tersebut kemudian dicocokkan kembali dengan interpretasi interval skala Likert sesuai Tabel 2.11 untuk mengetahui apakah keseluruhan pernyataan yang telah diajukan kepada responden dapat dinyatakan sangat setuju, setuju, netral, tidak setuju, atau sangat tidak setuju. Rumus perhitungan dari persentase rata-rata menurut Wardhono (2015) ditunjukkan pada Persamaan 2.7.

$$Rata - Rata = \frac{total\ persentase\ skor\ aktual}{jumlah\ pernyataan} \quad (2.7)$$

Contoh dari kuesioner dengan menggunakan skala Likert ditunjukkan pada Tabel 2.12.

Tabel 2.12 Format Kuesioner Skala Likert

No.	Pernyataan	Tingkat Persetujuan			
		STS	TS	S	SS

2.6.5 Pemeliharaan/*Operation and Maintenance*

Fase pemeliharaan bertujuan untuk menemukan kesalahan yang sebelumnya tidak terdeteksi selama proses pengembangan dan melakukan peningkatan layanan sistem ketika terdapat kebutuhan yang baru ditemukan (Sommerville, 2016). Akan tetapi proses ini dilakukan hanya untuk mengubah dari aplikasi atau perangkat lunak yang sudah ada saja dan bukan untuk membuat aplikasi atau perangkat lunak yang baru.

2.7 Database

Database atau basis data merupakan kumpulan dari suatu data yang saling berhubungan dan tersimpan pada suatu tempat serta menggunakan perangkat lunak untuk memanipulasinya (Novita and Sari, 2015). Manipulasi artinya mengolah suatu data seperti memasukkan data, mengubah data yang ada, menghapus data, dan menampilkan data. Dalam memanipulasi data, terdapat cara-cara yang harus digunakan agar komputer dapat memahami maksud dari manipulasi data tersebut seperti melakukan perintah dengan menuliskan seperti *query select, update, create, dan delete*. Penggunaan *Database* dapat membantu mempermudah dalam pengembangan suatu sistem informasi dikarenakan pengolahan data melalui *Database* dapat menjadi lebih aman, efektif, dan efisien.

2.8 Framework Laravel

Framework Laravel merupakan suatu *Framework* atau kerangka kerja dalam bahasa pemrograman PHP (*Hypertext Preprocessor*) yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya (Erinton, Negara and Sanjoyo, 2017). Laravel dibangun dengan basis MVC (*Model-View-Controller*) serta dilengkapi dengan *command line tool* yang bernama *Artisan* yang dapat digunakan menjadi *packaging bundle* dan instalasi *bundle*. *Framework* ini juga mempunyai beberapa kelebihan seperti dukungan dokumentasi yang baik serta terdapat dukungan *library* yang cukup lengkap.

Model merupakan bagian yang mengatur atau mengolah data yang terdapat pada *database*. Diperlukan sebuah DBMS atau *Database Management System* agar dapat mengolah data menjadi lebih efektif. Salah satu DBMS yang sering digunakan pada saat mengembangkan suatu *website* adalah MySQL. MySQL merupakan sebuah DBMS (*Database Management System*) yang memiliki sifat *open source* serta memungkinkan penggunaannya untuk melakukan modifikasi pada *source code* agar sesuai dengan kebutuhan (Habisal, Sinaga & Saputra, 2018). MySQL bersifat gratis serta memiliki fitur yang cukup lengkap dan didukung dengan beberapa program lainnya seperti XAMPP (*phpMyAdmin*) sehingga dapat diakses secara lokal.

View merupakan bagian yang mengatur tampilan atau *interface* dari suatu *website*. Biasanya *view* diisi dengan *file-file* berformat HTML (*HyperText Markup Language*) sebagai tempat untuk mengatur tampilan atau *interface* dari suatu *website*. HTML atau *HyperText Markup Language* merupakan suatu bahasa

markup yang berfungsi untuk memformat halaman *website* atau suatu alat untuk mengatur tampilan pada halaman *website* (Rahmayu, 2016). Dalam menampilkan suatu *interface*, HTML bertugas untuk memberikan *layout* atau komponen-komponen struktur pada halaman *website*. Untuk memberikan suatu *style* atau *model* untuk tiap komponen yang dibuat oleh *website*, hal tersebut merupakan tugas dari CSS (*Cascading Style Sheet*) yang merupakan suatu alat yang berfungsi untuk memisahkan konten dengan desain pada suatu *website* (Rahmad & Purnama, 2013). Dengan menggunakan HTML dan CSS, proses pembuatan desain antarmuka suatu *website* akan menjadi lebih menarik dan terstruktur.

Controller merupakan bagian yang menghubungkan antara data pada *database* dengan *view* atau tampilan sehingga dapat memberikan fungsi kepada *website* yang dibangun agar menjadi *website* yang dinamis. Sisi *controller* menggunakan bahasa pemrograman PHP yang memberikan perintah atau fungsi untuk dapat menjalankan suatu fungsional pada *website*. PHP merupakan suatu bahasa pemrograman *server-side scripting* yang menyatu dengan HTML (*HyperText Markup Language*) untuk membuat suatu halaman *website* menjadi dinamis (Erinton, Negara and Sanjoyo, 2017). *Server-side scripting* adalah suatu pemrograman yang cara pengeksekusiannya terdapat pada sisi *server*. Seluruh proses dilakukan di *server* akan tetapi hasil dari proses tersebut ditampilkan pada *browser* dengan menggunakan HTML dan CSS pada sisi *view*.

Contoh singkat dari potongan baris kode *Framework* Laravel untuk menghubungkan ke *database* dan melakukan CRUD (*Create, Read, Update, dan Delete*) ditunjukkan pada Tabel 2.13.

Tabel 2.13 Contoh Penulisan Kode Program *Framework* Laravel

Fungsi	Kode Program	Keterangan
Membuat <i>project</i> Laravel	<pre>composer create-project - -prefer-dist laravel/laravel belajar_laravel</pre>	<p>Pembuatan <i>project</i> Laravel dilakukan dengan menggunakan <i>composer</i> diikuti perintah <i>create-project</i>, versi Laravel, <i>directory</i>, nama <i>project</i>.</p>
Menghubungkan ke <i>database</i>	<pre>DB_CONNECTION=mysql DB_HOST=localhost DB_PORT=3306 DB_DATABASE=contoh DB_USERNAME=contoh DB_PASSWORD=contoh</pre>	<p>Pengguna perlu mengisikan masing-masing nilai pada kode program tersebut sesuai dengan data DBMS yang digunakan.</p>
Model	<pre>protected \$table = 'nama tabel';</pre>	<p>Kode program tersebut merupakan kode program yang terdapat pada file <i>model</i>. Pengguna perlu mendefinisikan nama tabel yang terdapat pada <i>database</i> agar dapat terhubung dengan <i>project</i> Laravel.</p>
<i>Create</i>	<pre>Model::create(['kolom' => \$value]);</pre>	<p>Proses tersebut disebut dengan <i>Eloquent database</i>. Pengguna hanya perlu mendefinisikan nama <i>model</i> diikuti dengan perintah <i>create</i> untuk memasukkan data ke <i>database</i>.</p>
<i>Read</i>	<pre>Model::get();</pre>	<p>Proses menggunakan <i>Eloquent</i> dengan menambahkan fungsi <i>get</i> untuk mengambil data pada <i>database</i>.</p>

Tabel 2.13 Contoh Penulisan Kode Program *Framework* Laravel (lanjutan)

<p><i>Update</i></p>	<pre>Model::update(['kolom' => \$value]);</pre>	<p>Proses menggunakan <i>Eloquent</i> dengan menambahkan fungsi <i>update</i> untuk mengubah data pada <i>database</i>.</p>
<p><i>Delete</i></p>	<pre>\$data->delete();</pre>	<p>Proses menghapus data pada <i>database</i> dengan menggunakan fungsi <i>delete</i>.</p>

2.9 XAMPP

XAMPP merupakan suatu *software web server* apache yang di dalamnya terdapat *server MySQL* serta didukung oleh bahasa pemrograman PHP agar dapat menghasilkan suatu *website* yang dinamis (Erinton, Negara and Sanjoyo, 2017). Secara lebih singkat, XAMPP memiliki fungsi sebagai *server* sementara ketika akan membangun sebuah aplikasi atau *website* yang dinamis. *Server XAMPP* dapat dijalankan secara lokal atau tidak memerlukan akses internet sehingga memudahkan pengembang ketika akan membangun sebuah *website* yang dinamis dengan menggunakan XAMPP sebagai *server* sementara.

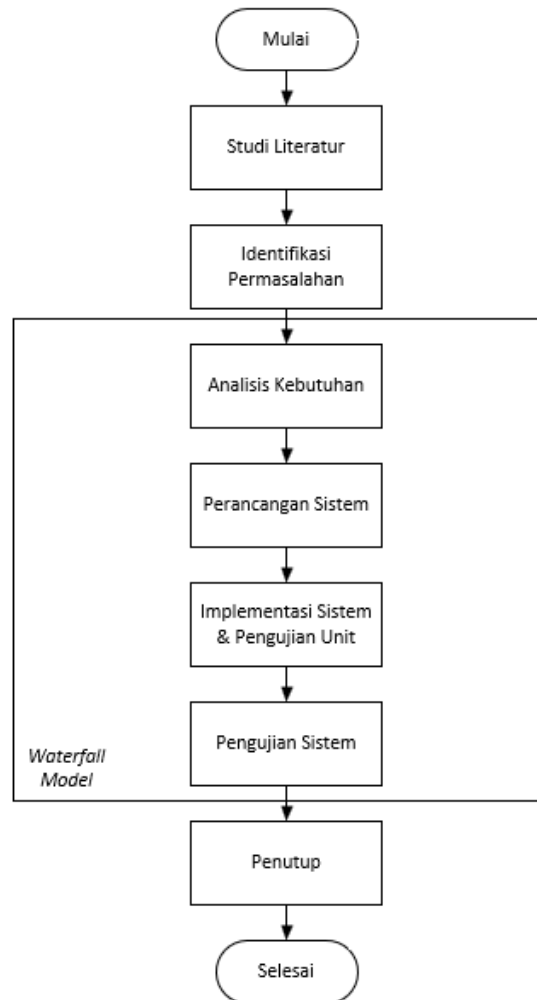
2.10 Bootstrap

Bootstrap merupakan salah satu *framework* atau kerangka kerja yang memiliki fungsi untuk menyelesaikan masalah terkait hal desain tampilan suatu *website* (Habisal, Sinaga & Saputra, 2018). *Framework bootstrap* ini telah menyediakan suatu *library* dalam bentuk CSS maupun Javascript yang dapat dipanggil ketika sedang melakukan desain tampilan *website*. Dokumentasi pada bootstrap juga cukup lengkap untuk memudahkan penggunaanya ketika menggunakan *framework* ini. Dengan menggunakan *framework bootstrap*, proses pengerjaan mendesain tampilan *website* atau *user interface* akan menjadi lebih cepat dikarenakan cara kerjanya yang hanya perlu memanggil tiap kelas yang telah disediakan oleh bootstrap sesuai dengan kebutuhan.

BAB 3 METODOLOGI PENELITIAN

3.1 Alur Penelitian

Dalam penelitian ini, peneliti menggunakan metode *Waterfall Model* dalam melakukan proses penggalian kebutuhan hingga melakukan pengujian terhadap sistem atau aplikasi yang akan dikembangkan. Beberapa tahap atau fase yang terdapat pada *Waterfall Model* yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan perawatan. Akan tetapi pada tahap perawatan/*maintenance* tidak akan dilakukan oleh peneliti dikarenakan program hanya diperuntukkan untuk penelitian kecuali dari pihak sekolah meminta untuk benar-benar mengimplementasikan aplikasi yang telah dibangun pada proses pengolahan poin pelanggaran siswa di sekolah tersebut. Gambar 3.1 merupakan alur penelitian yang akan dilakukan oleh peneliti.



Gambar 3.1 Alur Penelitian

3.2 Studi Literatur

Pada tahap awal atau tahap studi literatur, peneliti akan melakukan pencarian terhadap sumber atau bahan yang menjadi acuan atau dasar dalam melakukan

penelitian. Sumber-sumber yang digunakan oleh peneliti di antaranya yaitu jurnal, *e-book*, dan penelitian sebelumnya yang relevan. Dari sumber-sumber tersebut, seluruhnya masih termasuk dalam lingkup yang relevan dengan penelitian yaitu proses pengembangan sistem informasi. Dengan adanya sumber-sumber tersebut, diharapkan informasi dari penelitian yang akan dilakukan menjadi lebih valid.

3.3 Identifikasi Masalah

Tahap identifikasi masalah merupakan tahap dimana peneliti mencari permasalahan yang terdapat pada lingkungan sekitarnya. Dari tahap identifikasi masalah, peneliti akan mendapat permasalahan yang akan diselesaikan melalui proses penelitian. Beberapa permasalahan tersebut dapat membuat peneliti mengetahui siapa atau apa objek yang akan diteliti serta apa saja yang harus dipersiapkan selama proses penelitian berlangsung nantinya.

3.4 Analisis Kebutuhan

Setelah mendapatkan permasalahan dan mengetahui apa yang harus dilakukan, tahap selanjutnya yaitu peneliti akan melakukan analisis kebutuhan pada objek penelitian yang telah ditentukan oleh peneliti. Metode analisis kebutuhan yang akan dilakukan pada penelitian ini yaitu wawancara. Hasil dari wawancara yang telah dilakukan dapat dilihat pada Lampiran A. Tahap analisis kebutuhan akan menghasilkan data yang berfungsi untuk membantu dalam menyelesaikan penelitian. Pemetaan dari data tersebut berupa *use case diagram*, *use case scenario*, dan *activity diagram* untuk lebih memudahkan dalam memahami data yang telah diperoleh. Dari data-data yang telah didapatkan, peneliti dapat melakukan langkah yang akan dilakukan pada tahap selanjutnya yaitu perancangan sistem.

3.5 Perancangan Sistem

Tahap selanjutnya yaitu tahap perancangan sistem. Pada tahap ini, peneliti akan mencoba mengadopsi data-data kebutuhan yang telah didapatkan pada langkah sebelumnya yaitu analisis kebutuhan ke dalam sebuah proses perancangan sistem. Beberapa langkah yang digunakan peneliti agar mendapatkan perancangan sistem yang sesuai kebutuhan di antaranya yaitu dengan membuat perancangan *sequence diagram*, *class diagram*, pembuatan *Entity Relational Database (ERD)*, pembuatan *pseudocode*, dan pembuatan *Wireframe*.

3.6 Implementasi Sistem dan Pengujian Unit

Tahap ini merupakan tahap di mana hasil dari perancangan sistem yang sudah dilakukan akan diimplementasikan ke dalam bahasa yang dikenali oleh komputer. Tahap ini juga dapat disebut dengan tahap pengkodean, dikarenakan komputer dapat mengenali maksud dari peneliti melalui bahasa pemrograman. Dikarenakan penelitian ini bertujuan untuk mengembangkan suatu aplikasi berbasis *website*,

maka beberapa komponen yang akan digunakan pada penelitian ini diantaranya yaitu PHP (*Hypertext Preprocessor*), *framework* Laravel, *framework* Bootstrap untuk *user interface*, XAMPP sebagai *server* sementara, dan MySQL sebagai *database*.

Pada tahap ini juga akan dilakukan pengujian unit atau *unit testing* dengan menggunakan pengujian *basis path testing* dengan metode *flow graph* dan penentuan *cyclomatic complexity*. Pengujian unit dengan menggunakan metode *basis path testing* berfungsi sebagai dasar dalam penyusunan kasus-kasus uji dari fungsi yang terdapat pada sistem.

3.7 Pengujian Sistem

Pada tahap ini, program atau aplikasi yang telah dirancang dan diimplementasikan hingga menjadi sebuah program yang dapat berjalan masih harus diuji apakah fitur-fitur yang terdapat pada aplikasi tersebut sudah memenuhi kebutuhan pengguna atau belum. Hal ini dilakukan untuk mengetahui kesalahan-kesalahan yang mungkin terjadi pada aplikasi yang telah dibangun. Dengan terdeteksinya kesalahan-kesalahan tersebut, pengembang dapat memperbaikinya sebelum dapat digunakan nantinya.

Penelitian ini akan melakukan jenis pengujian *black box testing* dengan metode *validation testing* dan *user acceptance testing (UAT)*. Pengujian *black box* dengan menggunakan *validation testing* dilakukan untuk mendeteksi kesalahan melalui pengujian pada fungsional aplikasi. Ketika fungsional aplikasi telah diuji dan tidak ditemukan suatu kesalahan, maka aplikasi tersebut sudah dapat dinyatakan layak untuk digunakan oleh pengguna. Pengujian menggunakan *user acceptance testing (UAT)* bertujuan untuk mengetahui apakah hasil dari penelitian ini dapat membantu menyelesaikan permasalahan yang diangkat serta untuk mendapatkan masukan langsung dari pengguna agar ketika nantinya penelitian ini dikembangkan akan terdapat perbaikan pada proses maupun hasilnya. Pernyataan-pernyataan yang terdapat pada kuesioner UAT disesuaikan dengan kebutuhan bisnis yang telah dijelaskan oleh Hambling dan van Goethem (2014) pada sub bab 2.6.4.2 seperti pernyataan terhadap fungsional sistem yang telah diujikan, apakah sistem sesuai dengan kebutuhan, dan tidak boleh ada permasalahan yang tidak teridentifikasi pada laporan. Perhitungan dari pengujian UAT yaitu dengan menggunakan skala Likert agar peneliti dapat menarik kesimpulan apakah sistem yang dibangun dapat diterima atau tidak oleh pengguna dengan interpretasi nilai Likert yang telah ditentukan.

3.8 Penutup

Pemberian kesimpulan penelitian diharapkan mampu menjawab pertanyaan-pertanyaan dari rumusan masalah sehingga dapat diketahui apakah penelitian yang dilakukan berdampak atau tidak pada objek yang diteliti. Untuk pemberian saran, diharapkan dapat memberikan masukan terhadap penelitian selanjutnya agar peneliti pada penelitian selanjutnya dapat memperbaiki kekurangan-kekurangan pada penelitian ini.

BAB 4 HASIL DAN PEMBAHASAN

4.1 Hasil Analisis Kebutuhan

4.1.1 Analisis Permasalahan

Analisis permasalahan pada penelitian ini dilakukan dengan cara melakukan wawancara kepada pihak sekolah SMK Negeri 1 Pasuruan yang diwakili oleh Venus, S.Pd. dan dan Setyo Wahyu Wicaksono, M.Pd. selaku tim kedisiplinan sekolah. Wawancara dilaksanakan di SMK Negeri 1 Pasuruan pada tanggal 16 Oktober 2020. Hasil wawancara akan digunakan sebagai data untuk melakukan pengembangan aplikasi yang akan dibangun. Hasil dari proses wawancara tersebut dapat dilihat pada Lampiran A.

Dari proses wawancara (Lampiran A), peneliti menemukan sebuah permasalahan bahwa pihak sekolah masih menggunakan cara manual dengan ditulis di media kertas saat melakukan penambahan poin pelanggaran sehingga pengimplementasian sistem poin menjadi kurang efektif. Pihak sekolah juga masih merasa perlu adanya improvisasi atau peningkatan pada sisi perekapan data poin pelanggaran siswa karena sesuai hasil wawancara, sekolah masih merasa kesusahan dalam hal merekap poin pelanggaran siswa dan poin pelanggaran masih belum dapat dilaporkan pada rapor siswa. Untuk memudahkan pihak sekolah dalam mengimplementasikan sistem poin yang terdapat pada SMK Negeri 1 Pasuruan, peneliti mengembangkan sebuah aplikasi pengolahan poin pelanggaran siswa berbasis *website* bernama SiTalang (Sistem Tata Tertib Pelanggaran). Dengan pengembangan aplikasi yang berbasis *website*, pihak sekolah dapat melakukan penambahan dan cetak poin pelanggaran siswa melalui perangkat *desktop* maupun *mobile*.

4.1.2 Identifikasi Kebutuhan

Berdasarkan hasil dari proses wawancara yang sudah dilakukan oleh peneliti kepada pihak SMK Negeri 1 Pasuruan, peneliti menemukan sebuah permasalahan bahwa dalam proses implementasi sistem poin yang ada pada SMK Negeri 1 Pasuruan masih belum dapat merekap poin-poin yang didapatkan oleh siswa secara efektif dan belum dapat melaporkan poin pelanggaran siswa yang telah didapat pada rapor siswa. Dari permasalahan tersebut, peneliti dapat mengidentifikasi beberapa fungsional yang nantinya dapat ditambahkan ke aplikasi SiTalang yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Identifikasi Kebutuhan

No.	Kebutuhan
1.	Guru dapat menghitung total poin pelanggaran yang didapatkan oleh siswa.
2.	Guru dapat mengolah poin pelanggaran siswa seperti melihat, menambah, mengubah, dan menghapus poin pelanggaran.

Tabel 4.1 Identifikasi Kebutuhan (lanjutan)

No.	Kebutuhan
3.	Guru dapat mencetak poin pelanggaran yang didapatkan oleh siswa dalam bentuk format PDF agar dapat dilaporkan melalui rapor siswa.
4.	Guru dapat menambahkan teks motivasi untuk dicantumkan pada laporan poin pelanggaran siswa.
5.	Siswa dapat melihat poin pelanggaran yang telah didapatkan.

4.1.3 Identifikasi Aktor

Sub bab ini menjelaskan terkait aktor-aktor atau pengguna yang dapat berinteraksi dengan sistem tata tertib pelanggaran yang ditunjukkan pada Tabel 4.2.

Tabel 4.2 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Aktor pengguna merupakan aktor yang identitasnya masih belum dikenali oleh sistem apakah pengguna tersebut merupakan siswa atau guru. Aktor pengguna dapat dikenali identitasnya hanya setelah menjalankan fungsional masuk.
Guru	Aktor guru merupakan aktor yang identitasnya dikenali oleh sistem sebagai guru serta memiliki akses penuh terhadap sistem seperti melihat seluruh poin pelanggaran yang didapatkan siswa, melakukan tambah, edit, hapus, serta cetak pelanggaran siswa.
Siswa	Aktor siswa merupakan aktor yang identitasnya dikenali oleh sistem sebagai siswa. Aktor siswa hanya memiliki akses untuk melihat poin pelanggaran yang telah didapatkan.

4.1.4 Kebutuhan Fungsional dan Non-Fungsional

Hasil identifikasi kebutuhan fungsional yang telah didapatkan ditunjukkan pada Tabel 4.3.

Tabel 4.3 Identifikasi Kebutuhan Fungsional SiTalang

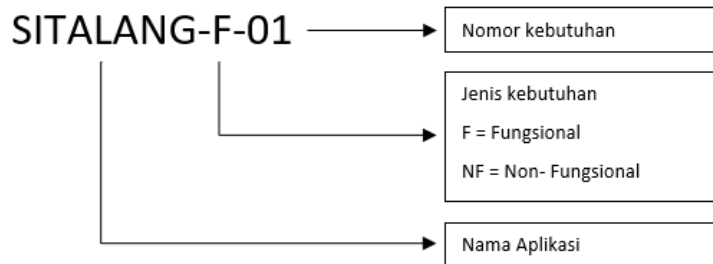
Aktor	Fungsional
Pengguna	1. Masuk 2. Reset <i>Password</i>

Tabel 4.3 Identifikasi Kebutuhan Fungsional SiTalang (lanjutan)

Aktor	Fungsional
Guru	<ol style="list-style-type: none"> 1. Ubah <i>Password</i> 2. Lihat Daftar Siswa 3. Lihat Pelanggaran Siswa 4. Tambah Poin Pelanggaran 5. Edit Poin Pelanggaran 6. Hapus Poin Pelanggaran 7. Cetak Poin Pelanggaran 8. Tambah Motivasi 9. Tambah Akun Siswa 10. Tambah Data Siswa 11. Keluar
Siswa	<ol style="list-style-type: none"> 1. Ubah <i>Password</i> 2. Lihat Pelanggaran 3. Keluar

Hasil kebutuhan dibagi menjadi dua tipe kebutuhan yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Pada tiap-tiap nomor dari kebutuhan fungsional dan non-fungsional memiliki aturan penulisan tersendiri. Gambar 4.1 merupakan aturan penulisan nomor kebutuhan fungsional dan non-fungsional.

Kode Kebutuhan



Gambar 4.1 Aturan Penomoran Kebutuhan

Daftar kebutuhan fungsional dari masing-masing aktor kemudian dijabarkan ke dalam bentuk tabel. Tabel 4.4 akan menjelaskan penjabaran kebutuhan fungsional dari aktor pengguna. Fungsional yang dapat diakses oleh aktor pengguna ditunjukkan pada Tabel 4.4.

Tabel 4.4 Penjabaran Kebutuhan Fungsional Pengguna

No.	Kode Fungsi	Nama Fungsi	Deskripsi
1.	SITALANG-F-01	Masuk	Sistem dapat menjalankan fungsi untuk masuk ke halaman daftar siswa atau halaman utama sistem dan dapat mengenali apakah pengguna merupakan siswa atau guru. Pengguna perlu memasukkan <i>username</i> dan <i>password</i> agar dapat mengakses halaman utama aplikasi.
2.	SITALANG-F-02	Reset <i>Password</i>	Sistem dapat menjalankan fungsi untuk mengatur ulang data <i>password</i> pengguna dan digantikan dengan <i>password</i> baru. Fungsional ini bertujuan untuk mengantisipasi apabila pengguna lupa dengan data <i>password</i> pengguna.

Penjabaran kebutuhan fungsional untuk aktor guru dijelaskan pada Tabel 4.5. Aktor guru merupakan aktor yang telah dikenali oleh sistem setelah melakukan fungsional masuk.

Tabel 4.5 Penjabaran Kebutuhan Fungsional Guru

No.	Kode Fungsi	Nama Fungsi	Deskripsi
1.	SITALANG-F-03	Ubah <i>Password</i>	Sistem dapat menjalankan fungsi untuk mengubah data <i>password</i> guru dan digantikan dengan <i>password</i> baru. Fungsional ini bertujuan untuk mengantisipasi apabila terdapat kebocoran data <i>password</i> guru kepada pihak yang tidak diinginkan.
2.	SITALANG-F-04	Lihat Daftar Siswa	Sistem dapat menjalankan fungsi untuk melihat daftar siswa diikuti dengan beberapa keterangan lainnya seperti jumlah poin yang didapat siswa, nama, dan kelas.

Tabel 4.5 Penjabaran Kebutuhan Fungsional Guru (lanjutan)

No.	Kode Fungsi	Nama Fungsi	Deskripsi
3.	SITALANG-F-05	Lihat Pelanggaran Siswa	Sistem dapat menjalankan fungsi untuk menampilkan pelanggaran-pelanggaran yang telah dilakukan oleh siswa dan jumlah poin pelanggaran yang telah didapatkan oleh siswa. Pada fungsional ini terdapat beberapa informasi siswa seperti nama, kelas, dan jurusan serta guru dapat menambahkan teks motivasi kepada siswa. Terdapat juga menu untuk menambah, mengedit, menghapus, maupun mencetak poin pelanggaran siswa.
4.	SITALANG-F-06	Tambah Poin Pelanggaran	Sistem dapat menjalankan fungsi untuk menambah poin pelanggaran siswa.
5.	SITALANG-F-07	Edit Poin Pelanggaran	Sistem dapat menjalankan fungsi untuk memperbarui informasi pelanggaran seperti pelanggaran yang dilakukan, tanggal, kategori, dan jumlah poin yang telah ditambahkan. Fungsional ini bertujuan untuk mengantisipasi kesalahan pemberian informasi pelanggaran oleh guru pada saat menambahkan pelanggaran siswa.
6.	SITALANG-F-08	Hapus Poin Pelanggaran	Sistem dapat menjalankan fungsi untuk menghapus poin pelanggaran yang telah ditambahkan. Fungsional ini bertujuan untuk mengantisipasi kesalahan oleh guru ketika menambahkan poin pelanggaran siswa.

Tabel 4.5 Penjabaran Kebutuhan Fungsional Guru (Lanjutan)

No.	Kode Fungsi	Nama Fungsi	Deskripsi
7.	SITALANG-F-09	Cetak Poin Pelanggaran	Sistem dapat menjalankan fungsi mencetak keseluruhan poin yang telah didapatkan oleh siswa agar dapat digunakan sebagai bahan untuk melaporkan perilaku siswa di lingkungan sekolah kepada orang tua siswa pada rapor siswa.
8.	SITALANG-F-10	Tambah Motivasi	Sistem dapat menjalankan fungsi untuk menambahkan motivasi yang akan dicantumkan pada laporan poin pelanggaran siswa.
9.	SITALANG-F-11	Tambah Akun Siswa	Sistem dapat menjalankan fungsi untuk menambahkan akun siswa. Fungsional ini bertujuan agar siswa yang telah memiliki akun dapat melihat poin pelanggaran yang telah didapatkan.
10.	SITALANG-F-12	Tambah Data Siswa	Sistem dapat menjalankan fungsi untuk menambahkan data siswa melalui <i>file</i> berformat CSV.
11.	SITALANG-F-13	Keluar	Sistem dapat menjalankan fungsi keluar dari aplikasi menuju halaman masuk dan kembali menjadi pengguna.

Setelah menjabarkan kebutuhan fungsional pada aktor guru, selanjutnya terdapat penjabaran fungsional yang dapat diakses oleh aktor siswa. Penjabaran fungsional yang dapat diakses oleh aktor siswa dapat dilihat pada Tabel 4.6.

Tabel 4.6 Penjabaran Kebutuhan Fungsional Siswa

No.	Kode Fungsi	Nama Fungsi	Deskripsi
1.	SITALANG-F-03	Ubah <i>Password</i>	Sistem dapat menjalankan fungsi untuk mengubah data <i>password</i> siswa dan digantikan dengan <i>password</i> baru. Fungsional ini bertujuan untuk mengantisipasi apabila terdapat kebocoran data <i>password</i> siswa kepada pihak yang tidak diinginkan.

Tabel 4.6 Penjabaran Kebutuhan Fungsional Siswa (lanjutan)

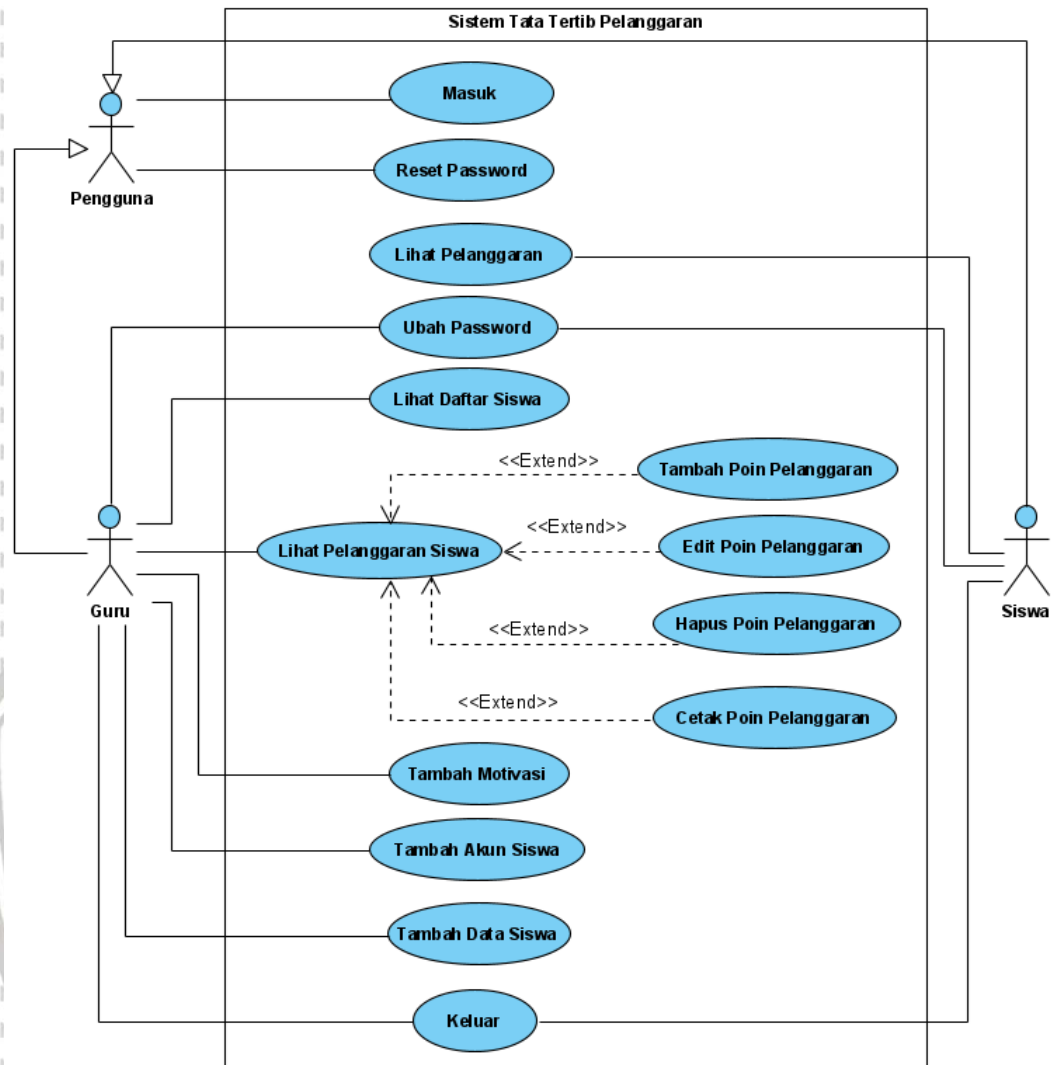
No.	Kode Fungsi	Nama Fungsi	Deskripsi
2.	SITALANG-F-10	Keluar	Sistem dapat menjalankan fungsi keluar dari aplikasi menuju halaman masuk dan kembali menjadi pengguna.
3.	SITALANG-F-11	Lihat Pelanggaran	Sistem dapat menjalankan fungsi untuk melihat pelanggaran yang telah didapatkan oleh siswa yang melakukan masuk.

Daftar kebutuhan non-fungsional aplikasi SiTalang dijabarkan ke dalam bentuk tabel yang dapat dilihat pada Tabel 4.7.

Tabel 4.7 Penjabaran Kebutuhan Non-Fungsional SiTalang

No.	Kode Fungsi	Nama Fungsi	Deskripsi
1.	SITALANG-NF-01	<i>Portability</i>	Sistem dapat mentransfer program dari satu lingkungan sistem perangkat keras dan/atau perangkat lunak ke yang lain (Pressman dan Maxim, 2015). Dengan kata lain sistem dapat diakses pada perangkat desktop maupun mobile selama perangkat tersebut memiliki browser dan koneksi internet

4.1.5 Use Case Diagram



Gambar 4.2 Use Case Diagram SiTalang

Gambar 4.2 menjelaskan bahwa aktor pengguna dapat menjalankan fungsi masuk dan reset *password*. Pengguna perlu melakukan fungsional masuk terlebih dahulu agar dapat dikenali oleh sistem apakah pengguna tersebut merupakan guru atau siswa. Aktor guru memiliki fungsional seperti ubah *password*, lihat daftar siswa, lihat pelanggaran yang didapatkan oleh siswa, tambah poin pelanggaran, hapus poin pelanggaran, edit poin pelanggaran, cetak poin pelanggaran, tambah motivasi, tambah akun siswa, tambah data siswa, dan keluar. Untuk aktor siswa memiliki fungsional yang lebih sedikit yaitu seperti ubah *password*, lihat pelanggaran, dan keluar.

Fungsional lihat pelanggaran pada aktor siswa berbeda dengan fungsional lihat pelanggaran siswa yang terdapat pada aktor guru dikarenakan pada fungsional lihat pelanggaran yang dimiliki oleh aktor siswa hanya dapat melihat poin yang didapatkan oleh siswa yang melakukan masuk dan tidak memiliki akses

untuk menjalankan fungsional seperti tambah, edit, hapus, dan cetak poin pelanggaran.

4.1.6 Use Case Scenario

Use case scenario merupakan penjelasan terkait hubungan antara sistem dengan aktor. Penjelasan dari *use case scenario* dijelaskan dengan menggunakan teks atau kalimat. Berikut merupakan penjelasan *use case scenario* dari masing-masing fungsional dari sistem yang akan dibangun.

4.1.6.1 Use Case Scenario Masuk

Use case scenario masuk menjelaskan bagaimana proses interaksi antara aktor dengan sistem ketika aktor akan melakukan masuk. Proses masuk bertujuan agar sistem dapat mengenali identitas pengguna sebagai aktor guru atau siswa. *Use case scenario* masuk dijelaskan dalam bentuk teks atau kalimat dalam tabel yang ditunjukkan pada Tabel 4.8.

Tabel 4.8 Use Case Scenario Masuk

Nama Fungsi	Masuk
Kode Fungsi	SITALANG-F-01
Objektif	Pengguna dapat masuk ke halaman utama sistem
Aktor	Pengguna
<i>Pre-Condition</i>	Pengguna mengakses <i>website</i> SiTalang
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman masuk. 2. Aktor memasukkan <i>username</i> ke dalam edit teks <i>username</i>. 3. Aktor memasukkan <i>password</i> ke dalam edit teks <i>password</i>. 4. Aktor menekan tombol masuk. 5. Sistem melakukan verifikasi terhadap <i>username</i> dan <i>password</i> yang telah dimasukkan serta membuat <i>session</i> untuk menyimpan informasi pengguna di dalam <i>server</i>. 6. Sistem menampilkan halaman utama aplikasi.
<i>Alternative Flows</i>	<p>5a. <i>Username</i> dan <i>password</i> tidak terverifikasi oleh sistem.</p> <p>5a.1 Sistem menampilkan pesan <i>error</i> "<i>username</i> atau <i>password</i> yang dimasukkan salah".</p> <p>5a.2 Kembali ke flow 1.</p>
<i>Post-Condition</i>	Berhasil masuk ke halaman utama aplikasi.

4.1.6.2 Use Scenario Reset Password

Use case scenario reset *password* menjelaskan bagaimana proses interaksi antara aktor dengan sistem pada saat mengatur ulang data *password* aktor. Pada proses reset *password*, terdapat beberapa langkah sebelum aktor dapat mengubah data *password* yaitu dengan melakukan verifikasi *email* dan verifikasi kode. Proses interaksi antara aktor dengan sistem mulai dari verifikasi *email* hingga berhasil mengatur ulang *password* akan dijelaskan melalui *use case scenario* yang terdapat pada Tabel 4.9.

Tabel 4.9 Use Case Scenario Reset Password

Nama Fungsi	Reset <i>Password</i>
Kode Fungsi	SITALANG-F-02
Objektif	Pengguna dapat mengatur ulang <i>password</i>
Aktor	Pengguna
<i>Pre-Condition</i>	Halaman masuk ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu reset <i>password</i>. 2. Sistem menampilkan halaman verifikasi <i>email</i>. 3. Aktor memasukkan <i>email</i> pengguna yang terdaftar. 4. Aktor menekan tombol <i>submit</i>. 5. Sistem melakukan validasi terhadap <i>email</i> yang telah dimasukkan oleh Aktor. 6. Sistem mengirimkan kode konfirmasi ke <i>email</i> yang telah dimasukkan oleh Aktor. 7. Aktor mendapatkan kode konfirmasi melalui <i>email</i>. 8. Sistem menampilkan halaman konfirmasi kode. 9. Aktor memasukkan kode konfirmasi di edit teks konfirmasi kode. 10. Aktor menekan tombol <i>submit</i>. 11. Sistem melakukan validasi terhadap kode konfirmasi yang telah dimasukkan oleh aktor. 12. Sistem menampilkan halaman reset <i>password</i>. 13. Aktor memasukkan <i>password</i> baru ke dalam edit teks <i>password</i> baru. 14. Aktor mengulang proses memasukkan <i>password</i> baru ke dalam edit teks konfirmasi <i>password</i>. 15. Aktor menekan tombol <i>submit</i>. 16. Sistem melakukan validasi <i>password</i> baru dan konfirmasi <i>password</i> yang telah dimasukkan oleh aktor. 17. Sistem menyimpan <i>password</i> baru.

Tabel 4.9 Use Case Scenario Reset Password (lanjutan)

Alternative Flows	5a.	Email tidak terverifikasi oleh sistem.
	5a.1	Sistem menampilkan pesan error "Email salah."
	5a.2	Kembali ke flow 2.
	11a.	Kode tidak terverifikasi oleh sistem.
	11a.1	Sistem menampilkan pesan error "Kode tidak sesuai."
	11a.2	Kembali ke flow 8.
	16a.	Password baru tidak tervalidasi oleh sistem.
Post-Condition	16a.1	Sistem menampilkan pesan error bahwa masukan aktor tidak valid.
	16a.2	Kembali ke flow 12.
Post-Condition	Berhasil mengatur ulang informasi <i>password</i> dan menampilkan halaman masuk.	

4.1.6.3 Use Case Scenario Ubah Password

Fungsional ubah *password* berbeda dengan fungsional reset *password*. Perbedaan antara kedua fungsional tersebut terletak pada proses ketika akan mengakses halaman ubah maupun reset *password*. Pada fungsional ubah *password*, aktor tidak perlu melakukan verifikasi *email* maupun verifikasi kode akan tetapi aktor harus melakukan masuk terlebih dahulu. Proses ubah *password* akan dijelaskan melalui *use case scenario* yang terdapat pada Tabel 4.10.

Tabel 4.10 Use Case Scenario Ubah Password

Nama Fungsi	Ubah <i>Password</i>
Kode Fungsi	SITALANG-F-03
Objektif	Guru/Siswa dapat mengubah <i>password</i>
Aktor	Guru, Siswa
Pre-Condition	Halaman utama ditampilkan

Tabel 4.10 Use Case Scenario Ubah Password (lanjutan)

Main Flow	1.	Aktor memilih menu ubah <i>password</i> .
	2.	Sistem menampilkan halaman ubah <i>password</i> .
	3.	Aktor memasukkan <i>password</i> baru ke dalam edit teks <i>password</i> baru.
	4.	Aktor mengulang proses memasukkan <i>password</i> baru ke dalam edit teks konfirmasi <i>password</i> .
	5.	Aktor menekan tombol <i>submit</i> .
	6.	Sistem menampilkan modal konfirmasi ubah <i>password</i> .
	7.	Aktor menekan tombol <i>submit</i> untuk menyimpan hasil ubah <i>password</i> .
	8.	Sistem melakukan validasi <i>password</i> baru dan konfirmasi <i>password</i> yang telah dimasukkan oleh Aktor.
	9.	Sistem menyimpan <i>password</i> baru.
Alternative Flows	7a.	Aktor membatalkan konfirmasi ubah <i>password</i> .
	7a.1	Sistem tidak menyimpan hasil ubah <i>password</i> .
	7a.2	Kembali ke flow 2.
	8a.	<i>Password</i> baru tidak tervalidasi oleh sistem.
	8a.1	Sistem menampilkan pesan <i>error</i> bahwa masukan aktor tidak valid.
8a.2	Kembali ke flow 2.	
Post-Condition	Berhasil mengubah informasi <i>password</i> dan menampilkan halaman masuk.	

4.1.6.4 Use Case Scenario Lihat Daftar Siswa

Use case scenario lihat daftar siswa menjelaskan bagaimana proses interaksi antara aktor dengan sistem pada saat melihat daftar siswa. Guru dapat memilih siswa yang akan ditambahkan poin pelanggarannya melalui daftar siswa tersebut.

Tabel 4.11 menjelaskan bagaimana interaksi antara sistem dengan aktor pada saat menjalankan fungsional lihat daftar siswa.

Tabel 4.11 Use Case Scenario Lihat Daftar Siswa

Nama Fungsi	Lihat Daftar Siswa
Kode Fungsi	SITALANG-F-04
Objektif	Guru dapat melihat daftar siswa
Aktor	Guru
Pre-Condition	Halaman utama aplikasi ditampilkan

Tabel 4.11 Use Case Scenario Lihat Daftar Siswa (lanjutan)

<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor melakukan <i>request</i> kepada sistem untuk menampilkan halaman daftar siswa. 2. Sistem mengambil seluruh data siswa dari <i>database</i>.
<i>Alternative Flows</i>	-
Post-Condition	Berhasil menampilkan seluruh data siswa pada halaman daftar siswa.

4.1.6.5 Use Case Scenario Lihat Pelanggaran Siswa

Use case scenario lihat pelanggaran siswa menjelaskan bahwa guru sebagai aktor dapat melihat pelanggaran-pelanggaran yang telah didapatkan oleh siswa dan dapat mengakses fungsional lain seperti tambah poin pelanggaran, edit poin pelanggaran, hapus poin pelanggaran, dan cetak poin pelanggaran. Proses dari fungsional lihat pelanggaran siswa akan dijelaskan melalui *use case scenario* yang terdapat pada Tabel 4.12.

Tabel 4.12 Use Case Scenario Lihat Pelanggaran Siswa

Nama Fungsi	Lihat Pelanggaran Siswa
Kode Fungsi	SITALANG-F-05
Objektif	Guru dapat melihat pelanggaran siswa.
Aktor	Guru
<i>Pre-Condition</i>	Halaman daftar siswa ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu detail yang terdapat pada baris daftar siswa. 2. Sistem melakukan seleksi data pelanggaran siswa yang akan ditampilkan. 3. Sistem menampilkan halaman pelanggaran siswa sesuai siswa yang dipilih.
<i>Alternative Flows</i>	-
Post-Condition	Berhasil menampilkan informasi poin pelanggaran siswa.

4.1.6.6 Use Case Scenario Tambah Poin Pelanggaran Siswa

Use case scenario tambah poin pelanggaran menjelaskan bagaimana proses interaksi antara aktor dengan sistem pada saat aktor akan menambahkan poin pelanggaran siswa. Fungsional tambah poin pelanggaran merupakan fungsional yang dapat diakses oleh aktor guru. Proses tambah poin pelanggaran akan dijelaskan pada *use case scenario* yang terdapat pada Tabel 4.13.

Tabel 4.13 Use Case Scenario Tambah Poin Pelanggaran

Nama Fungsi	Tambah Poin Pelanggaran
Kode Fungsi	SITALANG-F-06
Objektif	Guru dapat menambahkan poin pelanggaran siswa
Aktor	Guru
Pre-Condition	Halaman pelanggaran siswa ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu tambah poin pelanggaran. 2. Sistem menampilkan halaman tambah poin dengan edit teks yang terdiri dari kategori pelanggaran, jenis pelanggaran, tanggal, tindak lanjut, dan jumlah poin. 3. Aktor mengisi <i>form</i> data tambah poin secara lengkap. 4. Aktor menekan tombol <i>submit</i>. 5. Sistem menampilkan modal konfirmasi tambah poin pelanggaran. 6. Aktor menekan tombol <i>submit</i> untuk menyimpan hasil tambah poin pelanggaran. 7. Sistem melakukan validasi dan menyimpan data sesuai masukan aktor.
Alternative Flows	<ol style="list-style-type: none"> 6a. Aktor membatalkan konfirmasi tambah poin pelanggaran. 6a.1 Sistem tidak menyimpan hasil tambah poin pelanggaran. 6a.2 Kembali ke flow 2. 7a. Informasi pelanggaran tidak tervalidasi oleh sistem. 7a.1 Sistem menampilkan pesan <i>error</i> bahwa masukan aktor tidak valid. 7a.2 Kembali ke flow 2.
Post-Condition	Berhasil menambah poin pelanggaran siswa dan menampilkan halaman pelanggaran siswa.

4.1.6.7 Use Case Scenario Edit Poin Pelanggaran Siswa

Use case scenario edit poin pelanggaran menjelaskan bahwa guru sebagai aktor dapat mengubah informasi poin pelanggaran siswa yang telah ditambahkan pada proses tambah poin pelanggaran. Proses fungsional edit poin pelanggaran siswa akan dijelaskan melalui *use case scenario* yang terdapat pada Tabel 4.14.

Tabel 4.14 Use Case Scenario Edit Poin Pelanggaran

Nama Fungsi	Edit Poin Pelanggaran
-------------	-----------------------

Tabel 4.14 Use Case Scenario Edit Poin Pelanggaran (lanjutan)

Kode Fungsi	SITALANG-F-07
Objektif	Guru dapat memperbarui informasi poin pelanggaran siswa
Aktor	Guru
Pre-Condition	Halaman pelanggaran siswa ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu edit yang terdapat pada baris data poin pelanggaran siswa. 2. Sistem melakukan seleksi data poin pelanggaran siswa yang akan diedit. 3. Sistem menampilkan halaman edit poin pelanggaran dengan edit teks yang terdiri dari kategori pelanggaran, jenis pelanggaran, tanggal, tindak lanjut, dan jumlah poin yang masing-masing <i>form</i> sudah terisi data sesuai poin pelanggaran yang dipilih. 4. Aktor mengganti informasi kategori pelanggaran, jenis pelanggaran, tanggal, dan jumlah poin pada data yang terdapat pada masing-masing edit teks secara lengkap. 5. Aktor menekan tombol <i>submit</i>. 6. Sistem menampilkan modal konfirmasi edit poin pelanggaran. 7. Aktor menekan tombol <i>submit</i> untuk menyimpan hasil edit poin pelanggaran. 8. Sistem melakukan validasi dan menyimpan data sesuai masukan aktor.
Alternative Flows	<ol style="list-style-type: none"> 7a. Aktor membatalkan konfirmasi edit poin pelanggaran. 7a.1 Sistem tidak menyimpan hasil edit poin pelanggaran. 7a.2 Kembali ke flow 3. 8a. Informasi pelanggaran tidak tervalidasi oleh sistem. 8a.1 Sistem menampilkan pesan <i>error</i> bahwa masukan aktor tidak valid. 8a.2 Kembali ke flow 3.
Post-Condition	Berhasil memperbarui informasi poin pelanggaran siswa dan menampilkan halaman pelanggaran siswa.

4.1.6.8 Use Case Scenario Hapus Poin Pelanggaran

Use case scenario hapus poin pelanggaran menjelaskan bagaimana proses pada saat guru sebagai aktor dapat menghapus poin pelanggaran yang telah

ditambahkan pada proses tambah poin pelanggaran. Proses hapus poin pelanggaran dijelaskan melalui *use case scenario* yang terdapat pada Tabel 4.15.

Tabel 4.15 Use Case Scenario Hapus Poin Pelanggaran

Nama Fungsi	Hapus Poin Pelanggaran
Kode Fungsi	SITALANG-F-08
Objektif	Guru dapat menghapus poin pelanggaran siswa
Aktor	Guru
<i>Pre-Condition</i>	Halaman pelanggaran siswa ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu hapus yang terdapat pada baris data poin pelanggaran siswa. 2. Sistem menampilkan peringatan modal konfirmasi hapus poin pelanggaran kepada aktor. 3. Aktor mengonfirmasi hapus poin. 4. Sistem melakukan seleksi terhadap data poin pelanggaran yang akan dihapus. 5. Sistem menghapus poin pelanggaran.
<i>Alternative Flows</i>	<ol style="list-style-type: none"> 3a Aktor membatalkan konfirmasi hapus poin pelanggaran. 3a.1 Sistem tidak menghapus poin pelanggaran.
<i>Post-Condition</i>	Berhasil menghapus poin pelanggaran siswa dan menampilkan halaman pelanggaran siswa.

4.1.6.9 Use Case Scenario Cetak Poin Pelanggaran

Use case scenario cetak poin pelanggaran menjelaskan bagaimana interaksi aktor dengan sistem pada saat aktor akan mencetak atau menyimpan informasi pelanggaran siswa ke dalam penyimpanan lokal. Proses fungsional cetak poin pelanggaran akan dijelaskan melalui *use case scenario* yang terdapat pada Tabel 4.16.

Tabel 4.16 Use Case Scenario Cetak Poin Pelanggaran

Nama Fungsi	Cetak Poin Pelanggaran
Kode Fungsi	SITALANG-F-09
Objektif	Guru dapat mencetak poin pelanggaran siswa dalam bentuk dokumen dengan format pdf
Aktor	Guru
<i>Pre-Condition</i>	Halaman pelanggaran siswa ditampilkan

Tabel 4.16 Use Case Scenario Cetak Poin Pelanggaran (lanjutan)

Main Flow	1.	Aktor memilih menu cetak poin pelanggaran.
	2.	Sistem melakukan seleksi terhadap data poin pelanggaran yang akan disimpan atau dicetak.
	3.	Sistem menampilkan <i>preview</i> dari hasil dokumen yang akan disimpan atau dicetak.
	4.	Aktor menekan tombol simpan atau cetak pada <i>preview browser</i> tampilan dokumen.
Alternative Flows	-	
Post-Condition		Berhasil menyimpan atau mencetak poin pelanggaran siswa dalam bentuk dokumen dengan format pdf dan menampilkan halaman pelanggaran siswa.

4.1.6.10 Use Case Scenario Tambah Motivasi

Use case scenario tambah motivasi menjelaskan bagaimana proses pada saat guru sebagai aktor dapat menambahkan motivasi kepada siswa ketika ingin ditampilkan pada hasil cetak poin pelanggaran. Proses tambah motivasi dijelaskan melalui *use case scenario* yang terdapat pada Tabel 4.17.

Tabel 4.17 Use Case Scenario Tambah Motivasi

Nama Fungsi	Tambah Motivasi	
Kode Fungsi	SITALANG-F-10	
Objektif	Guru dapat menambahkan motivasi	
Aktor	Guru	
Pre-Condition	Halaman pelanggaran siswa ditampilkan	
Main Flow	1.	Aktor menekan tombol edit yang terdapat pada bagian motivasi dan penguatan.
	2.	Sistem mengubah status teks area dari <i>disabled</i> menjadi <i>fillable</i> agar dapat diisi oleh aktor.
	3.	Aktor mengisi atau mengubah teks area motivasi.
	4.	Aktor menekan tombol simpan.
	5.	Sistem menyimpan data motivasi sesuai masukan aktor.
Alternative Flows	4a Aktor membatalkan tambah motivasi. 4a.1 Sistem tidak menyimpan masukan motivasi aktor.	
Post-Condition	Berhasil menyimpan motivasi dan menampilkan halaman pelanggaran siswa.	

4.1.6.11 Use Case Scenario Tambah Akun Siswa

Use case scenario tambah akun siswa menjelaskan bagaimana proses ketika guru sebagai aktor dapat menambahkan akun siswa agar siswa dapat melakukan masuk ke sistem dan dapat melihat poin pelanggaran yang telah didapatkan. Proses tambah akun siswa dijelaskan melalui *use case scenario* yang terdapat pada Tabel 4.18.

Tabel 4.18 Use Case Scenario Tambah Akun Siswa

Nama Fungsi	Tambah Akun Siswa
Kode Fungsi	SITALANG-F-11
Objektif	Guru dapat menambahkan akun siswa
Aktor	Guru
<i>Pre-Condition</i>	Halaman kelola siswa ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol tambah akun siswa yang berada pada bagian tambah akun siswa. 2. Sistem menampilkan halaman tambah akun siswa dengan edit teks yang terdiri dari nis, email siswa, <i>set password</i>, dan konfirmasi <i>password</i>. 3. Aktor mengisi <i>form</i> data tambah akun secara lengkap. 4. Aktor menekan tombol cek data. 5. Sistem melakukan validasi terhadap masukan aktor. 6. Sistem menampilkan halaman konfirmasi tambah akun serta menampilkan informasi siswa dan informasi akun siswa yang akan ditambahkan. 7. Aktor menekan tombol konfirmasi. 8. Sistem menampilkan modal konfirmasi. 9. Aktor menekan tombol submit pada modal konfirmasi untuk menyimpan data akun siswa. 10. Sistem menyimpan data akun sesuai masukan aktor.
<i>Alternative Flows</i>	<ol style="list-style-type: none"> 5a. Informasi akun tidak terverifikasi dan tervalidasi oleh sistem. <ol style="list-style-type: none"> 5a.1 Sistem menampilkan pesan <i>error</i> bahwa masukan aktor tidak valid atau tidak terverifikasi. 5a.2 Kembali ke flow 2. 9a. Aktor membatalkan konfirmasi tambah akun. <ol style="list-style-type: none"> 9a.1 Sistem tidak menyimpan data akun siswa. 9a.2 Kembali ke flow 6.

Tabel 4.18 Use Case Scenario Tambah Akun Siswa (lanjutan)

Post-Condition	Berhasil menyimpan data akun siswa dan menampilkan halaman kelola siswa.
----------------	--

4.1.6.12 Use Case Scenario Tambah Data Siswa

Use case scenario tambah data siswa menjelaskan bagaimana proses ketika guru sebagai aktor dapat menambahkan data siswa ke dalam database agar dapat ditampilkan pada halaman daftar siswa. Proses tambah data siswa dijelaskan melalui use case scenario yang terdapat pada Tabel 4.19.

Tabel 4.19 Use Case Scenario Tambah Data Siswa

Nama Fungsi	Tambah Data Siswa
Kode Fungsi	SITALANG-F-12
Objektif	Guru dapat menambahkan data siswa
Aktor	Guru
Pre-Condition	Halaman kelola siswa ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>import</i> data siswa yang berada pada bagian <i>import</i> data siswa. 2. Sistem menampilkan halaman <i>import</i> data siswa dengan elemen yang terdiri dari tombol <i>download</i> format <i>file</i> CSV, tombol <i>download file</i> keterangan kelas id, tombol <i>import</i> untuk mengunggah <i>file</i> data siswa, dan tombol kembali. 3. Aktor menekan tombol <i>import</i>. 4. Sistem menampilkan modal untuk mengunggah <i>file</i> data siswa. 5. Aktor menekan tombol pilih <i>file</i>. 6. Aktor memilih <i>file</i> yang akan diunggah. 7. Aktor menekan tombol <i>import</i> untuk melakukan tambah data siswa. 8. Sistem melakukan validasi terhadap data dari <i>file</i> yang telah diunggah oleh aktor. 9. Sistem menyimpan data siswa.
Alternative Flows	<ol style="list-style-type: none"> 8a. Data siswa tidak tervalidasi oleh sistem. 8a.1 Sistem menampilkan pesan <i>error</i> bahwa data atau format pada <i>file</i> yang diunggah tidak valid. 8a.2 Kembali ke flow 2.
Post-Condition	Berhasil menambahkan data siswa dan menampilkan halaman daftar siswa.

4.1.6.13 Use Case Scenario Keluar

Use case scenario keluar menjelaskan bagaimana interaksi aktor dengan sistem pada saat keluar dari sistem dan kembali menjadi aktor yang belum dikenali oleh sistem. Proses fungsional cetak poin pelanggaran akan dijelaskan melalui *use case scenario* yang terdapat pada Tabel 4.20.

Tabel 4.20 Use Case Scenario Keluar

Nama Fungsi	Keluar
Kode Fungsi	SITALANG-F-13
Objektif	Guru/Siswa dapat keluar dari sistem dan kembali ke halaman masuk
Aktor	Guru, Siswa
<i>Pre-Condition</i>	Halaman utama aplikasi ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu keluar. 2. Sistem menghapus session yang telah disimpan ketika melakukan masuk.
<i>Alternative Flows</i>	-
<i>Post-Condition</i>	Berhasil keluar dari sistem dan menampilkan halaman masuk.

4.1.6.14 Use Case Scenario Lihat Pelanggaran

Use case scenario lihat pelanggaran menjelaskan bahwa siswa sebagai aktor dapat melihat pelanggaran-pelanggaran yang telah didapaknya. Proses dari fungsional lihat pelanggaran akan dijelaskan melalui *use case scenario* yang terdapat pada Tabel 4.21.

Tabel 4.21 Use Case Scenario Lihat Pelanggaran

Nama Fungsi	Lihat Daftar Siswa
Kode Fungsi	SITALANG-F-11
Objektif	Siswa dapat melihat pelanggaran yang didapat
Aktor	Siswa
<i>Pre-Condition</i>	Halaman utama aplikasi ditampilkan
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor melakukan <i>request</i> kepada sistem untuk menampilkan profil siswa dan data pelanggaran siswa. 2. Sistem melakukan seleksi dan mengambil seluruh data siswa dari <i>database</i> sesuai dengan akun siswa yang melakukan masuk.

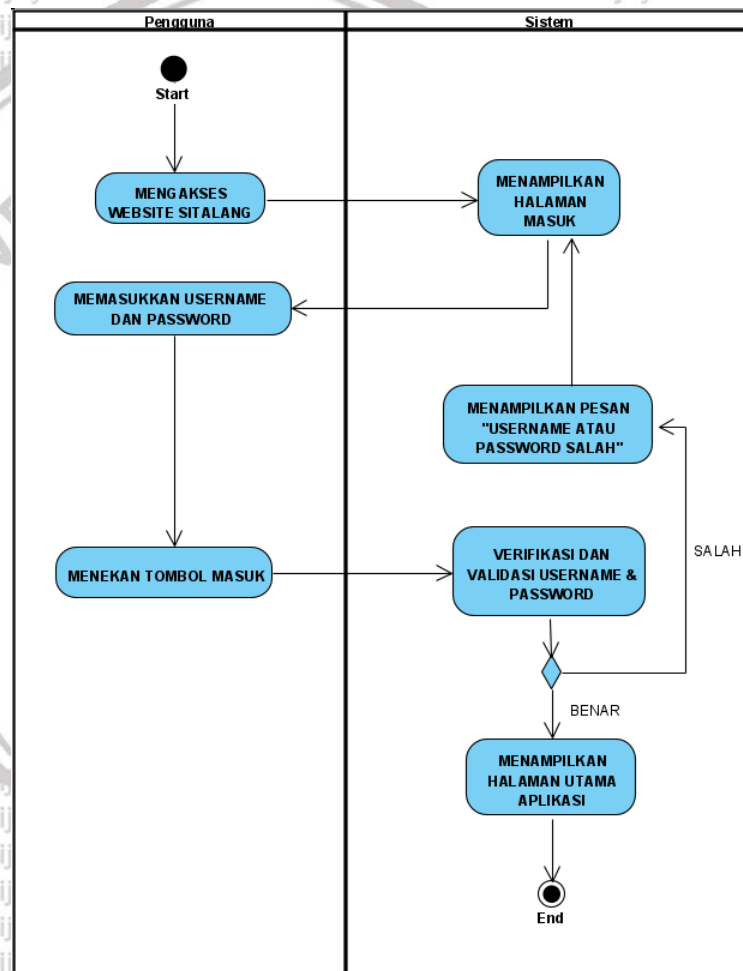
Tabel 4.21 Use Case Scenario Lihat Pelanggaran (lanjutan)

Alternative Flows	
Post-Condition	Berhasil menampilkan data siswa dan pelanggaran siswa.

4.1.7 Activity Diagram

Activity diagram memberikan gambaran terkait alur jalannya proses dari suatu sistem di setiap menu atau fungsional sistem itu sendiri. Activity diagram juga dapat disebut sebagai visualisasi dari use case scenario agar dapat lebih memudahkan untuk dipahami. Pada sub bab ini akan dijelaskan activity diagram dari masing-masing menu atau fungsional sistem yang akan dibangun.

4.1.7.1 Activity Diagram Masuk

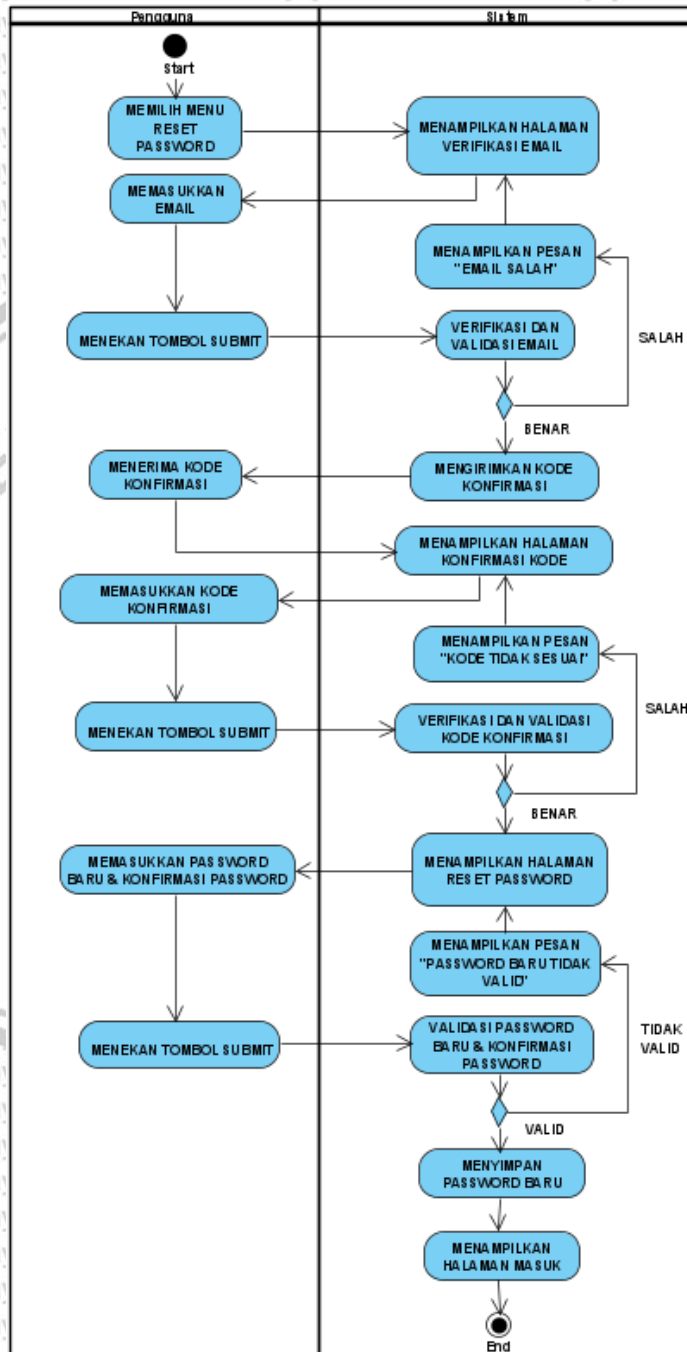


Gambar 4.3 Activity Diagram Masuk

Gambar 4.3 merupakan gambaran dari activity diagram fungsional masuk. Pertama aktor melakukan request kepada sistem untuk menampilkan halaman masuk. Setelah sistem menampilkan halaman masuk, aktor memasukkan username dan password diikuti dengan menekan tombol masuk. Sistem kemudian melakukan verifikasi username dan password yang telah dimasukkan oleh aktor

tersebut apakah benar atau salah. Jika masukan dari pengguna salah, sistem akan kembali menampilkan halaman masuk dan memberikan informasi bahwa *username* atau *password* salah. Apabila masukan dari aktor benar, sistem akan menampilkan halaman utama dari aplikasi yang menandakan bahwa proses masuk telah berhasil.

4.1.7.2 Activity Diagram Reset Password



Gambar 4.4 Activity Diagram Reset Password

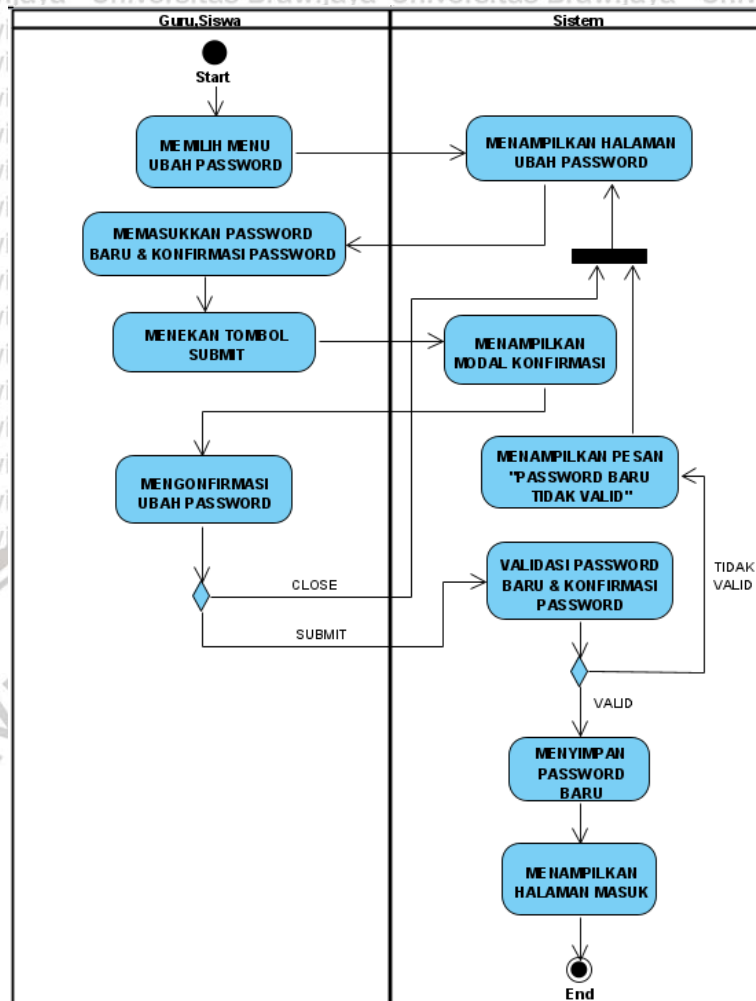
Rancangan dari *activity diagram* dari fungsional reset *password* dapat dilihat pada Gambar 4.4. Aktor memilih menu reset *password* dengan cara menekan *link*

reset *password* yang terdapat pada halaman masuk dan sistem akan menampilkan halaman konfirmasi *email*. Pada konfirmasi *email*, aktor memasukkan *email* yang telah terdaftar untuk dapat segera divalidasi dan diverifikasi oleh sistem. Jika *email* yang dimasukkan oleh aktor valid dan terdaftar pada sistem, sistem akan mengirimkan kode konfirmasi kepada aktor melalui *email* pengguna diikuti dengan menampilkan halaman konfirmasi kode. Apabila masukan email oleh aktor tidak valid dan *email* tidak terdaftar, maka sistem akan tetap menampilkan halaman masukan *email* diikuti dengan menampilkan pesan bahwa masukan tidak valid atau *email* salah.

Pada halaman konfirmasi kode, aktor akan memasukkan kode sesuai dengan kode yang telah diterima oleh aktor melalui *email*. Sistem akan memvalidasi dan memverifikasi kode yang dimasukkan oleh aktor apakah valid dan sesuai dengan kode yang dikirimkan ke *email* aktor atau tidak. Jika kode yang dimasukkan aktor sesuai dengan kode yang telah dikirimkan ke *email*, sistem akan menampilkan halaman reset *password*. Apabila kode yang dimasukkan oleh aktor tidak sesuai dengan kode yang telah dikirimkan ke *email*, sistem akan tetap menampilkan halaman konfirmasi kode diikuti dengan informasi bahwa kode tidak sesuai.

Pada halaman reset *password*, aktor memasukkan *password* baru dan konfirmasi *password*. Masukan *password* baru dan konfirmasi *password* dari aktor kemudian divalidasi oleh sistem untuk memeriksa apakah masukan aktor valid atau tidak. Jika masukan aktor tidak valid, sistem akan menampilkan informasi bahwa masukan *password* baru tidak valid dan tetap menampilkan halaman reset *password*. Apabila masukan *password* baru valid, sistem akan menyimpan *password* baru ke *database*. Setelah proses reset *password* telah berhasil, sistem akan menampilkan halaman masuk dan aktor dapat masuk dengan menggunakan *password* yang sudah diperbarui.

4.1.7.3 Activity Diagram Ubah Password



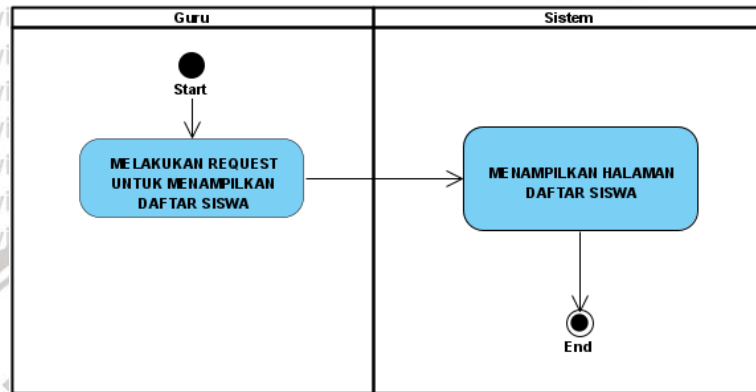
Gambar 4.5 Activity Diagram Ubah Password

Gambar 4.5 merupakan *activity diagram* dari fungsional ubah *password*. Fungsional ubah *password* berbeda dengan fungsional reset *password* dikarenakan pada fungsional ubah *password* ini, aktor dapat melakukan perubahan *password* setelah melakukan masuk ke dalam sistem. Fungsional ini juga bertujuan agar ketika aktor ingin mengubah *password* dapat dilakukan dengan lebih efektif karena aktor tidak perlu lagi melalui proses verifikasi *email* dan konfirmasi kode.

Awal proses dari fungsional ubah *password* ini yaitu aktor memilih menu ubah *password*. Aktor kemudian memasukkan *password* baru dan konfirmasi *password* pada edit teks yang terdapat di halaman ubah *password* tersebut. Setelah memasukkan *password* baru dan konfirmasi *password* serta menekan tombol *submit*, sistem lalu menampilkan modal konfirmasi ubah *password*. Jika aktor memilih untuk menekan tombol *submit* pada modal konfirmasi yang ditampilkan, sistem akan memvalidasi masukan *password* baru dan konfirmasi *password* dari aktor apakah valid atau tidak. Jika masukan tidak valid, sistem akan menampilkan pesan bahwa masukan *password* baru tidak valid dan sistem akan kembali

menampilkan halaman ubah *password*. Apabila masukan aktor tersebut valid, sistem akan menyimpan *password* baru ke dalam *database* sekaligus menandakan bahwa perubahan *password* telah berhasil dilakukan. Jika aktor memilih untuk menekan tombol *close*, sistem tidak akan melakukan aksi apapun dan akan kembali menampilkan halaman ubah *password*. Setelah proses ubah *password* berhasil, sistem akan menampilkan halaman masuk dan aktor dapat masuk kembali dengan menggunakan *password* yang baru.

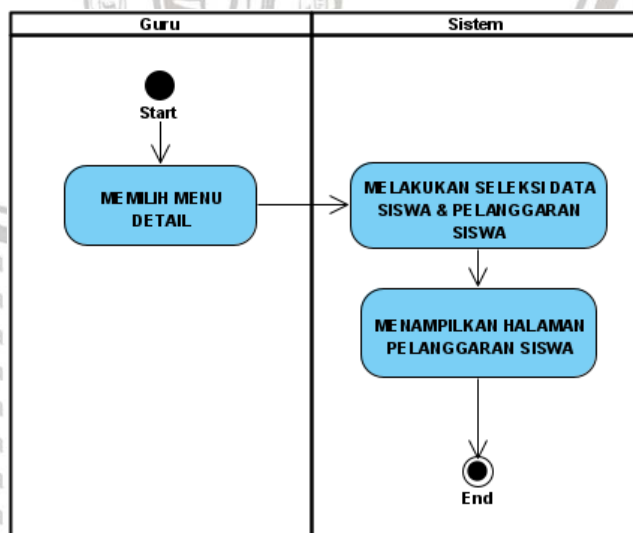
4.1.7.4 Activity Diagram Lihat Daftar Siswa



Gambar 4.6 Activity Diagram Lihat Daftar Siswa

Gambar 4.6 merupakan gambar *activity diagram* untuk menjalankan fungsional melihat daftar siswa. Aktor melakukan *request* kepada sistem untuk menampilkan seluruh daftar siswa yang terdapat pada *database*. Sistem kemudian mengambil data siswa yang terdapat pada *database* dan menampilkan data siswa tersebut pada halaman daftar siswa.

4.1.7.5 Activity Diagram Lihat Pelanggaran Siswa

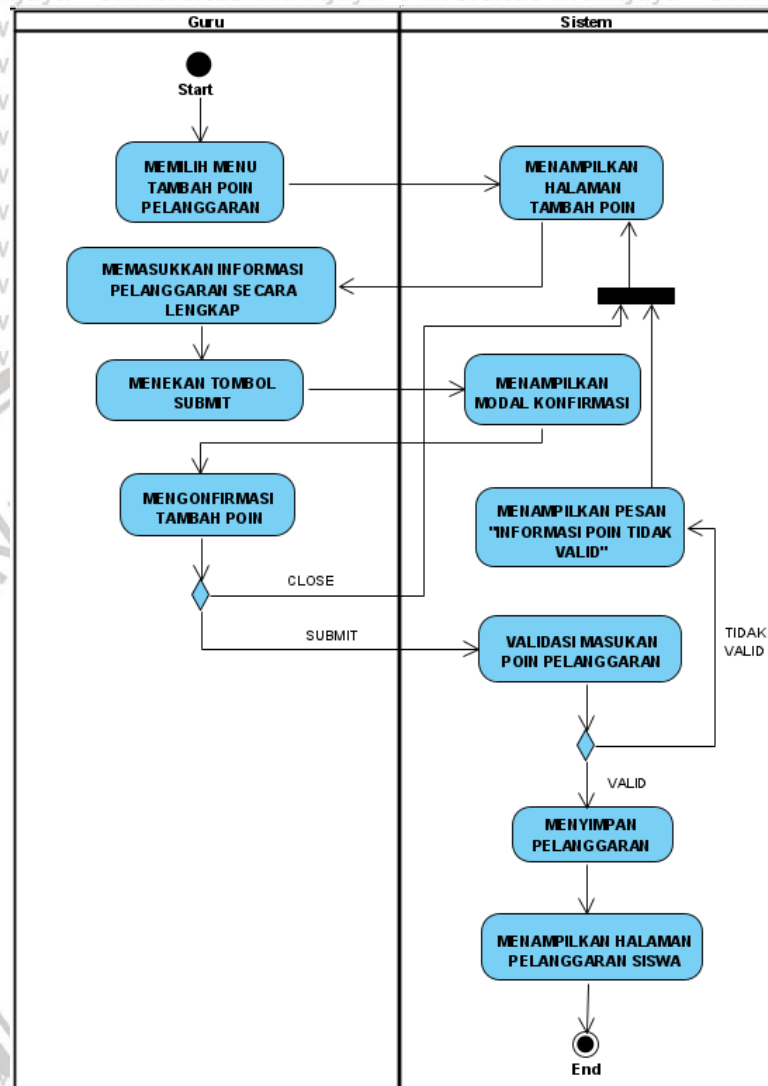


Gambar 4.7 Activity Diagram Lihat Pelanggaran Siswa

Gambar 4.7 merupakan *activity diagram* untuk menjalankan fungsional melihat pelanggaran siswa. Aktor memilih menu detail yang terdapat pada

halaman daftar siswa dengan cara menekan tombol detail pada daftar siswa. Setelah melakukan *request* berdasarkan siswa yang dipilih, sistem akan melakukan seleksi data yang terdapat pada *database* diikuti dengan menampilkannya di halaman pelanggaran siswa sesuai dengan data siswa yang dipilih.

4.1.7.6 Activity Diagram Tambah Poin Pelanggaran

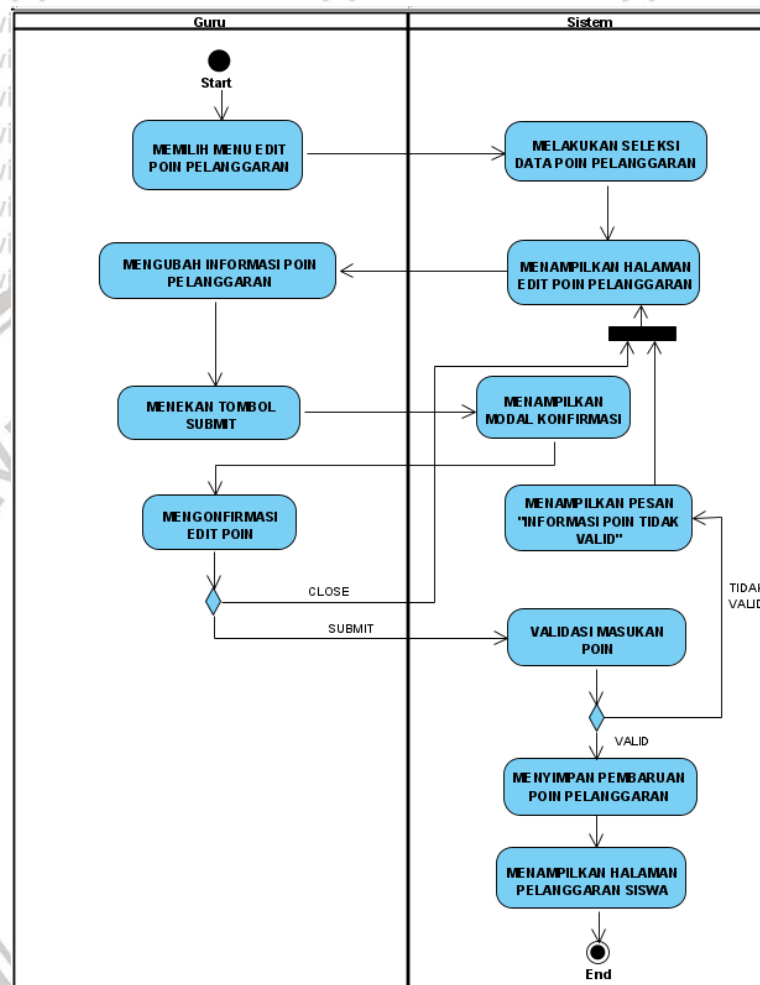


Gambar 4.8 Activity Diagram Tambah Poin Pelanggaran

Gambar 4.8 merupakan *activity diagram* untuk menjalankan fungsional tambah poin pelanggaran siswa. Pada halaman pelanggaran siswa, aktor memilih menu tambah poin pelanggaran dengan cara menekan tombol tambah poin pelanggaran yang terdapat pada halaman pelanggaran siswa. Pada halaman tambah poin pelanggaran, aktor mengisi informasi poin pelanggaran sesuai dengan masing-masing edit teks di halaman tambah poin pelanggaran secara lengkap. Aktor lalu menekan tombol *submit* dan sistem menampilkan modal konfirmasi tambah poin pelanggaran. Jika pada modal konfirmasi aktor memilih untuk menekan tombol *submit*, sistem akan melakukan validasi terhadap hasil masukan aktor tersebut apakah sudah valid atau tidak. Jika tidak valid, maka

sistem akan menampilkan pesan bahwa informasi poin tidak valid dan sistem tetap menampilkan halaman tambah poin pelanggaran. Apabila masukan dari aktor valid, sistem akan menyimpan hasil masukan aktor ke *database* diikuti dengan sistem menampilkan halaman pelanggaran siswa. Jika aktor menekan tombol *close* pada modal konfirmasi, sistem tidak akan melakukan aksi apapun dan kembali menampilkan halaman tambah poin pelanggaran.

4.1.7.7 Activity Diagram Edit Poin Pelanggaran

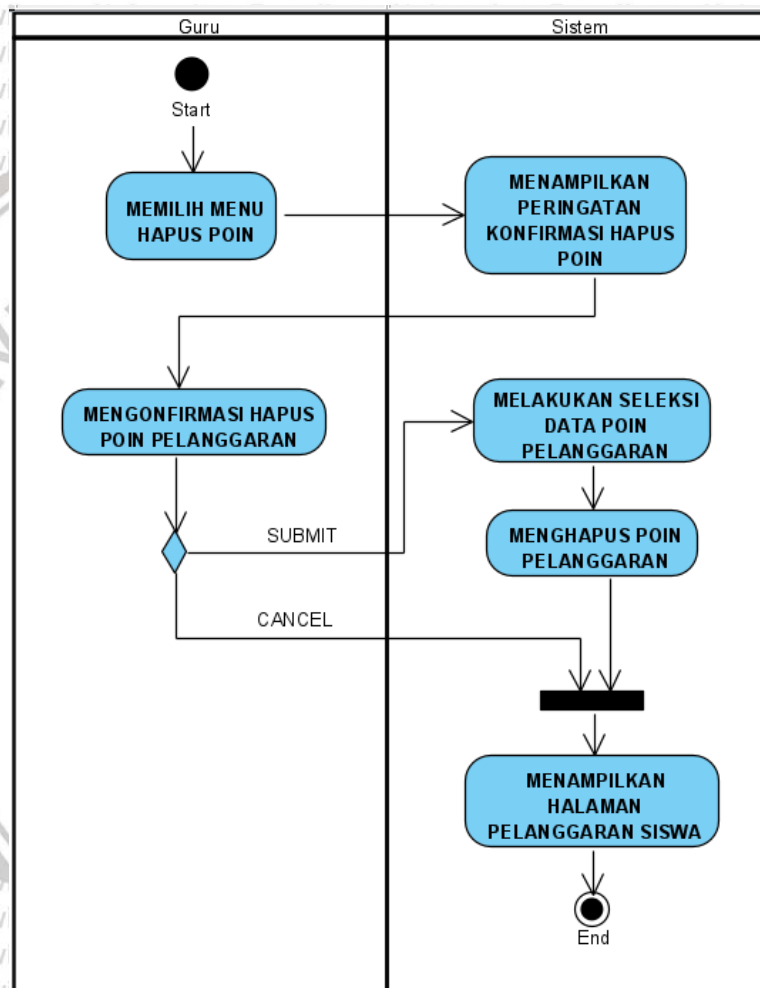


Gambar 4.9 Activity Diagram Edit Poin Pelanggaran

Gambar 4.9 merupakan *activity diagram* untuk menjalankan fungsional edit poin pelanggaran siswa. Pada halaman pelanggaran siswa, aktor memilih menu edit poin pelanggaran dengan cara menekan tombol edit yang terdapat pada halaman pelanggaran siswa. Sebelum menampilkan halaman edit poin pelanggaran, sistem terlebih dahulu melakukan seleksi terhadap data pelanggaran yang telah dipilih untuk dapat ditampilkan pada tiap edit teks yang ada di halaman edit poin pelanggaran untuk memudahkan aktor pada saat akan mengubah informasi poin pelanggaran. Setelah melakukan perubahan terhadap nilai dari edit teks diikuti aktor menekan tombol *submit*, sistem lalu menampilkan modal konfirmasi edit poin pelanggaran. Jika aktor memilih untuk menekan tombol

submit pada modal konfirmasi, sistem akan melakukan validasi terhadap hasil perubahan yang dilakukan oleh aktor apakah valid atau tidak. Jika tidak valid, maka sistem akan menampilkan pesan bahwa informasi poin tidak valid dan sistem kembali menampilkan halaman edit poin pelanggaran. Apabila masukan dari aktor valid, sistem akan menyimpan hasil perubahan informasi poin pelanggaran ke database diikuti dengan sistem menampilkan halaman pelanggaran siswa. Jika aktor memilih untuk menekan tombol *close* pada modal konfirmasi, sistem tidak akan melakukan aksi apapun dan tetap menampilkan halaman edit poin pelanggaran.

4.1.7.8 Activity Diagram Hapus Poin Pelanggaran

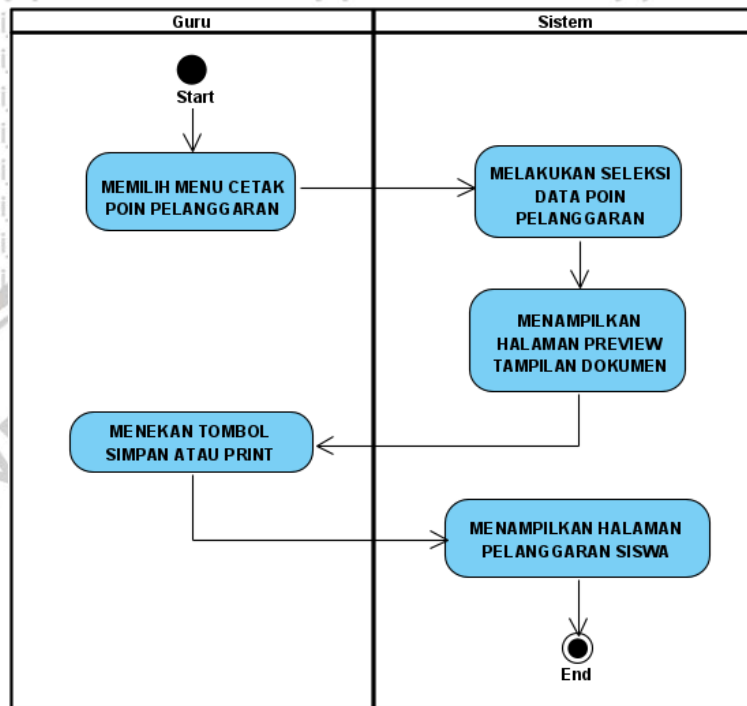


Gambar 4.10 Activity Diagram Hapus Poin Pelanggaran

Gambar 4.10 merupakan *activity diagram* dari fungsional hapus poin pelanggaran siswa. Pada halaman pelanggaran siswa, aktor memilih menu hapus poin pelanggaran dengan cara menekan tombol hapus yang terdapat pada halaman pelanggaran siswa. Sistem kemudian memberikan sebuah peringatan konfirmasi apakah aktor benar-benar ingin menghapus poin pelanggaran yang telah dipilih tersebut atau tidak. Jika yakin, maka aktor dapat mengonfirmasi hapus poin pelanggaran dan sistem akan melakukan seleksi terhadap data poin

pelanggaran siswa yang akan dihapus tersebut. Setelah data poin pelanggaran ditemukan, sistem kemudian menghapus poin pelanggaran yang telah dipilih oleh aktor tersebut dan sistem menampilkan halaman pelanggaran siswa. Jika aktor tidak yakin yang ditandai dengan menekan tombol *cancel*, maka sistem tidak akan menghapus data poin pelanggaran dan kembali menampilkan halaman pelanggaran siswa.

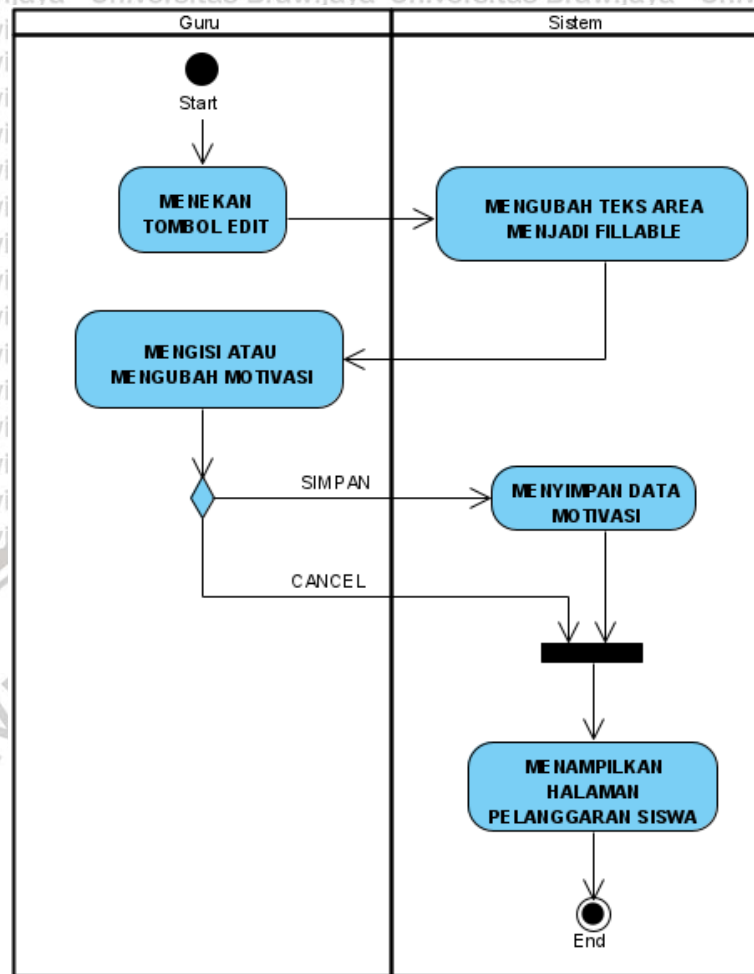
4.1.7.9 Activity Diagram Cetak Poin Pelanggaran



Gambar 4.11 Activity Diagram Cetak Poin Pelanggaran

Gambar 4.11 merupakan gambaran dari *activity diagram* yang menjalankan fungsional cetak poin pelanggaran. Aktor memilih menu cetak poin pelanggaran dengan cara menekan tombol cetak poin pelanggaran yang terdapat pada halaman pelanggaran siswa. Sistem kemudian melakukan seleksi terhadap data siswa maupun data poin pelanggaran yang dipilih dan menampilkan preview tampilan dari hasil cetak poin pelanggaran tersebut. Aktor dapat menyimpan hasil cetak poin pada penyimpanan lokal aktor atau dapat langsung mencetaknya ke dalam bentuk *hardfile*. Setelah berhasil melakukan simpan atau cetak poin, sistem kemudian akan menampilkan halaman pelanggaran siswa.

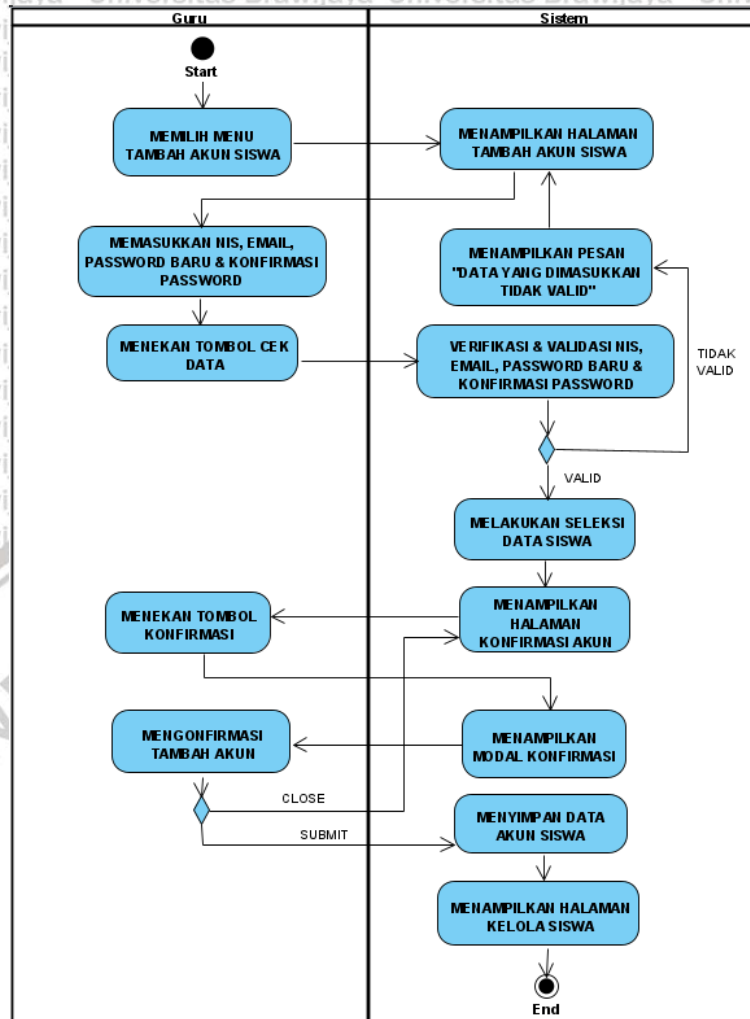
4.1.7.10 Activity Diagram Tambah Motivasi



Gambar 4.12 Activity Diagram Tambah Motivasi

Gambar 4.12 merupakan *activity diagram* dari fungsional tambah motivasi. Pada halaman pelanggaran siswa, aktor memilih menu edit motivasi dengan cara menekan tombol edit yang terdapat pada bagian motivasi dan penguatan. Sistem kemudian mengubah teks area yang tadinya berstatus *disabled* menjadi *fillable* agar dapat diisi oleh aktor. Aktor lalu mengisi teks area motivasi. Aktor kemudian dapat menyimpan maupun membatalkan aksi untuk menambahkan motivasi siswa tersebut. Jika aktor memilih untuk menyimpan, maka sistem akan menyimpan hasil dari masukan aktor pada teks area motivasi kemudian sistem akan menampilkan halaman pelanggaran siswa. Apabila aktor memilih untuk membatalkan pengisian motivasi, maka sistem tidak akan melakukan aksi apapun dan langsung menampilkan halaman pelanggaran siswa.

4.1.7.11 Activity Diagram Tambah Akun Siswa

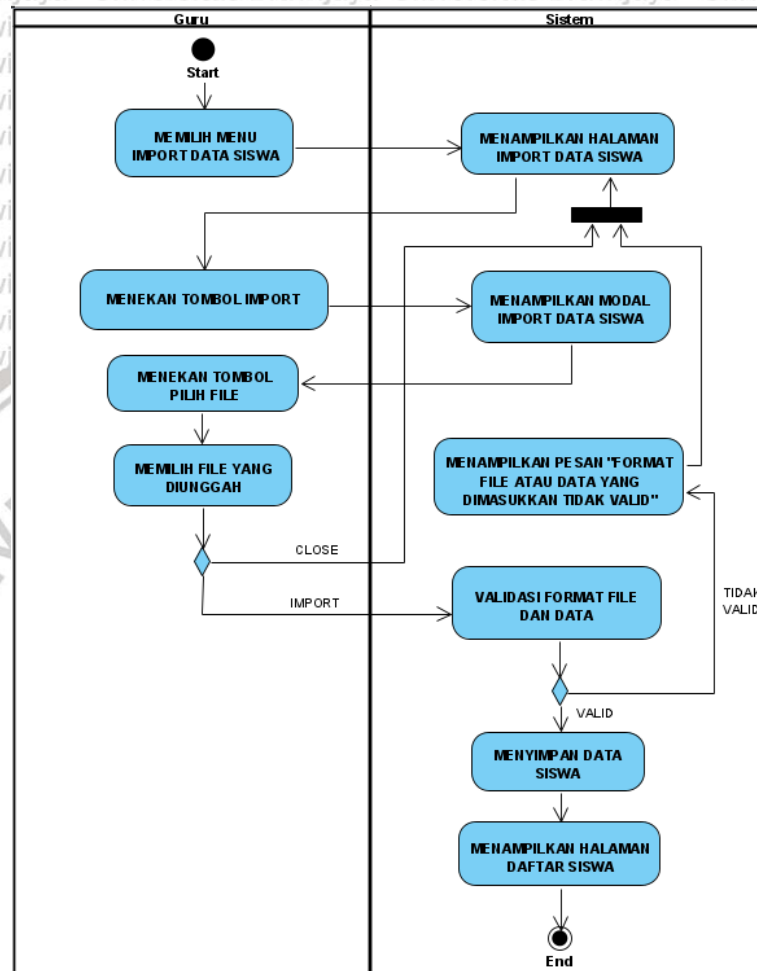


Gambar 4.13 Activity Diagram Tambah Akun Siswa

Gambar 4.13 merupakan *activity diagram* untuk menjalankan fungsional tambah akun siswa. Pada halaman kelola siswa, aktor memilih menu tambah akun siswa dengan cara menekan tombol tambah akun siswa yang terdapat pada halaman kelola siswa. Pada halaman tambah akun siswa, aktor mengisi informasi data akun sesuai dengan masing-masing edit teks yang terdapat pada halaman tambah akun siswa secara lengkap. Aktor lalu menekan tombol cek data. Sistem kemudian melakukan validasi terhadap hasil masukan aktor tersebut apakah sudah valid atau tidak. Jika tidak valid dan tidak terverifikasi, maka sistem akan menampilkan pesan bahwa informasi data yang dimasukkan tidak valid dan sistem tetap menampilkan halaman tambah akun siswa. Apabila masukan dari aktor valid dan terverifikasi, sistem akan menampilkan halaman konfirmasi akun. Sebelum menampilkan halaman konfirmasi akun, sistem terlebih dahulu melakukan seleksi terhadap data siswa yang akan ditambahkan akunnnya. Pada halaman konfirmasi akun, sistem menampilkan informasi data siswa dan data akun siswa yang akan dibuat. Aktor lalu menekan tombol konfirmasi diikuti dengan sistem menampilkan modal konfirmasi. Jika pada modal konfirmasi aktor memilih untuk menekan

tombol submit, sistem akan menyimpan data akun siswa ke *database* diikuti dengan sistem menampilkan halaman kelola siswa. Jika aktor menekan tombol *close* pada modal konfirmasi, sistem tidak akan melakukan aksi apapun dan kembali menampilkan halaman konfirmasi akun.

4.1.7.12 Activity Diagram Tambah Data Siswa

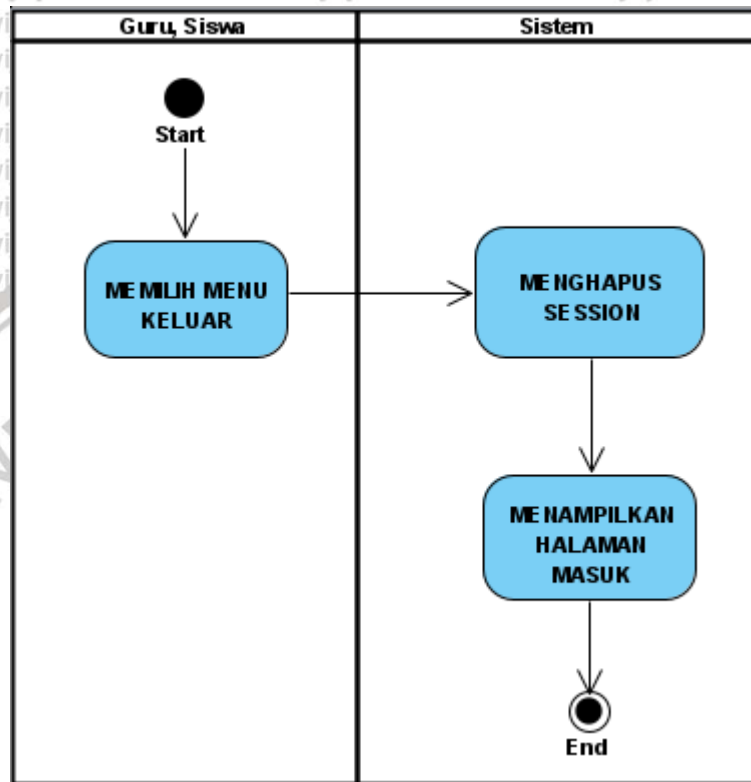


Gambar 4.14 Activity Diagram Tambah Data Siswa

Gambar 4.14 merupakan *activity diagram* untuk menjalankan fungsional tambah data siswa. Pada halaman kelola siswa, aktor memilih menu *import* data siswa dengan cara menekan tombol *import* data siswa yang terdapat pada halaman kelola siswa. Pada halaman *import* data siswa, aktor menekan tombol *import* diikuti dengan sistem menampilkan modal *import* data siswa. Aktor lalu menekan tombol pilih *file* diikuti dengan memilih *file* yang akan diunggah. Setelah aktor memilih *file* yang akan diunggah, aktor dapat memilih untuk menekan tombol *import* pada modal *import* data siswa atau memilih untuk menekan tombol *close* agar proses tambah data siswa dibatalkan. Jika aktor memilih menekan tombol *import* pada modal *import* data siswa, sistem akan melakukan validasi terhadap *file* yang diunggah tersebut apakah sudah valid atau tidak dari sisi format *file* maupun data pada *file* yang diunggah. Jika tidak valid, maka sistem akan menampilkan pesan bahwa format *file* atau data yang dimasukkan tidak valid dan

sistem kembali menampilkan halaman import data siswa. Apabila format *file* atau data yang berada di dalam *file* valid, sistem akan menyimpan data yang terdapat pada *file* yang diunggah ke dalam *database* lalu menampilkan halaman daftar siswa. Jika aktor menekan tombol *close* pada modal *import* data siswa, sistem tidak akan melakukan aksi apapun dan kembali menampilkan halaman *import* data siswa.

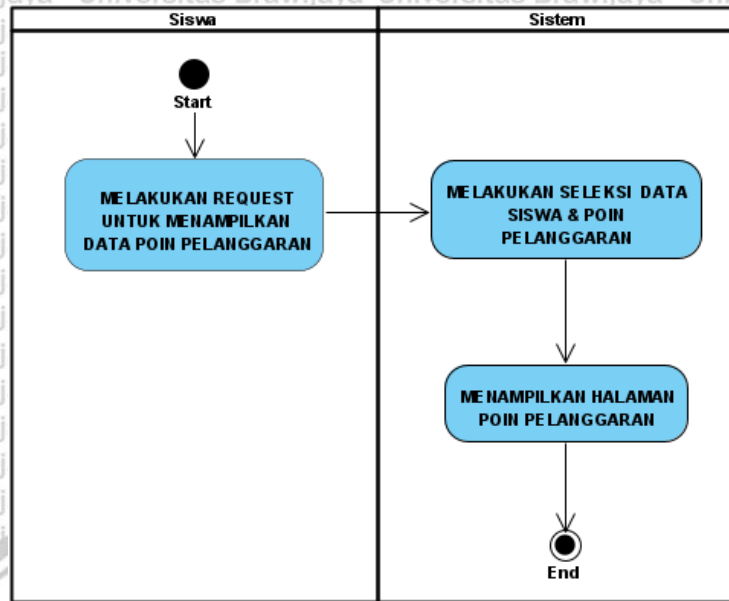
4.1.7.13 Activity Diagram Keluar



Gambar 4.15 Rancangan Activity Diagram Keluar

Gambar 4.15 merupakan gambaran *activity diagram* yang menjalankan fungsional keluar dari sistem untuk kembali menjadi aktor yang tidak dikenali oleh sistem. Aktor memilih menu keluar dengan cara menekan tombol keluar. Sistem kemudian memproses aksi keluar tersebut diikuti dengan menghapus *session* yang telah disimpan pada saat menjalankan proses masuk. Setelah berhasil melakukan proses keluar, sistem kemudian menampilkan halaman masuk sebagai tanda bahwa proses keluar telah berhasil.

4.1.7.14 Activity Diagram Lihat Pelanggaran



Gambar 4.16 Rancangan Activity Diagram Lihat Pelanggaran

Gambar 4.16 merupakan gambar *activity diagram* untuk menjalankan fungsional melihat pelanggaran yang telah diperoleh oleh siswa yang melakukan masuk. Aktor melakukan *request* kepada sistem untuk menampilkan informasi terkait data siswa dan poin pelanggaran siswa. Sistem kemudian melakukan seleksi terhadap data siswa dan poin pelanggaran siswa pada database yang sesuai dengan data aktor siswa yang melakukan masuk dan menampilkannya pada halaman sistem.

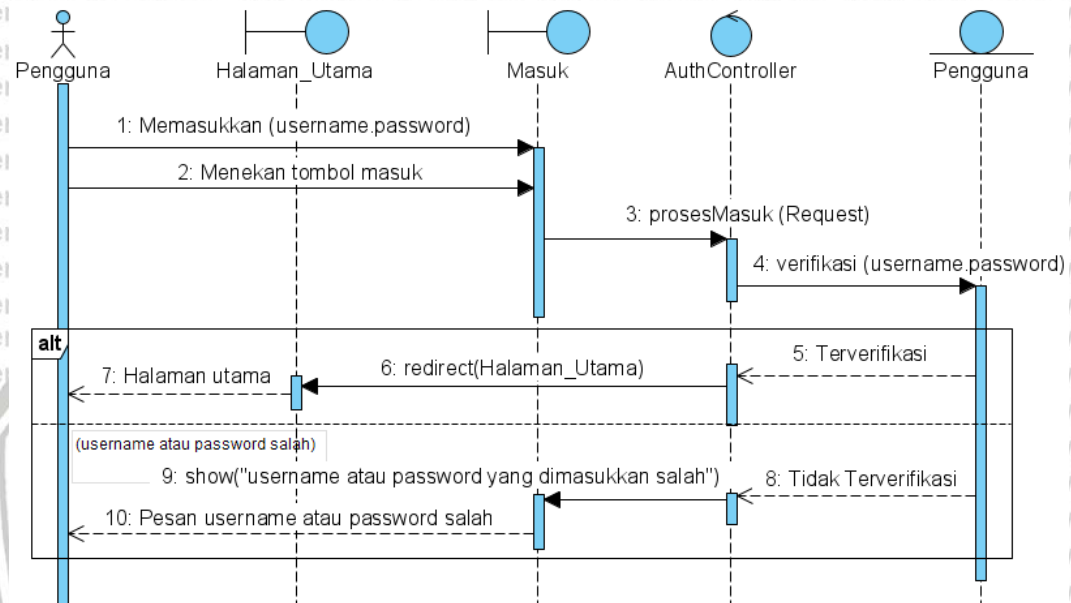


4.2 Hasil Perancangan Sistem

4.2.1 Sequence Diagram

Sequence diagram menjelaskan tentang bagaimana alur yang dijalankan oleh sistem mulai dari *user interface* yang ditampilkan hingga proses penyimpanan maupun validasi data yang terjadi pada *database*. Berikut merupakan *sequence diagram* dari aplikasi yang akan dibangun.

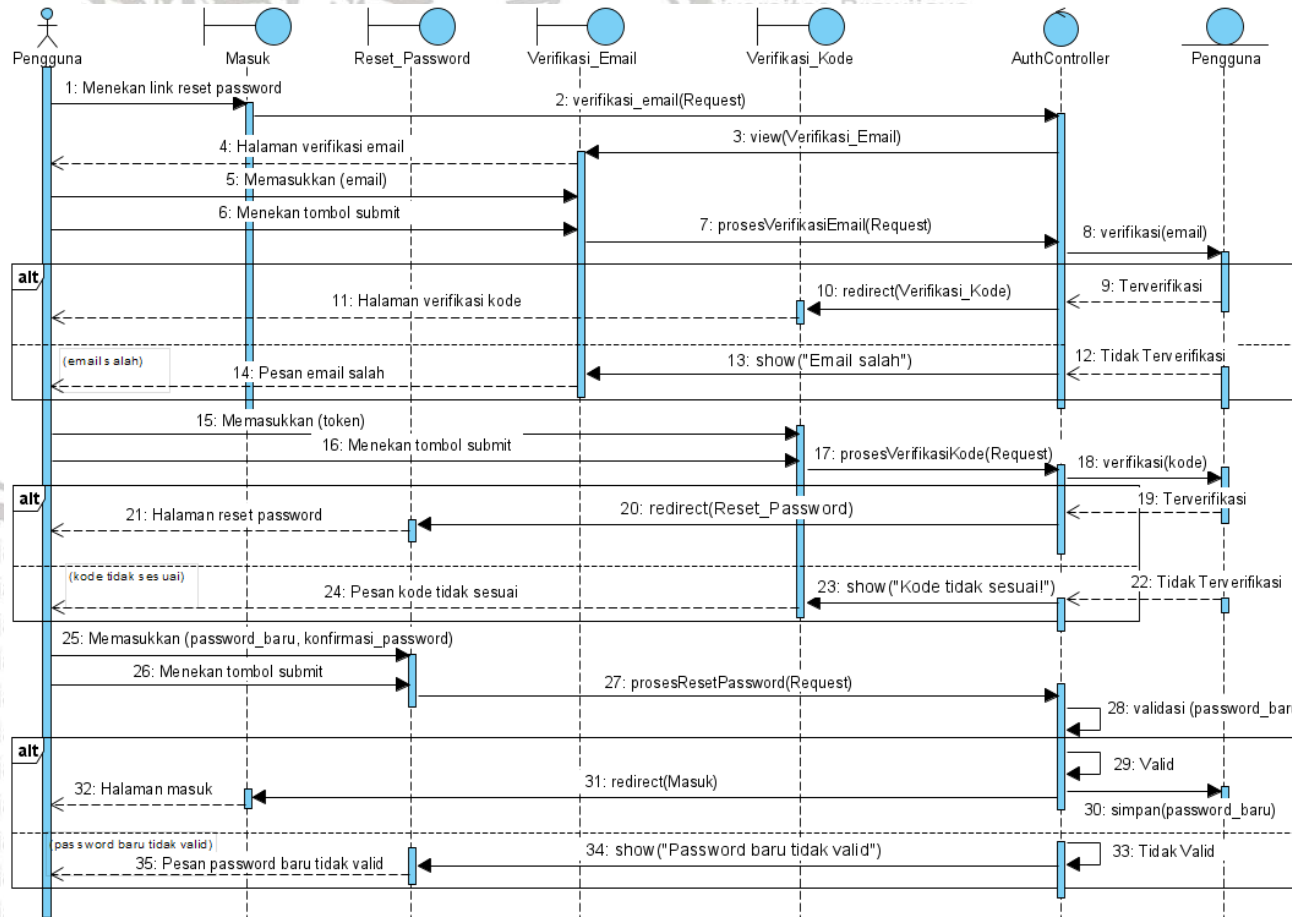
4.2.1.1 Sequence Diagram Masuk



Gambar 4.17 Sequence Diagram Masuk

Gambar 4.17 merupakan bentuk *sequence diagram* dari fungsional masuk. Proses awal dari fungsional masuk tersebut yaitu aktor terlebih dahulu memasukkan *username* dan *password* pada halaman masuk yang ditunjukkan pada *class* *Masuk*. Setelah aktor memasukkan *username* dan *password*, aktor kemudian menekan tombol masuk agar informasi masukan aktor dapat dikirimkan ke model bernama *Pengguna* untuk diperiksa dan divalidasi melalui *class* *AuthController* dengan menggunakan *Request* untuk mengambil data masukan aktor. Jika data masukan oleh aktor terverifikasi, sistem akan mengarahkan tampilan ke halaman utama aplikasi yang ditunjukkan pada *class* *Halaman_Utama*. Akan tetapi jika masukan dari aktor salah atau tidak terverifikasi oleh sistem, sistem akan memberikan peringatan bahwa *username* atau *password* salah serta mengembalikan tampilan sistem tetap pada halaman masuk yang ditunjukkan pada *class* *Masuk*.

4.2.1.2 Sequence Diagram Reset Password



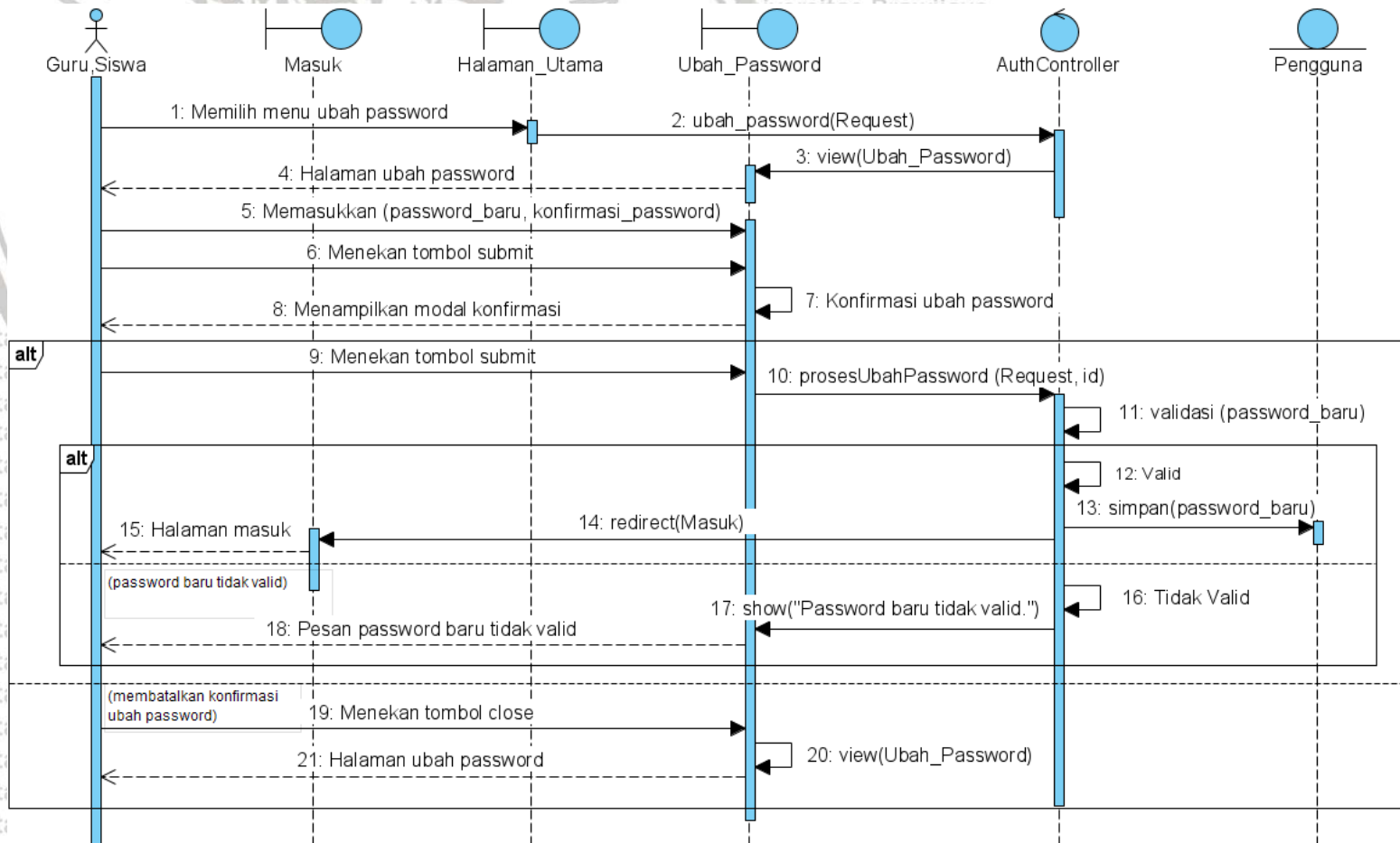
Gambar 4.18 Sequence Diagram Reset Password

Sequence diagram yang ditunjukkan pada Gambar 4.18 menjelaskan tentang alur proses dari fungsional reset *password*. Awalnya aktor mengakses halaman verifikasi *email* dengan menekan *link* reset *password* yang terdapat pada halaman masuk. Lalu sistem melalui *AuthController* akan mengarahkan halaman sistem untuk menampilkan halaman verifikasi email agar aktor dapat memasukkan email untuk mendapatkan kode konfirmasi yang ditunjukkan pada *class* bernama *Verifikasi_Email*. Sistem kemudian akan memverifikasi *email* yang telah dimasukkan oleh aktor dan memprosesnya pada *class* bernama *AuthController* dengan menggunakan *Request* untuk memproses masukan aktor. Jika sistem tidak dapat memverifikasi masukan *email* aktor, sistem akan memberikan peringatan dan mengembalikan pesan bahwa email yang dimasukkan salah serta mengembalikan tampilan sistem tetap pada halaman verifikasi *email* yang ditunjukkan pada *class* bernama *Verifikasi_Email*. Jika sistem dapat memverifikasi masukan *email* yang dimasukkan oleh aktor, sistem akan menampilkan halaman verifikasi kode yang ditunjukkan dengan *class* bernama *Verifikasi_Kode*.

Setelah mendapatkan kode konfirmasi, aktor memasukkan kode konfirmasi tersebut ke dalam edit teks kode yang tersedia pada halaman verifikasi kode. Sistem akan memverifikasi kode yang telah dimasukkan oleh aktor tersebut melalui model bernama *Pengguna*. Proses verifikasi kode tersebut dilakukan pada *class* bernama *AuthController* dengan menggunakan *Request* untuk memproses masukan aktor. Jika kode yang dimasukkan tidak sesuai, maka sistem akan mengembalikan pesan bahwa kode yang dimasukkan tidak sesuai dan sistem tetap menampilkan halaman verifikasi kode. Apabila kode yang dimasukkan aktor sesuai, sistem akan menampilkan halaman reset *password* yang ditunjukkan dengan *class* bernama *Reset_Password*.

Setelah verifikasi kode berhasil, aktor kemudian memasukkan *password* baru dan konfirmasi *password* pada halaman reset *password* yang ditunjukkan pada *class* *Reset_Password* lalu memprosesnya pada *class* *AuthController* dengan menggunakan *Request* untuk memproses masukan aktor. Pada *class* *AuthController*, sistem melakukan validasi terhadap masukan *password* baru aktor yang telah dimasukkan pada halaman reset *password* apakah masukan aktor valid atau tidak. Jika tidak valid, sistem akan mengembalikan pesan bahwa *password* baru yang telah dimasukkan tidak valid. Apabila masukan *password* baru aktor sesuai maka sistem akan menyimpan *password* baru pada model bernama *Pengguna*. Ketika telah berhasil melakukan reset *password*, tampilan sistem akan diarahkan ke halaman masuk yang ditunjukkan dengan *class* *Masuk*.

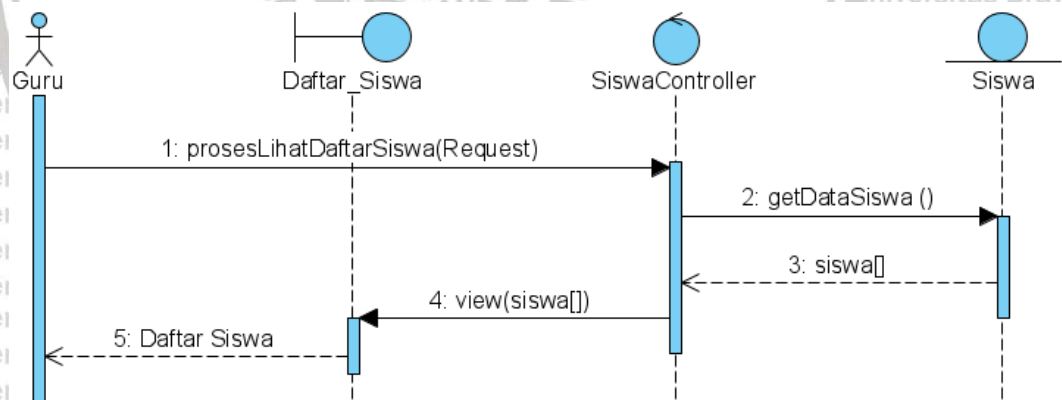
4.2.1.3 Sequence Diagram Ubah Password



Gambar 4.19 Sequence Diagram Ubah Password

Gambar 4.19 menunjukkan gambaran *sequence diagram* dari fungsional ubah *password*. Aktor berada pada halaman utama sistem diikuti dengan memilih menu ubah *password* dan ditampilkan oleh sistem melalui *class* bernama *AuthController*. Setelah halaman ubah *password* ditampilkan, aktor memasukkan *password* baru dan konfirmasi *password* diikuti dengan menekan tombol *submit* pada halaman ubah *password* yang ditunjukkan pada *class* *Ubah_Password*. Sistem kemudian menampilkan modal konfirmasi ubah *password* yang ditampilkan melalui *class* *Ubah_Password*. Jika pada modal konfirmasi ubah *password* aktor memilih untuk menekan tombol *submit*, sistem akan memproses hasil ubah *password* melalui *class* *AuthController* dengan menggunakan *Request* untuk memproses masukan aktor dan id untuk menyeleksi akun aktor yang melakukan ubah *password*. Pada *class* *AuthController*, sistem melakukan validasi terhadap masukan *password* baru dan konfirmasi *password* aktor apakah sudah valid atau tidak. Jika tidak valid, sistem akan memberikan informasi bahwa masukan *password* baru yang telah dimasukkan oleh aktor tidak valid. Apabila masukan *password* baru aktor valid, *password* baru yang telah dimasukkan oleh aktor akan disimpan ke model bernama *Pengguna*. Setelah berhasil menyimpan *password* baru, sistem kemudian menampilkan halaman masuk yang menandakan bahwa *password* berhasil diubah. Apabila pada modal konfirmasi aktor memilih untuk menekan tombol *close*, sistem tidak akan melakukan aksi apapun dan tetap menampilkan halaman ubah *password* yang ditunjukkan pada *class* *Ubah_Password*.

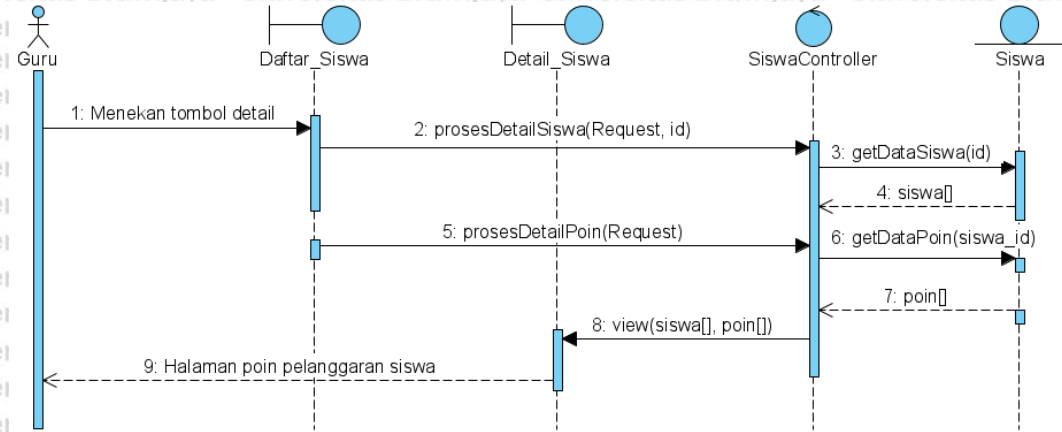
4.2.1.4 Sequence Diagram Lihat Daftar Siswa



Gambar 4.20 Sequence Diagram Lihat Daftar Siswa

Sequence diagram pada Gambar 4.20 menjelaskan bagaimana proses sistem untuk menampilkan seluruh daftar siswa yang tersimpan pada model bernama *Siswa*. Setelah melakukan proses masuk, aktor melakukan *request* kepada sistem melalui *class* *SiswaController* untuk mendapatkan data siswa dengan menggunakan *Request*. *Class* *SiswaController* kemudian melakukan *request* kepada model bernama *Siswa* untuk ditampilkan pada halaman daftar siswa yang ditunjukkan dengan *class* bernama *Daftar_Siswa*.

4.2.1.5 Sequence Diagram Lihat Pelanggaran Siswa

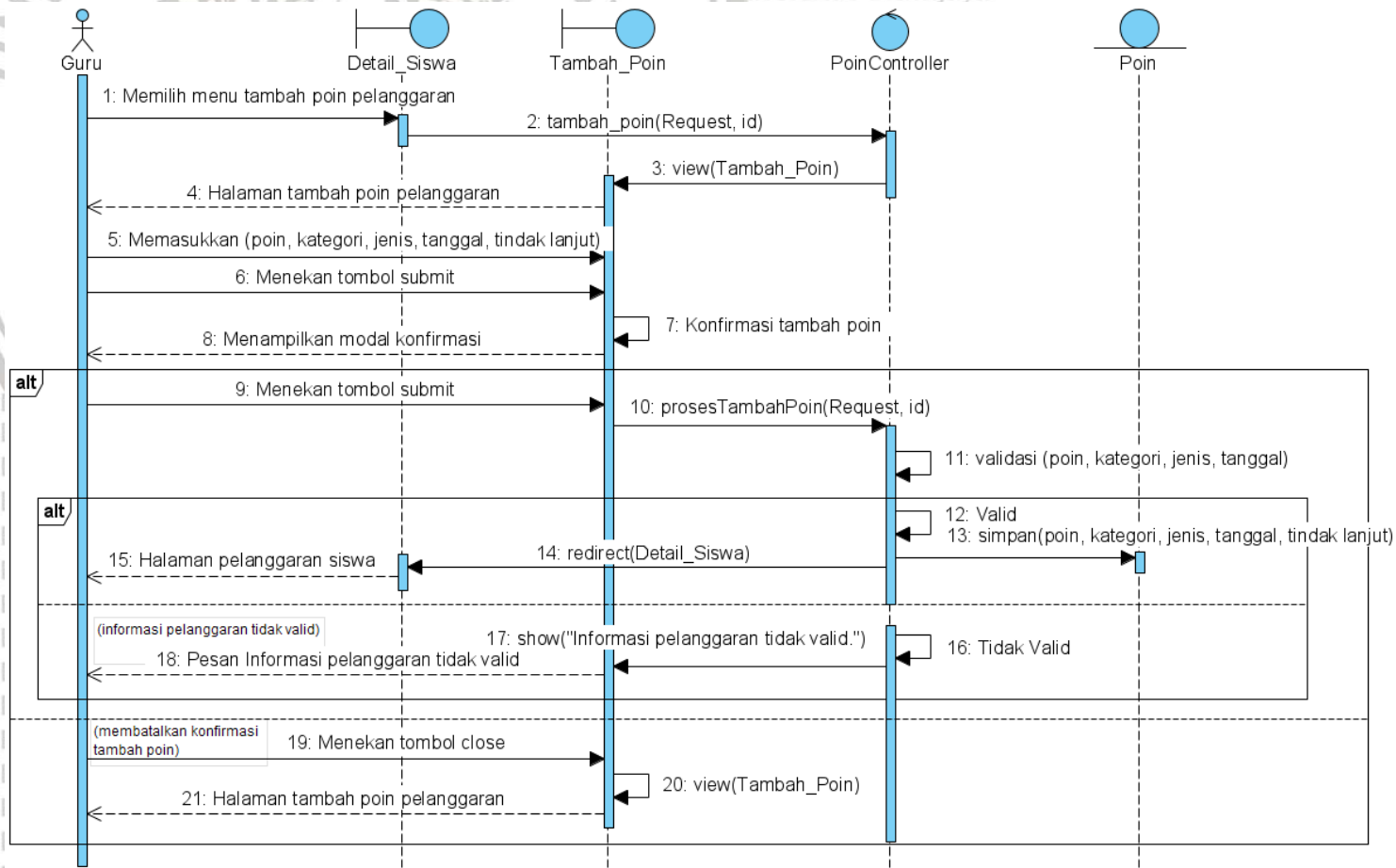


Gambar 4.21 Sequence Diagram Lihat Pelanggaran Siswa

Sequence diagram pada Gambar 4.21 menjelaskan terkait proses jalannya sistem ketika menampilkan pelanggaran dan informasi siswa. Sistem menampilkan data siswa sesuai id dan poin pelanggaran siswa sesuai siswa_id dengan melakukan request melalui class SiswaController ke model bernama Siswa. Setelah melakukan request, data yang diperoleh akan ditampilkan pada halaman detail siswa yang ditunjukkan pada class bernama Detail_Siswa.



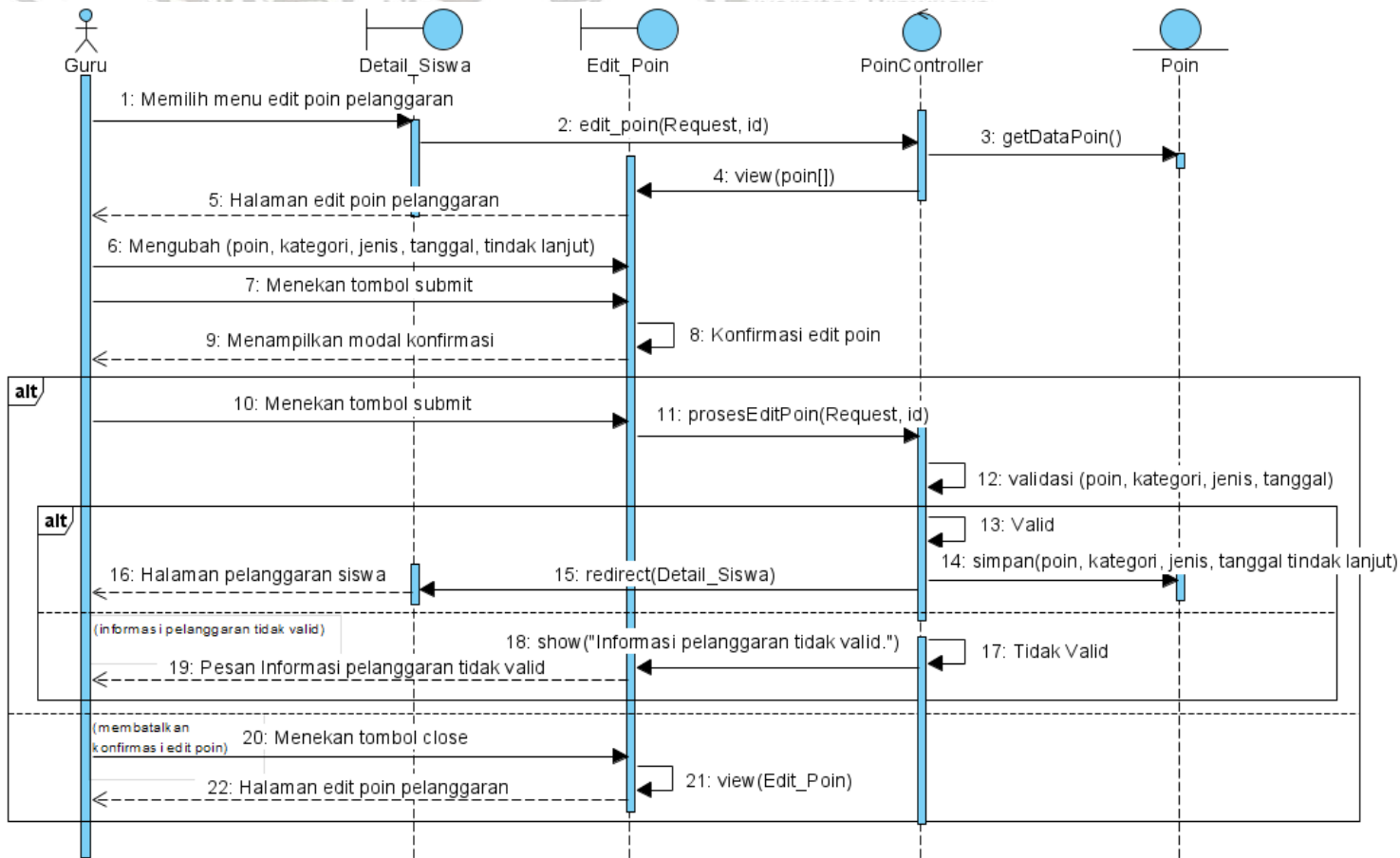
4.2.1.6 Sequence Diagram Tambah Poin Pelanggaran



Gambar 4.22 Sequence Diagram Tambah Poin Pelanggaran

Sequence diagram pada Gambar 4.22 menjelaskan bagaimana proses penambahan poin pelanggaran siswa. Aktor melakukan *request* kepada sistem untuk menampilkan halaman tambah poin dengan menekan tombol tambah poin pelanggaran. Lalu *request* dari aktor diproses melalui *class* PoinController untuk menampilkan halaman tambah poin. Setelah halaman tambah poin ditampilkan, aktor lalu memasukkan poin, kategori, jenis pelanggaran, dan tanggal melalui halaman tambah poin yang ditunjukkan pada *class* Tambah_Poin diikuti dengan menekan tombol *submit*. Setelah aktor menekan tombol *submit*, sistem kemudian menampilkan modal konfirmasi tambah poin yang ditunjukkan pada *class* Tambah_Poin. Jika pada modal konfirmasi tambah poin aktor menekan tombol *submit*, sistem akan memproses hasil masukan aktor yang ditunjukkan pada *class* PoinController menggunakan *Request* dan sistem melakukan validasi apakah informasi masukan tambah poin yang dimasukkan aktor valid atau tidak. Jika valid, informasi poin yang ditambahkan akan disimpan di model bernama Poin diikuti dengan informasi bahwa poin telah berhasil ditambahkan dan sistem menampilkan halaman pelanggaran siswa. Jika masukan aktor tidak valid, sistem akan mengembalikan informasi bahwa masukan tambah poin yang telah dimasukkan tidak valid dan sistem tetap menampilkan halaman tambah poin yang ditunjukkan dengan *class* Tambah_Poin. Jika pada modal konfirmasi tambah poin aktor memilih untuk menekan tombol *close*, sistem tidak akan melakukan aksi apapun dan tetap menampilkan halaman tambah poin pelanggaran yang ditunjukkan pada *class* Tambah_Poin.

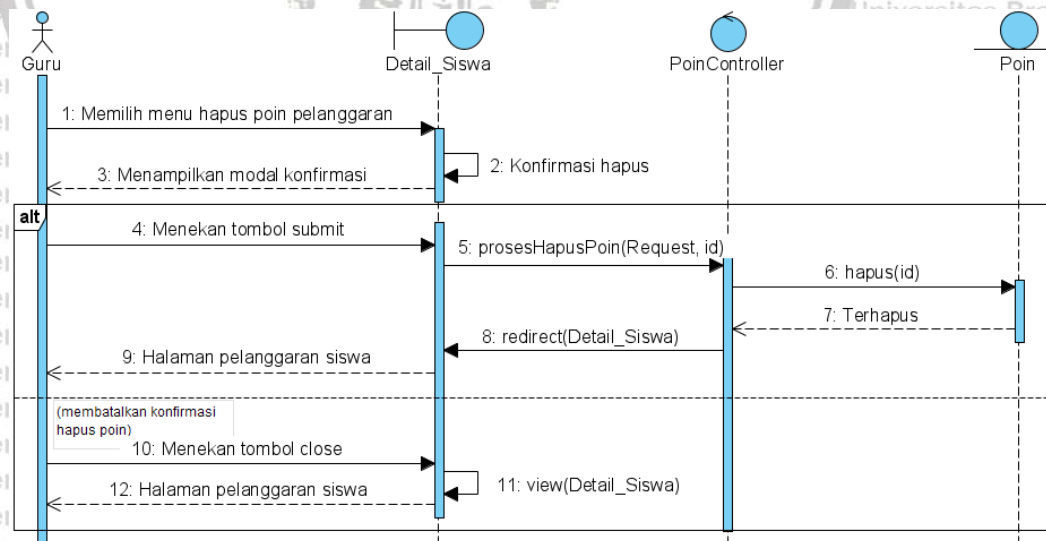
4.2.1.7 Sequence Diagram Edit Poin Pelanggaran



Gambar 4.23 Sequence Diagram Edit Poin Pelanggaran

Gambar 4.23 merupakan gambar dari *sequence diagram* yang menjelaskan bagaimana proses dari fungsional edit poin pelanggaran siswa. Aktor melakukan *request* kepada sistem untuk dapat menampilkan halaman edit poin dengan cara menekan memilih menu edit poin. Setelah memilih menu edit poin, sistem kemudian melakukan *request* data dengan *attribute* poin, kategori, jenis, dan tanggal yang dilakukan pada *control class* PoinController kepada model bernama Poin untuk ditampilkan pada masing-masing edit teks di halaman edit poin pelanggaran siswa. Setelah halaman edit poin dan data poin pelanggaran telah ditampilkan pada masing-masing edit teks di halaman edit poin pelanggaran siswa, aktor dapat mengubah masing-masing data tersebut lalu diikuti dengan menekan tombol *submit*. Sistem lalu menampilkan modal konfirmasi edit poin pelanggaran yang ditunjukkan pada *class* Edit_Poin. Jika pada modal konfirmasi edit poin pelanggaran aktor memilih untuk menekan tombol *submit*, sistem akan memproses hasil dari perubahan yang dilakukan oleh aktor yang ditunjukkan pada *class* PoinController menggunakan *Request* untuk mendapatkan masukan aktor dan sistem akan melakukan validasi masukan aktor apakah valid atau tidak. Jika masukan dari aktor valid, sistem akan menyimpan hasil perubahan yang telah dilakukan oleh aktor ke *database* melalui model Poin. Apabila hasil perubahan dari aktor tidak valid, sistem akan menampilkan pesan bahwa informasi poin pelanggaran yang telah dimasukkan oleh aktor tidak valid dan sistem akan tetap menampilkan halaman pelanggaran siswa. Jika pada modal konfirmasi edit poin pelanggaran aktor memilih untuk menekan tombol *close*, sistem tidak akan melakukan aksi apapun dan tetap menampilkan halaman edit poin pelanggaran yang ditunjukkan pada *class* Edit_Poin.

4.2.1.8 Sequence Diagram Hapus Poin Pelanggaran

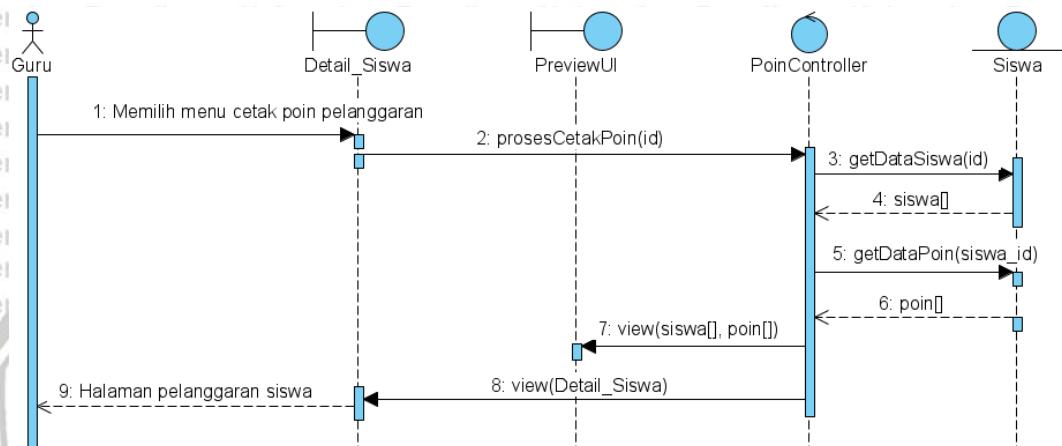


Gambar 4.24 Sequence Diagram Hapus Poin Pelanggaran

Gambar 4.24 merupakan bentuk dari *sequence diagram* fungsional hapus poin pelanggaran siswa. Aktor memilih poin pelanggaran yang akan dihapus terlebih dahulu melalui halaman detail siswa dengan cara menekan menu hapus

poin yang ditunjukkan pada *class* Detail_Siswa. Sistem kemudian menampilkan modal konfirmasi hapus poin pelanggaran. Jika aktor menekan tombol submit, sistem akan memproses poin yang telah dipilih berdasarkan id pada *class* prosesHapusPoin dan melakukan penghapusan data poin pelanggaran yang sesuai dengan id dari *database* melalui model bernama Poin. Setelah berhasil melakukan penghapusan data pada *database*, sistem memberikan informasi bahwa poin pelanggaran telah berhasil dihapus. Jika aktor menekan tombol *close*, sistem tidak akan melakukan aksi hapus poin dan langsung menampilkan halaman pelanggaran siswa kembali yang ditunjukkan pada *class* Detail_Siswa.

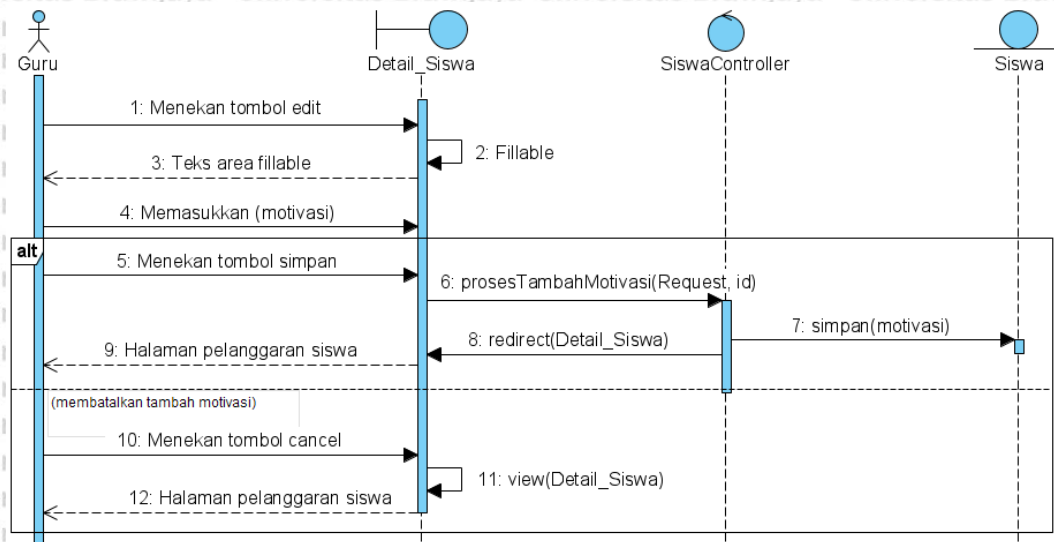
4.2.1.9 Sequence Diagram Cetak Poin Pelanggaran



Gambar 4.25 Sequence Diagram Cetak Poin Pelanggaran

Sequence diagram pada Gambar 4.25 menunjukkan fungsional dari cetak poin pelanggaran siswa. Aktor memilih menu cetak poin pelanggaran siswa melalui halaman pelanggaran siswa yang ditunjukkan pada *class* Detail_Siswa. Sistem kemudian melakukan *request* data siswa sesuai id dan poin pelanggaran siswa sesuai siswa_id melalui PoinController terhadap model bernama Siswa untuk ditampilkan pada dokumen yang akan dicetak. Sistem lalu menampilkan preview dari dokumen yang dapat dicetak tersebut yang ditandai dengan *class* bernama PreviewUI diikuti dengan menampilkan kembali halaman pelanggaran siswa.

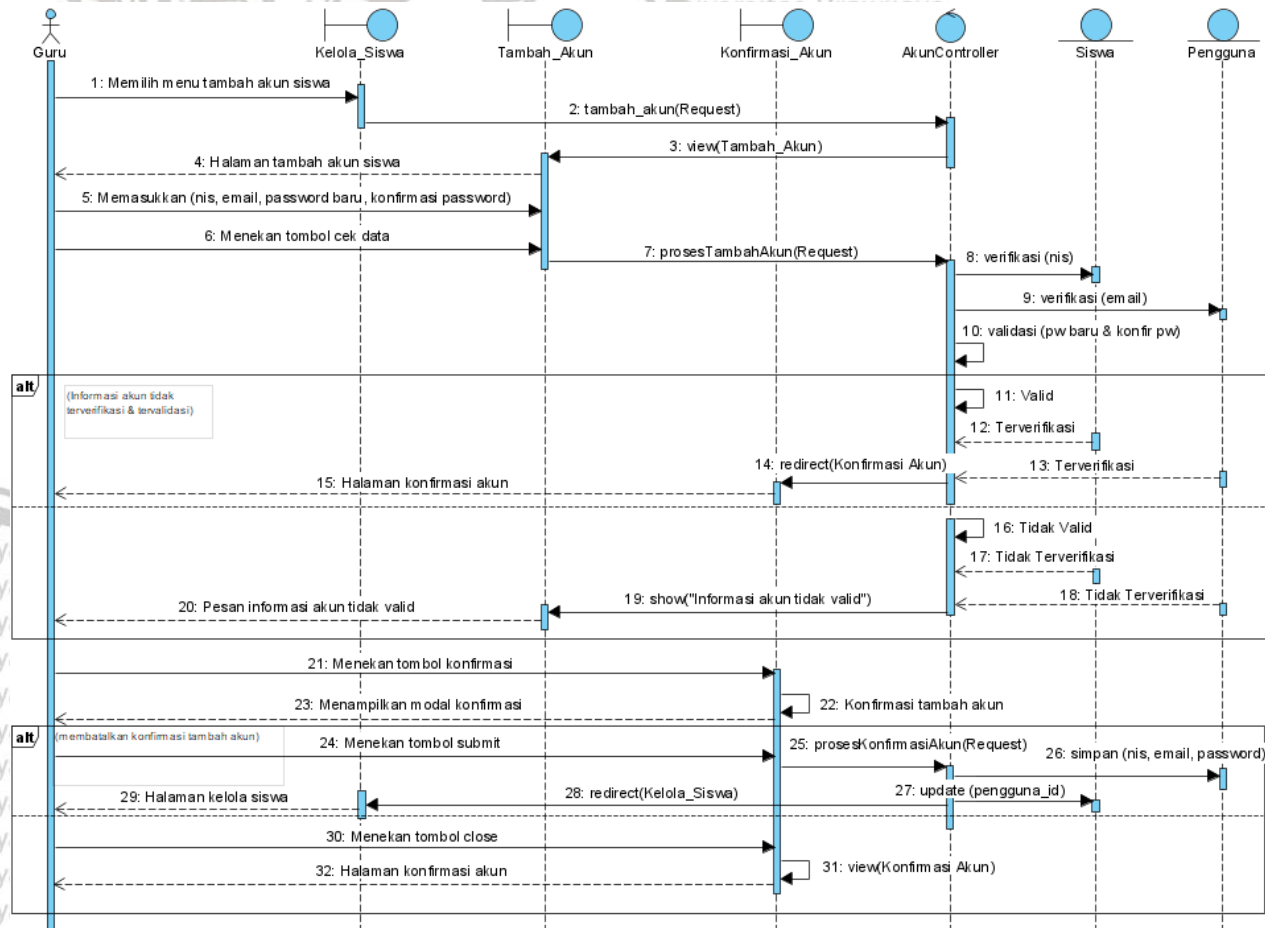
4.2.1.10 Sequence Diagram Tambah Motivasi



Gambar 4.26 Sequence Diagram Tambah Motivasi

Gambar 4.26 merupakan *sequence diagram* dari fungsional tambah motivasi. Aktor menekan tombol edit pada bagian motivasi dan penguatan di halaman pelanggaran siswa yang ditunjukkan pada *class* *Detail_Siswa*. Sistem kemudian mengubah teks area motivasi yang awalnya *disabled* menjadi *fillable* agar dapat diisi oleh aktor diikuti aktor memasukkan motivasi ke dalam teks area tersebut. Setelah memasukkan motivasi, aktor dapat menekan tombol simpan agar motivasi yang telah diisi dapat diproses oleh sistem melalui *class* *SiswaController* atau menekan tombol *cancel* untuk membatalkan aksi tambah motivasi. Jika aktor memilih untuk menekan tombol simpan, motivasi yang telah ditambahkan akan disimpan di *database* melalui model bernama *Motivasi* diikuti dengan informasi bahwa motivasi berhasil disimpan dan sistem menampilkan halaman pelanggaran siswa. Apabila aktor memilih untuk menekan tombol *cancel*, sistem tidak akan melakukan aksi apapun dan kembali menampilkan halaman pelanggaran siswa yang ditunjukkan pada *class* *Detail_Siswa*.

4.2.1.11 Sequence Diagram Tambah Akun Siswa



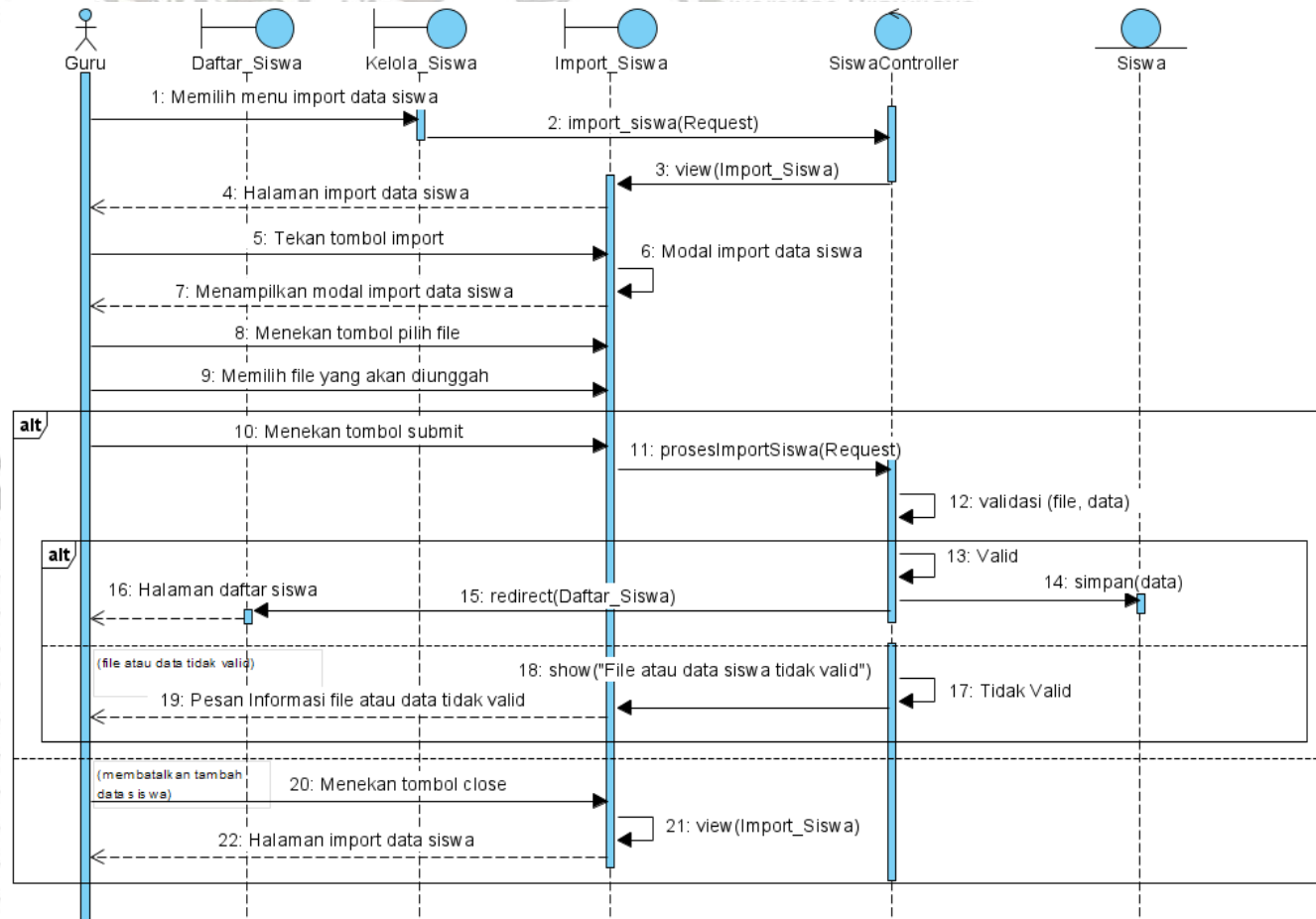
Gambar 4.27 Sequence Diagram Tambah Akun Siswa



Sequence diagram pada Gambar 4.27 menjelaskan bagaimana proses untuk menambahkan akun siswa agar siswa dapat melakukan masuk ke dalam sistem. Aktor melakukan *request* kepada sistem untuk menampilkan halaman tambah akun siswa dengan cara menekan tambah akun siswa. Lalu *request* dari aktor diproses melalui *class* AkunController untuk menampilkan halaman tambah akun siswa. Setelah halaman tambah akun siswa ditampilkan, aktor lalu memasukkan NIS, email, *password* baru, dan konfirmasi *password* melalui halaman tambah akun yang ditunjukkan pada *class* Tambah_Akun diikuti dengan menekan tombol cek data. Setelah aktor menekan tombol cek data, sistem kemudian melakukan validasi dan verifikasi data dari masing-masing masukan aktor tersebut melalui *class* AkunController. Jika masukan aktor tidak valid atau NIS dan *email* tidak terverifikasi oleh sistem, sistem akan mengembalikan pesan bahwa masukan informasi akun siswa tidak valid. Apabila masukan aktor dapat diverifikasi dan divalidasi oleh sistem, sistem akan menampilkan halaman konfirmasi akun beserta data akun siswa yang akan ditambahkan yang ditunjukkan pada *class* Konfirmasi_Akun.

Pada halaman konfirmasi akun, aktor kemudian menekan tombol konfirmasi diikuti dengan sistem menampilkan modal konfirmasi konfirmasi tambah akun yang ditunjukkan pada *class* Konfirmasi_Akun. Jika pada modal konfirmasi tambah akun aktor menekan tombol *submit*, sistem akan memproses hasil masukan aktor yang ditunjukkan pada *class* AkunController menggunakan *Request* dan sistem akan menyimpan informasi tambah akun siswa pada *database* melalui model bernama Pengguna serta melakukan *update* *pengguna_id* pada model bernama Siswa. Jika pada modal konfirmasi tambah akun aktor memilih untuk menekan tombol *close*, sistem tidak akan melakukan aksi apapun dan tetap menampilkan halaman konfirmasi akun yang ditunjukkan pada *class* Konfirmasi_Akun.

4.2.1.12 Sequence Diagram Tambah Data Siswa

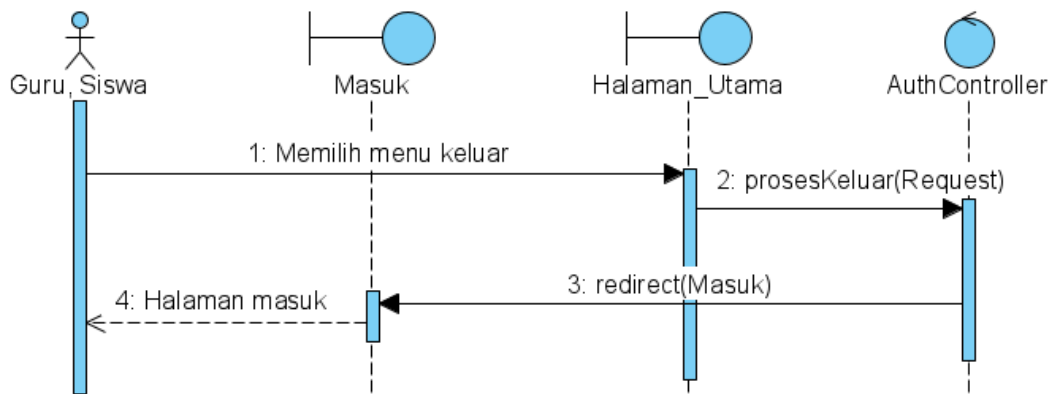


Gambar 4.28 Sequence Diagram Tambah Data Siswa

Sequence diagram pada Gambar 4.28 menjelaskan bagaimana proses tambah data siswa. Aktor melakukan *request* kepada sistem untuk menampilkan halaman *import* data siswa dengan cara menekan tombol *import* data siswa. Lalu *request* dari aktor tersebut diproses melalui *class* SiswaController untuk menampilkan halaman *import* data siswa. Setelah halaman *import* data siswa ditampilkan, aktor lalu menekan tombol *import* diikuti dengan sistem menampilkan modal *import* data yang ditunjukkan pada *class* Import_Siswa. Setelah modal *import* data ditampilkan, aktor kemudian menekan tombol pilih *file* dan memilih *file* yang akan diunggah. Setelah mengunggah *file*, aktor dapat menekan tombol *submit* agar *file* yang telah diunggah segera diproses oleh sistem atau menekan tombol *close* jika aktor ingin membatalkan proses tambah data siswa. Jika aktor memilih untuk menekan tombol *submit*, sistem akan memproses *file* yang telah diunggah melalui *class* SiswaController menggunakan *Request* dan sistem melakukan validasi apakah *file* atau data yang dimasukkan aktor valid atau tidak. Jika valid, data pada *file* yang telah diunggah akan disimpan di *database* melalui model bernama Siswa diikuti dengan informasi bahwa data siswa telah berhasil ditambahkan dan sistem menampilkan halaman daftar siswa yang ditunjukkan pada *class* Daftar_Siswa. Jika *file* atau data yang telah diunggah aktor tidak valid, sistem akan mengembalikan informasi bahwa *file* atau data yang telah diunggah tidak valid dan sistem tetap menampilkan halaman *import* data siswa yang ditunjukkan dengan *class* Import_Siswa. Jika pada modal *import* data aktor memilih untuk menekan tombol *close*, sistem tidak akan melakukan aksi apapun dan tetap menampilkan halaman *import* data siswa yang ditunjukkan pada *class* Import_Siswa.



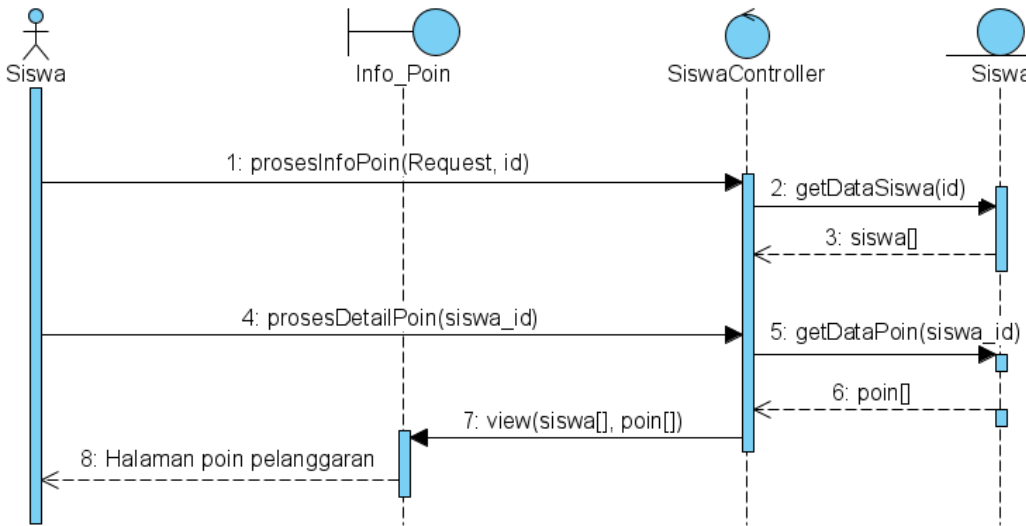
4.2.1.13 Sequence Diagram Keluar



Gambar 4.29 Sequence Diagram Keluar

Sequence diagram pada Gambar 4.29 menjelaskan terkait fungsional keluar. Aktor melakukan proses keluar melalui class AuthController dengan menggunakan Request untuk menghapus session yang tersimpan. Sistem lalu menampilkan halaman masuk yang ditunjukkan pada class masuk dan menandakan bahwa proses keluar telah berhasil.

4.2.1.14 Sequence Diagram Lihat Pelanggaran

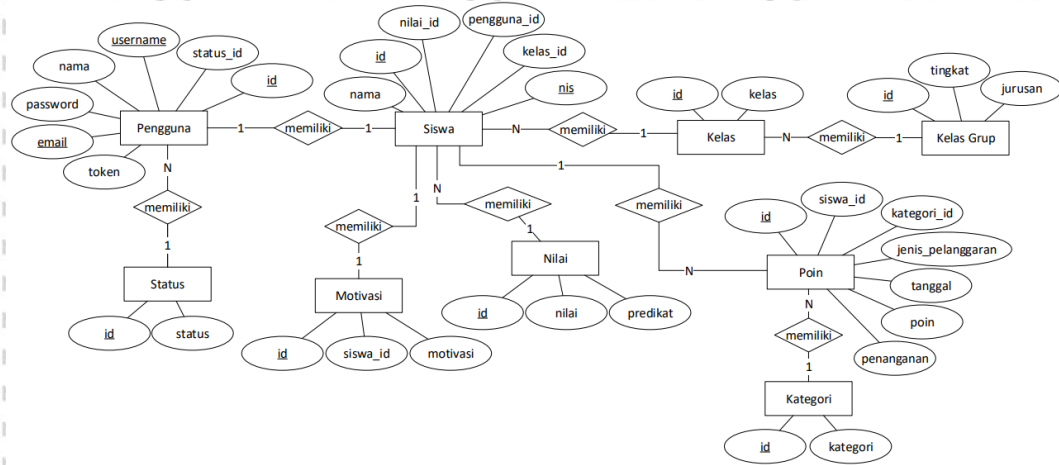


Gambar 4.30 Sequence Diagram Lihat Pelanggaran

Sequence diagram pada Gambar 4.30 menjelaskan terkait fungsional lihat pelanggaran dari aktor siswa. Pada saat sistem telah mengenali status pengguna adalah siswa sebagai aktor, sistem lalu melakukan request melalui class SiswaController untuk mengambil data siswa berdasarkan id dan poin pelanggaran siswa berdasarkan siswa_id sesuai dengan aktor yang telah dikenali oleh sistem tersebut. Setelah mendapatkan data siswa dan poin pelanggaran, sistem

kemudian menampilkan data siswa dan poin pelanggaran tersebut pada halaman poin pelanggaran yang ditunjukkan pada *class* bernama Info_Poin.

4.2.2 Entity Relationship Diagram (ERD)

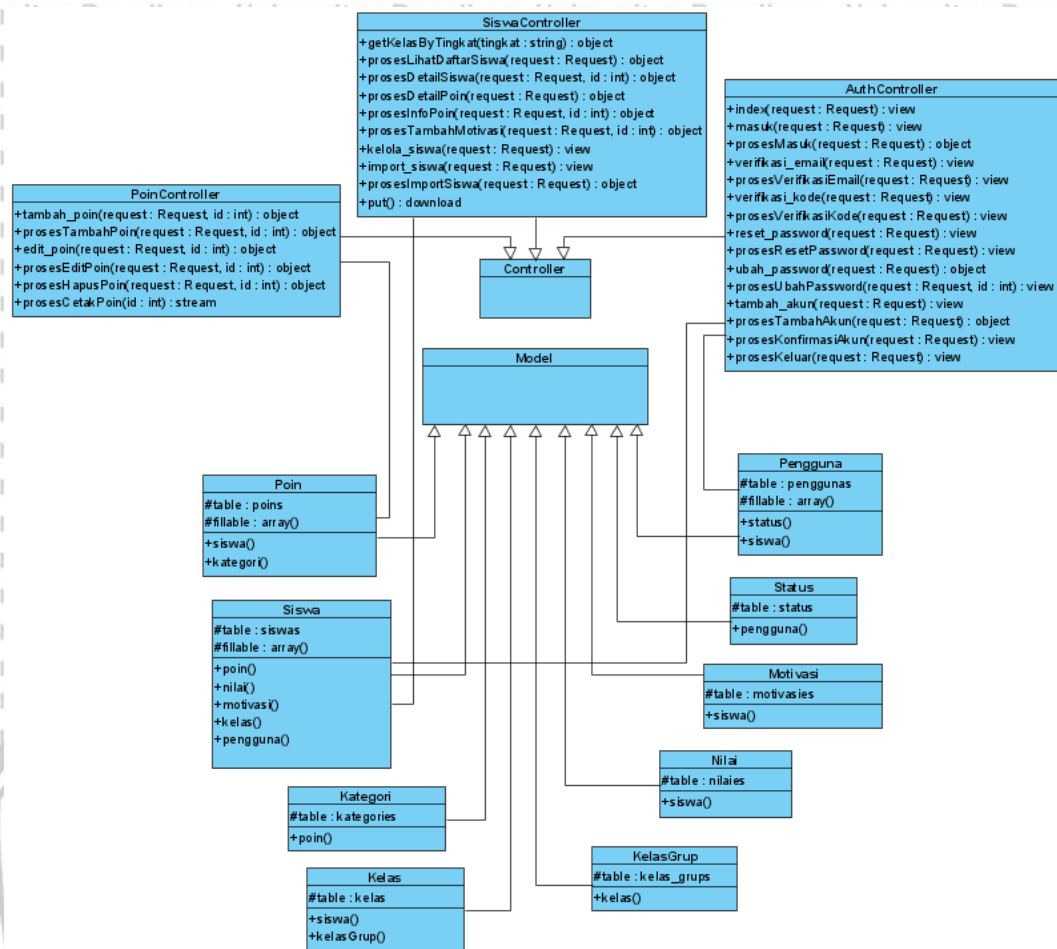


Gambar 4.31 Entity Relationship Diagram (ERD)

Gambar 4.31 menunjukkan terkait gambaran dari *Entity Relationship Diagram* atau yang biasa disebut dengan ERD aplikasi SiTalang. Pada gambar tersebut dijelaskan hubungan antar entitas yang terdapat pada *database* disertai dengan penjelasan kardinalitas dari masing-masing relasi antar entitas. Beberapa kardinalitas yang terdapat pada ERD Gambar 4.31 tersebut menjelaskan seperti *one to many* dan *one to one*. Terdapat sembilan entitas pada aplikasi SiTalang yang dibangun seperti status, pengguna, siswa, kelas, kelas grup, motivasi, nilai, poin, dan kategori.

4.2.3 Class Diagram

Class diagram didefinisikan menjadi dua jenis *class* yaitu *model* dan *controller*. Masing-masing dari *class* tersebut menginterpretasikan peran sebagai jembatan antara *view* dengan *database* (*controller*) dan berperan untuk mengatur data pada program yaitu *model*. Gambar 4.32 merupakan *class diagram* dari aplikasi yang dibangun.



Gambar 4.32 Class Diagram

Gambar 4.32 merupakan gambaran dari *class diagram* dari aplikasi yang dibangun. Pada gambar tersebut terdapat beberapa *class* yang diantaranya mempunyai hubungan *Generalization* atau turunan dari *class Controller* dan *Model*. *Class* turunan dari *Controller* memiliki hubungan dengan beberapa *class* turunan dari *Model* yang digambarkan dengan *Association*.

4.2.4 Pseudocode

Pembuatan *pseudocode* bertujuan untuk agar penulisan atau alur dari program yang akan dikembangkan dapat mudah dipahami dikarenakan penulisan dari *pseudocode* menggunakan bahasa yang mudah dipahami oleh manusia. Berikut merupakan penulisan *pseudocode* dari masing-masing fungsi atau *method* tiap-tiap fungsional yang dikembangkan pada sistem.

4.2.4.1 Pseudocode Fungsional Masuk

Tabel 4.22 merupakan *pseudocode* dari fungsional masuk pada sistem yang dikembangkan.

Tabel 4.22 Pseudocode Masuk

Pseudocode Masuk	
1	Mulai
2	validate(input(username, password))
3	\$username, \$password = input(username, password)
4	\$pengguna = pengguna(username, username)->first();
5	if(\$pengguna != null && \$pengguna->status = guru)then
6	if(check \$password: true)
7	create session
8	redirect daftar siswa
9	else
10	redirect masuk
11	else if (\$pengguna != null && \$pengguna->status = siswa)then
12	if(check \$password: true)
13	create session
14	redirect pelanggaran siswa
15	else
16	redirect masuk
17	else
18	redirect masuk
19	endif

4.2.4.2 Pseudocode Fungsional Reset Password

Tabel 4.23 merupakan *pseudocode* dari fungsional proses reset *password* pada sistem yang dikembangkan. Pada proses reset *password* ini dibutuhkan beberapa langkah seperti verifikasi *email* dan verifikasi kode sehingga penulisan dari *pseudocode* dari masing-masing fungsi tersebut ditunjukkan pada Tabel 4.23, Tabel 4.24, Tabel 4.25.

Tabel 4.23 Pseudocode Verifikasi Email

Pseudocode Verifikasi Email	
1	Mulai
2	validate(input(email))
3	\$email= input(email)
4	\$pengguna = pengguna(email, email)->first();
5	if(\$pengguna != null)then
6	\$nama = \$pengguna->nama
7	\$kode = encrypt(string)
8	\$pengguna->update(token)->\$kode
9	mail(\$pengguna->nama)
10	create session(log kode)
11	redirect verifikasi kode

Tabel 4.23 Pseudocode Verifikasi Email (Lanjutan)

Pseudocode Verifikasi Email	
12	else
13	redirect verifikasi_email
14	endif

Tabel 4.24 Pseudocode Verifikasi Kode

Pseudocode Verifikasi Kode	
1	Mulai
2	validate(input(kode))
3	\$kode = input(kode)
4	\$email = session(log_kode)
5	\$pengguna = pengguna(email, email)->first();
6	\$token = \$pengguna->token
7	\$dekrip = decrypt(\$token)
8	\$pengguna->token->update(\$dekrip)
9	\$pengguna->save
10	if(\$pengguna != null && \$pengguna->token = \$kode)then
11	\$enkrip = encrypt(\$token)
12	\$pengguna->token->update(\$enkrip)
13	create session(log_reset_password)
14	redirect reset_password
15	else
16	\$enkrip = encrypt(\$token)
17	\$pengguna->token->update(\$enkrip)
18	redirect verifikasi_kode
19	endif

Tabel 4.25 Pseudocode Reset Password

Pseudocode Reset Password	
1	Mulai
2	validate(input(password_baru))
3	\$email = session(log_kode)
4	\$password_baru = input(password_baru)
5	pengguna(email, \$email)->password->update(\$password_baru)
6	redirect masuk

4.2.4.3 Pseudocode Fungsional Ubah Password

Tabel 4.26 merupakan *pseudocode* dari fungsional proses ubah *password* pada sistem yang dikembangkan.

Tabel 4.26 Pseudocode Ubah Password

Pseudocode Ubah Password	
1	Mulai
2	validate(input(konfirmasi_password))
3	\$konfirmasi_password = input(konfirmasi_password)
4	\$pengguna = pengguna(id, id)->first();
5	\$pengguna->password->update(\$konfirmasi_password)
6	redirect masuk

4.2.4.4 Pseudocode Fungsional Lihat Daftar Siswa

Tabel 4.27 merupakan *pseudocode* dari fungsional lihat daftar siswa pada sistem yang dikembangkan.

Tabel 4.27 Pseudocode Lihat Daftar Siswa

Pseudocode Daftar Siswa	
1	Mulai
2	if (ajax) then
3	if (!empty(input(kelas))) then
4	\$data = siswa(kelas_id, input(kelas))->get()
5	else
6	\$data = siswa->get()
7	endif
8	return datatables
9	endif
10	\$siswa = siswa->get();
12	return view(daftar_siswa, \$siswa)

4.2.4.5 Pseudocode Fungsional Lihat Pelanggaran Siswa

Pada proses melihat pelanggaran siswa, terdapat dua proses yang dijalankan oleh sistem. Proses yang pertama yaitu untuk menampilkan data profil dari siswa, lalu proses yang kedua yaitu untuk menampilkan data poin pelanggaran siswa yang dipilih. Masing-masing dari proses tersebut dijelaskan pada *pseudocode* yang ditunjukkan pada Tabel 4.28 dan Tabel 4.29.

Tabel 4.28 Pseudocode Profil Siswa

Pseudocode Profil Siswa	
1	Mulai
2	\$siswa = siswa(id, id)->first()
3	\$total_poin = siswa(sum(poin))
4	\$kategori = kategori()->get()
5	if(total_poin = 0)then
6	\$siswa->nilai_id->update(1)
7	else if(\$total_poin > 0 && \$total_poin <= 25)
8	\$siswa->nilai_id->update(2)
9	else if(\$total_poin > 25 && \$total_poin <= 50)
10	\$siswa->nilai_id->update(3)
11	else if(\$total_poin > 50 && \$total_poin <= 75)
12	\$siswa->nilai_id->update(4)
13	else if(\$total_poin > 75)
14	\$siswa->nilai_id->update(5)
15	return view(detail_siswa, \$siswa, \$kategori)

Tabel 4.29 Pseudocode Pelanggaran Siswa

Pseudocode Pelanggaran Siswa	
1	Mulai
2	\$siswa = siswa(id, id)->first()
3	if(ajax)then
4	if(!empty(input(kategori)))then
5	\$data = siswa->poin(kategori, input(kategori))->get()
6	else
7	\$data = siswa->poin()->get()
8	endif
9	return datatables

4.2.4.6 Pseudocode Fungsional Tambah Poin Pelanggaran

Tabel 4.30 merupakan *pseudocode* dari fungsional tambah poin pelanggaran pada sistem yang dikembangkan.

Tabel 4.30 Pseudocode Tambah Poin Pelanggaran

Pseudocode Tambah Poin Pelanggaran	
1	Mulai
2	validate(input(pelanggaran, jumlah_poin, tanggal, tindak_lanjut))
3	if(input(kategori) = null)then
4	validate(input(kategori))
5	endif
6	poin()->create(siswa_id, kategori, jenis_pelanggaran,
7	penanganan, poin, tanggal)
8	redirect detail siswa

4.2.4.7 Pseudocode Fungsional Edit Poin Pelanggaran

Tabel 4.31 merupakan *pseudocode* dari fungsional edit poin pelanggaran pada sistem yang dikembangkan.

Tabel 4.31 Pseudocode Edit Poin Pelanggaran

Pseudocode Edit Poin Pelanggaran	
1	Mulai
2	validate(input(pelanggaran_edit, jumlah_poin_edit, tanggal_edit,
3	tindak_lanjut_edit))
4	if(input(kategori_edit) = null)then
5	validate(input(kategori_edit))
6	endif
7	\$poin = poin(id, id)->get();
8	poin()->update(pelanggaran_edit, jumlah_poin_edit,
9	tanggal_edit, tindak_lanjut_edit)
10	redirect detail siswa

4.2.4.8 Pseudocode Fungsional Hapus Poin Pelanggaran

Tabel 4.32 merupakan *pseudocode* dari fungsional hapus poin pelanggaran pada sistem yang dikembangkan.

Tabel 4.32 Pseudocode Hapus Poin Pelanggaran

Pseudocode Hapus Poin Pelanggaran	
1	Mulai
2	\$poin= poin(id, id)->get()
3	if(!empty(\$poin))then
4	\$poin->delete
5	else
6	create_session(gagal_hapus)
7	endif
8	redirect detail siswa

4.2.4.9 Pseudocode Fungsional Cetak Poin Pelanggaran

Tabel 4.33 merupakan *pseudocode* dari fungsional cetak poin pelanggaran pada sistem yang dikembangkan.

Tabel 4.33 Pseudocode Cetak Poin Pelanggaran

Pseudocode Cetak Poin Pelanggaran	
1	Mulai
2	\$siswa = siswa()->get()
3	\$poin = siswa->poin()->get()
4	\$pdf = loadview(siswa_pdf, siswa, poin)
5	return \$pdf->stream

4.2.4.10 Pseudocode Fungsional Tambah Motivasi

Tabel 4.34 merupakan *pseudocode* dari fungsional untuk menambahkan teks motivasi pada sistem yang dikembangkan.

Tabel 4.34 Pseudocode Tambah Motivasi

Pseudocode Tambah Motivasi	
1	Mulai
2	\$siswa = siswa(id, id)->first()
3	\$motivasi = siswa->motivasi()->first()
4	if(\$motivasi = null && input(motivasi) != null)then
5	\$siswa->motivasi()->create(siswa_id, motviiasi)
6	else if(input(motivasi) = "" && \$motivasi != null)
7	\$motivasi->delete
8	else if(\$motivasi != null)
9	\$motivasi->motivasi = input(motivasi)
10	endif
11	redirect detail siswa

4.2.4.11 Pseudocode Fungsional Tambah Akun Siswa

Pada proses menambah akun siswa, terdapat dua proses yang dijalankan oleh sistem. Proses yang pertama yaitu untuk melakukan pengecekan terhadap data siswa yang akan ditambahkan akunya, lalu proses yang kedua yaitu untuk melakukan konfirmasi proses penambahan akun siswa. Masing-masing dari proses tersebut dijelaskan pada *pseudocode* yang ditunjukkan pada Tabel 4.35 dan Tabel 4.36.

Tabel 4.35 Pseudocode Konfirmasi Data Siswa

Pseudocode Konfirmasi Data Siswa	
1	Mulai
2	validate(input(username, email, password_baru))
3	\$email = input(email)
4	\$pw = input(password_baru)
5	\$siswa = siswa(nis, username)->first()
6	\$username = pengguna(username, input(username))->first()
7	\$email = pengguna(email, \$email)->first()
8	if(\$siswa != null && \$username == null)then
9	if(\$email = null)
10	view(konfirmasi_akun, \$siswa, \$email, \$pw)
11	else
12	redirect tambah_akun_siswa
13	endif
14	else if(\$siswa = null)
15	redirect tambah_akun_siswa
16	else if(\$username != null)
17	redirect tambah_akun_siswa
18	endif

Tabel 4.36 Pseudocode Konfirmasi Tambah Akun Siswa

Pseudocode Konfirmasi Tambah Akun Siswa	
1	Mulai
2	\$pengguna = pengguna(username, input(nis))->first()
3	if(empty(\$pengguna))then
4	\$id = pengguna()->create(username, status_id, nama, password,
5	email)
6	siswa(nis, input(nis))->update(pengguna_id)
7	redirect kelola_siswa
8	else
9	redirect tambah_akun_siswa
10	endif

4.2.4.12 Pseudocode Fungsional Tambah Data Siswa

Tabel 4.37 merupakan *pseudocode* dari fungsional untuk menambahkan data siswa melalui *file* berformat CSV pada sistem yang dikembangkan.

Tabel 4.37 Pseudocode Tambah Data Siswa

Pseudocode Tambah Data Siswa	
1	Mulai
2	try
3	\$catchSiswa = ExcelToArray(file(data_siswa));
4	\$nis = []
5	\$collapse = \$catchSiswa
6	foreach(\$collapse)
7	\$nis[] = \$collapse[nis]
8	endforeach
9	\$siswa = siswa(nis, \$nis)->get()
10	if(count(\$siswa))then
11	view(import_data_siswa, \$siswa)
12	endif
13	try
14	ExcelImport(data_siswa)
15	redirect daftar_siswa
16	catch(\$e)
17	\$failures = \$e->failure
18	view(import_data_siswa, \$failure)
19	catch(\$e)
20	redirect import_siswa
21	endtrycatch

4.2.4.13 Pseudocode Fungsional Keluar

Tabel 4.38 merupakan *pseudocode* dari fungsional untuk keluar dari sistem yang dikembangkan.

Tabel 4.38 Pseudocode Keluar

Pseudocode Tambah Data Siswa	
1	Mulai
2	forget_session(session)
3	forget_session(nama)
4	redirect masuk

4.2.4.14 Pseudocode Fungsional Lihat Pelanggaran

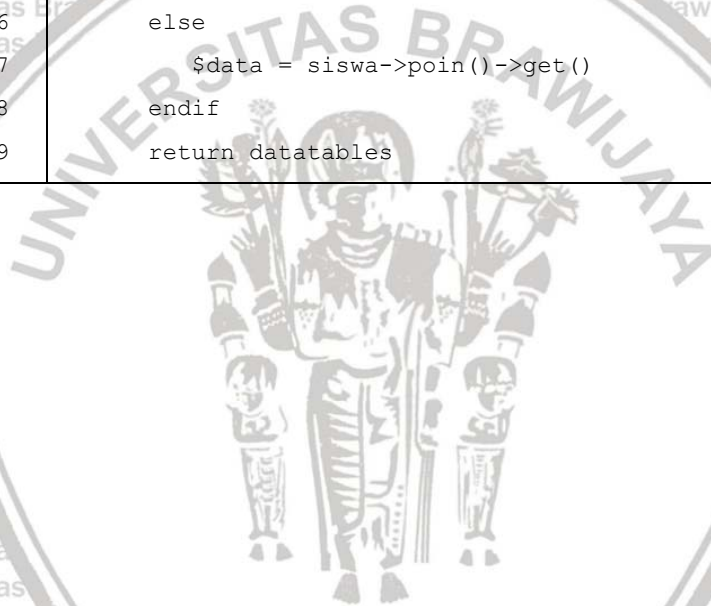
Pada proses melihat pelanggaran siswa, terdapat dua proses yang dijalankan oleh sistem. Proses yang pertama yaitu untuk menampilkan data profil dari siswa, lalu proses yang kedua yaitu untuk menampilkan data poin pelanggaran siswa yang dipilih. Masing-masing dari proses tersebut dijelaskan pada *pseudocode* yang ditunjukkan pada Tabel 4.39 dan Tabel 4.40.

Tabel 4.39 Pseudocode Lihat Profil Siswa

Pseudocode Lihat Profil Siswa	
1	Mulai
2	\$siswa = siswa(id, id)->first()
3	\$kategori = kategori()->get()
4	view(info_poin, \$siswa, \$kategori)

Tabel 4.40 Pseudocode Lihat Pelanggaran Siswa

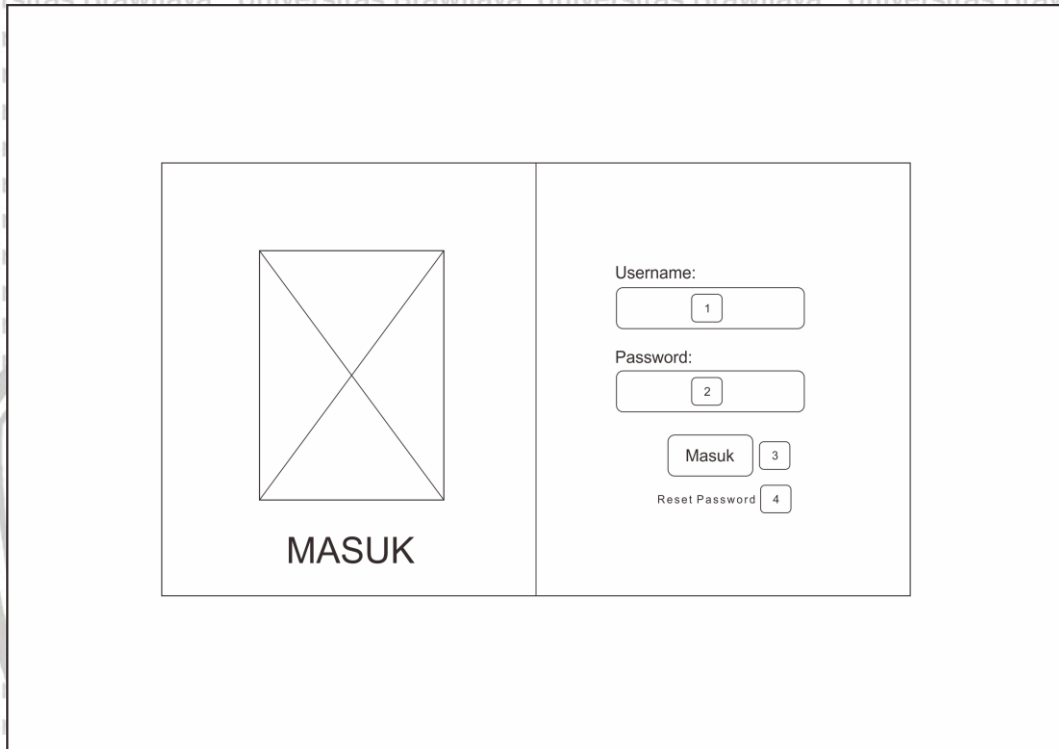
Pseudocode Pelanggaran Siswa	
1	Mulai
2	\$siswa = siswa(id, id)->first()
3	if (ajax) then
4	if (!empty(input(kategori))) then
5	\$data = siswa->poin(kategori, input(kategori))->get()
6	else
7	\$data = siswa->poin()->get()
8	endif
9	return datatables



4.2.5 Desain Wireframe

Desain *wireframe* merupakan suatu desain yang memiliki tujuan untuk memudahkan dalam menyusun tampilan pada saat proses implementasi dilakukan. Desain *wireframe* menggambarkan secara kasar bagaimana *user interface* dari sistem yang akan dibangun serta dapat memudahkan dalam memberikan informasi terkait tampilan dari sistem yang akan dibangun. Berikut merupakan *wireframe* dari sistem aplikasi SiTalang.

4.2.5.1 Desain Wireframe Masuk



Gambar 4.33 Wireframe Masuk

Gambar 4.33 merupakan tampilan dari desain *wireframe* halaman masuk. Terdapat gambar logo sekolah dan edit teks untuk memasukkan *username* dan *password* yang ditunjukkan pada nomor 1 dan 2. Ketika aktor sudah memasukkan *username* dan *password*, aktor dapat menekan tombol masuk pada nomor 3 agar proses masuk dapat segera dijalankan. Jika aktor ingin melakukan reset *password*, aktor dapat menekan *link* bertuliskan reset *password* yang ditandai dengan nomor 4.

4.2.5.2 Desain Wireframe Verifikasi Email

VERIFIKASI EMAIL

Masukkan E-mail:

1

Kirim Kode 2

<< Kembali ke Halaman Masuk 3

Gambar 4.34 Wireframe Verifikasi Email

Gambar 4.34 merupakan bentuk dari *wireframe* ketika aktor menekan tombol reset *password* pada halaman masuk. Aktor kemudian diarahkan untuk menuju ke halaman verifikasi *email* seperti pada Gambar 4.33. Aktor kemudian mengisikan *email* yang terdaftar ke edit teks yang ditunjukkan pada nomor 1. Setelah aktor memasukkan *email*, pengguna kemudian dapat menekan tombol kirim kode yang ditandai dengan nomor 2 sebagai bentuk konfirmasi untuk dapat segera melakukan proses verifikasi *email*. Jika aktor ingin kembali ke tampilan awal atau halaman masuk, actor dapat menekan tulisan kembali ke halaman masuk yang ditandai dengan nomor 3.

4.2.5.3 Desain Wireframe Konfirmasi Kode



Gambar 4.35 Wireframe Verifikasi Kode

Ketika pengguna sudah mendapatkan kode verifikasi melalui *email*, selanjutnya aktor melakukan validasi kode yang telah dikirim ke *email* aktor seperti yang ditunjukkan dengan Gambar 4.35. Setelah selesai memasukkan kode verifikasi pada edit teks yang ditandai dengan nomor 1, aktor dapat menekan tombol *submit* yang ditandai dengan nomor 2 agar proses validasi kode dapat segera dijalankan. Aktor juga dapat kembali ke halaman login dengan cara menekan link bertuliskan kembali ke halaman masuk yang ditandai dengan nomor 3.

4.2.5.4 Desain Wireframe Reset Password

RESET PASSWORD

Password Baru 1

Konfirmasi Password 2

Submit 3

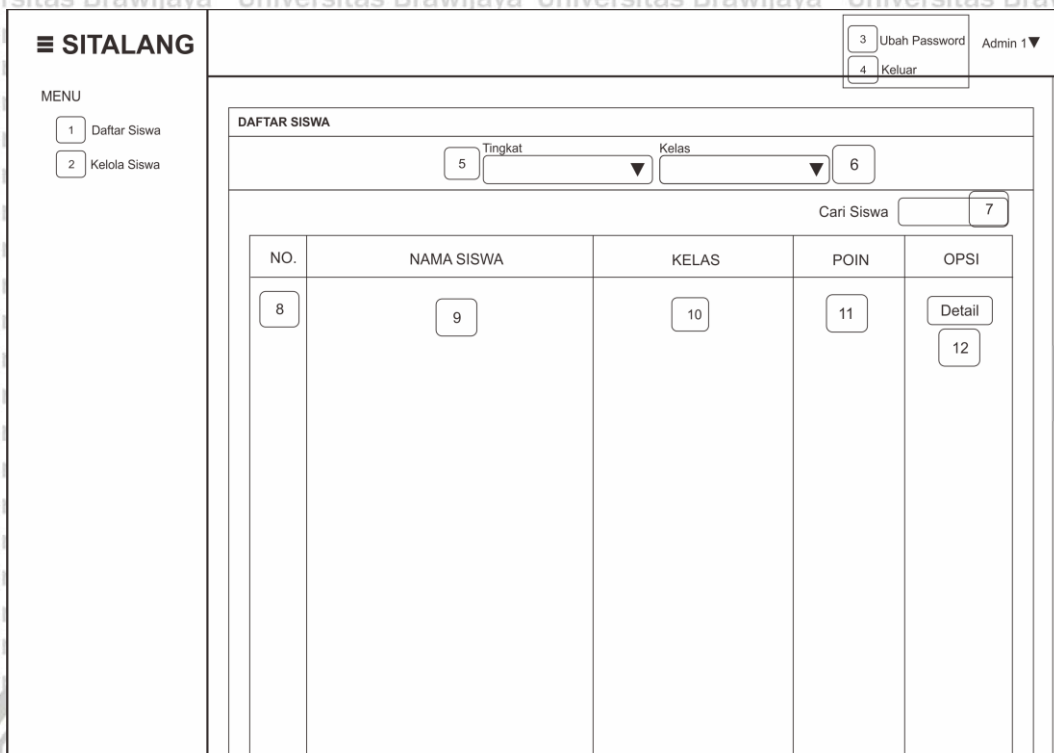
[<< Kembali ke Halaman Masuk](#) 4

Gambar 4.36 Wireframe Reset Password

Setelah melakukan proses validasi kode, aktor dapat mengatur ulang *password* pada halaman reset *password* seperti tampilan pada Gambar 4.36. Pengguna dapat mengisi *password* baru dan konfirmasi *password* pada edit teks yang ditandai dengan nomor 1 dan 2. Setelah mengisi *password* baru dan konfirmasi *password*, aktor dapat menekan tombol bertuliskan *submit* yang ditandai dengan nomor 3 agar proses reset *password* dapat segera dijalankan. Pengguna juga dapat kembali ke halaman masuk dengan menekan tulisan *link* yang ditandai dengan nomor 4.



4.2.5.5 Desain Wireframe Daftar Siswa



Gambar 4.37 Wireframe Daftar Siswa

Gambar 4.37 memberikan gambaran terkait bagaimana tampilan dari halaman daftar siswa atau halaman utama aplikasi aktor guru. Terdapat *sidebar* dengan menu yang bertuliskan daftar siswa dan kelola siswa yang ditandai dengan nomor 1 dan 2. Menu daftar siswa merupakan menu yang bertujuan untuk menampilkan halaman daftar siswa, sedangkan menu kelola siswa merupakan menu yang bertujuan untuk menampilkan halaman kelola siswa agar bisa mengakses fungsional tambah akun siswa dan tambah data siswa. Dengan adanya menu *dropdown* tersebut diharapkan nantinya pada saat aplikasi dikembangkan lebih lanjut, fitur tambahan dapat ditambahkan pada *dropdown* menu tersebut. Di bagian kanan *navigation bar* terdapat sebuah *dropdown* untuk menampilkan menu ubah *password* dan keluar yang masing-masing ditandai dengan nomor 3 dan 4.

Pada bagian konten halaman daftar siswa terdapat *dropdown* yang bertujuan untuk menyeleksi siswa sesuai dengan tingkat dan kelas yang ditandai dengan nomor 5 dan 6 serta terdapat edit teks pencarian siswa yang ditandai dengan nomor 7 agar dapat mempercepat proses pencarian siswa. Terdapat juga beberapa informasi dari masing-masing kolom tabel yang menampilkan data ringkas siswa yang ditandai dengan nomor yang masing-masing mempunyai keterangan berurutan sebagai berikut, nomor 8 hingga 11 menampilkan keterangan nomor, nama siswa, kelas, serta total poin yang telah didapatkan oleh siswa. Pada nomor 12 terdapat sebuah tombol untuk menampilkan halaman detail siswa sesuai dengan baris masing-masing data siswa.

4.2.5.6 Desain Wireframe Lihat Pelanggaran Siswa

DATA SISWA

PROFIL SISWA

NIS (Nomor Induk Siswa) : 87346212

Nama : Muhammad Sodik

Kelas : 10 Multimedia 3

Predikat Tingkat Laku : A - Amat Baik

MOTIVASI DAN PENGUATAN

Masukkan teks...

Edit Simpan

Catatan: Motivasi dan penguatan harap diisi sebelum dicetak

Tambah Poin Pelanggaran

Cetak Poin Pelanggaran

DATA POIN

Kategori

No.	Tanggal	Pelanggaran	Kategori	Poin	OPSI
8	9	10	11	12	13
Total Poin				14	

Pertama Terakhir

15

Gambar 4.38 Wireframe Detail Poin Pelanggaran Siswa

Wireframe pada Gambar 4.38 menjelaskan terkait detail dari siswa dan detail poin yang telah didapatkan oleh siswa. Di antara informasi tersebut terdapat informasi data siswa seperti NIS (Nomor Induk Siswa), nama, kelas, dan predikat tingkah laku yang ditandai dengan kotak nomor 1. Pada bagian data siswa juga terdapat kolom *textarea* untuk mengisi motivasi yang dapat diisikan oleh aktor guru yang ditandai pada nomor 2. Isi dari motivasi dan penguatan tersebut dapat diedit dan disimpan dengan menekan tombol yang ditandai dengan nomor 3 dan 4. Selanjutnya terdapat tombol untuk menambah poin pelanggaran dan mencetak poin pelanggaran yang ditandai dengan nomor 5 dan 6.

Aktor dapat melihat informasi poin-poin yang pernah didapatkan oleh siswa. Hal tersebut dapat dilihat pada tanda nomor 8, 9, 10, 11 dan 12 yang masing-masing adalah data dari nomor, tanggal, jenis pelanggaran, kategori, dan poin diikuti dengan total poin yang ditandai dengan nomor 14. Pada data nomor 13 terdapat kolom opsi yang nantinya akan terdapat opsi untuk dapat mengedit maupun menghapus poin pelanggaran yang telah ditambahkan. Terdapat juga *dropdown* yang ditandai dengan nomor 7 yang memiliki fungsi agar aktor dapat melakukan filter terhadap pelanggaran siswa berdasarkan kategori untuk lebih memudahkan dalam melakukan pengelompokan informasi kategori pelanggaran yang didapatkan oleh siswa. Nomor 15 merupakan sebuah *pagination* untuk dapat menuju ke halaman berikutnya ketika poin yang didapatkan siswa cukup banyak sehingga membutuhkan lebih dari satu halaman data poin pelanggaran siswa.

4.2.5.7 Desain Wireframe Tambah Poin Pelanggaran Siswa

Gambar 4.39 Wireframe Tambah Poin

Pada *wireframe* tambah poin yang ditunjukkan pada Gambar 4.39, aktor dapat memasukkan poin pelanggaran yang akan diberikan kepada siswa. Pada edit teks nomor 1, aktor dapat memasukkan pelanggaran yang dilakukan oleh siswa. Pada edit teks nomor 2, aktor dapat memilih kategori pelanggaran yang dilakukan oleh siswa apakah ringan, sedang, maupun berat. Pada edit teks nomor 3, aktor dapat memasukkan jumlah poin yang akan diberikan kepada siswa. Pada edit teks nomor 4, aktor dapat menambahkan tanggal ditamahnya poin pelanggaran siswa. Yang terakhir yaitu pada nomor 5, aktor dapat memasukkan penanganan atau tindak lanjut yang akan diberikan kepada siswa sesuai dengan pelanggaran yang telah dilakukan siswa.

Setelah mengisikan jumlah poin pada masing-masing edit teks tersebut, aktor dapat menekan tombol *submit* yang ditandai dengan nomor 7 agar proses penambahan poin pelanggaran siswa dapat segera diproses. Aktor juga dapat menekan tombol *kembali* yang ditandai dengan nomor 6 untuk kembali ke halaman detail poin pelanggaran siswa.

4.2.5.8 Desain Wireframe Edit Poin Pelanggaran

The wireframe shows a web interface for editing student violations. It features a top navigation bar with a hamburger menu icon and the text 'SITALANG', and a user profile 'Admin 1'. A left sidebar contains a 'MENU' section with 'Daftar Siswa' and 'Kelola Siswa'. The main content area is titled 'EDIT POIN' and contains a form titled 'EDIT POIN PELANGGARAN'. The form includes the following elements:

- 1**: Text input field for 'Pelanggaran yang Dilakukan'.
- 2**: Dropdown menu for 'Kategori'.
- 3**: Text input field for 'Jumlah Poin'.
- 4**: Text input field for 'Tanggal'.
- 5**: Text area for 'Tindak Lanjut/Penyelesaian'.
- 6**: 'Kembali' button.
- 7**: 'Submit' button.

Gambar 4.40 Wireframe Edit Poin Pelanggaran

Gambar 4.40 merupakan gambar dari *wireframe* menu edit poin pelanggaran siswa. Beberapa susunan komponen dari tampilan *wireframe* edit poin pelanggaran tersebut sebenarnya tidak jauh berbeda dengan tampilan *wireframe* menu tambah poin pelanggaran siswa. Yang membedakan dengan halaman tambah poin pelanggaran yaitu bahwa pada halaman edit poin pelanggaran sudah terdapat nilai pada masing-masing edit teks yang diambil dari *database* sehingga pengguna dapat mengubah nilai-nilai tersebut kemudian disimpan pada *database* untuk menyimpan perubahan yang dilakukan oleh aktor.

Terdapat edit teks yang ditandai dengan nomor 1 yang berfungsi agar pengguna dapat mengubah pelanggaran yang dilakukan oleh siswa. Pada nomor 2 terdapat edit teks untuk mengubah kategori pelanggaran yang telah dilakukan oleh siswa. Pada nomor 3 aktor dapat mengubah jumlah poin yang telah ditambahkan kepada siswa yang melakukan pelanggaran. Pada nomor 4 aktor dapat mengubah data dari tanggal ditambahkan pelanggaran siswa. Pada nomor 5 aktor dapat mengubah informasi dari penanganan yang diberikan kepada siswa.

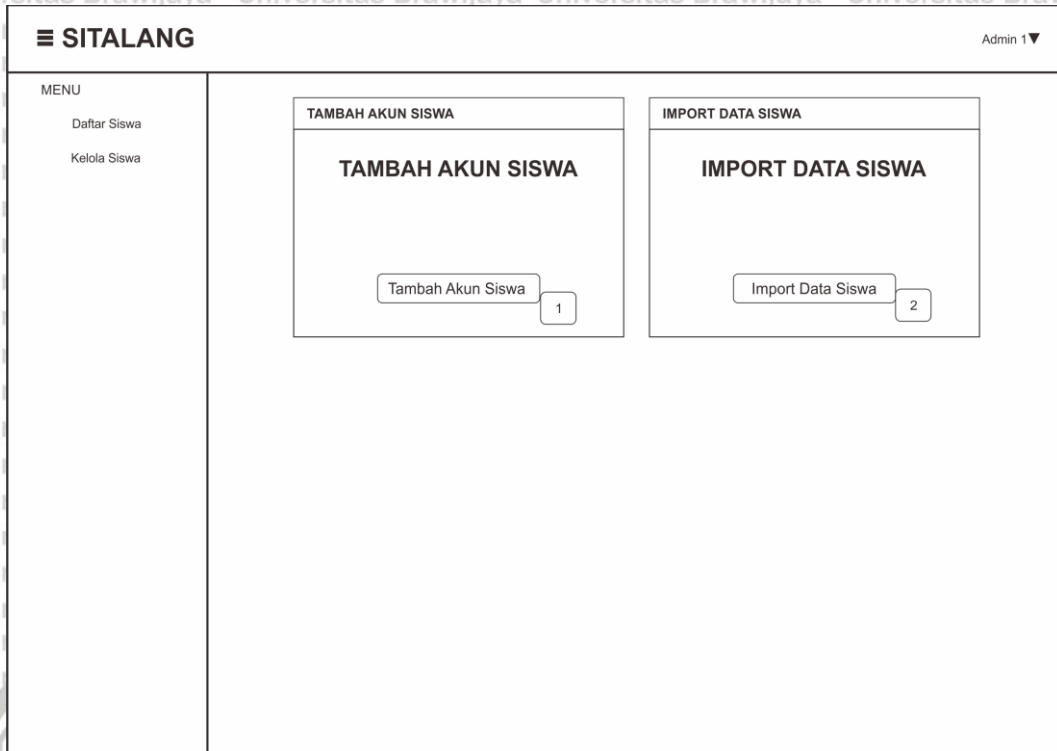
Setelah mengubah nilai pada masing-masing edit teks tersebut, aktor dapat menekan tombol *submit* yang ditandai dengan nomor 7 agar proses edit poin pelanggaran siswa dapat segera diproses. Aktor juga dapat menekan tombol *kembali* yang ditandai dengan nomor 6 untuk kembali ke halaman detail poin pelanggaran siswa.

4.2.5.9 Desain Wireframe Ubah Password

Gambar 4.41 Wireframe Ubah Password

Gambar 4.41 merupakan gambaran *wireframe* dari halaman ubah *password*. Aktor selanjutnya hanya perlu mengisikan pada edit teks *password baru* dan konfirmasi *password* yang ditandai dengan nomor 1 dan 2. Setelah mengisikan edit teks *password* baru dan konfirmasi *password* tersebut, aktor dapat menekan tombol *submit* yang ditandai dengan nomor 3 agar sistem dapat memproses perubahan *password*. Terdapat tombol kembali ditandai pada nomor 4 yang dapat mengembalikan tampilan sistem ke halaman daftar siswa.

4.2.5.10 Desain Wireframe Halaman Kelola Siswa



Gambar 4.42 Wireframe Halaman Kelola Siswa

Gambar 4.42 merupakan gambaran *wireframe* dari halaman kelola siswa. Pada halaman kelola siswa tersebut terdapat dua menu yaitu tambah akun siswa dan *import* data siswa. Jika aktor menekan tombol bertuliskan tambah akun siswa yang ditandai dengan nomor 1, aktor akan diarahkan oleh sistem untuk menuju ke halaman tambah akun siswa. Apabila aktor menekan tombol bertuliskan *import* data siswa yang ditandai dengan nomor 2, aktor akan diarahkan pada halaman tambah data siswa atau *import* data siswa.

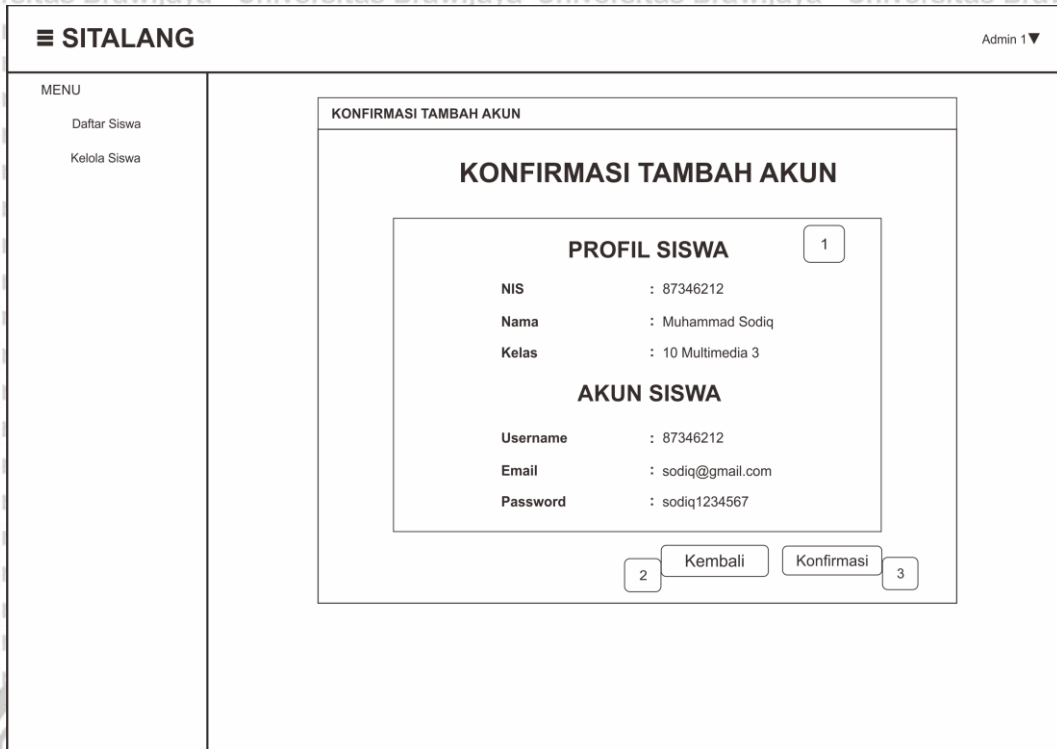
4.2.5.11 Desain Wireframe Tambah Akun Siswa

Gambar 4.43 Wireframe Tambah Akun Siswa

Pada *wireframe* tambah akun siswa yang ditunjukkan pada Gambar 4.43, aktor dapat menambahkan akun siswa agar nantinya siswa dapat melakukan masuk untuk melihat poin pelanggaran yang telah didapatkan oleh siswa tersebut. Pada edit teks nomor 1, aktor dapat memasukkan *username* atau NIS milik siswa yang akan ditambahkan akunnya. Pada edit teks nomor 2, aktor mengisi *email* dari siswa yang akan ditambahkan akunnya. Aktor kemudian mengatur *password* akun siswa yang ditandai pada edit teks nomor 3 dan 4.

Setelah mengatur *password*, aktor dapat menekan tombol bertuliskan cek data yang ditandai dengan nomor 6 agar sistem dapat melakukan verifikasi dan validasi terhadap data yang telah dimasukan tersebut dan menampilkan halaman konfirmasi tambah akun siswa. Aktor juga dapat kembali ke halaman kelola siswa dengan menekan tombol kembali yang ditandai dengan nomor 5.

4.2.5.12 Desain Wireframe Konfirmasi Tambah Akun Siswa

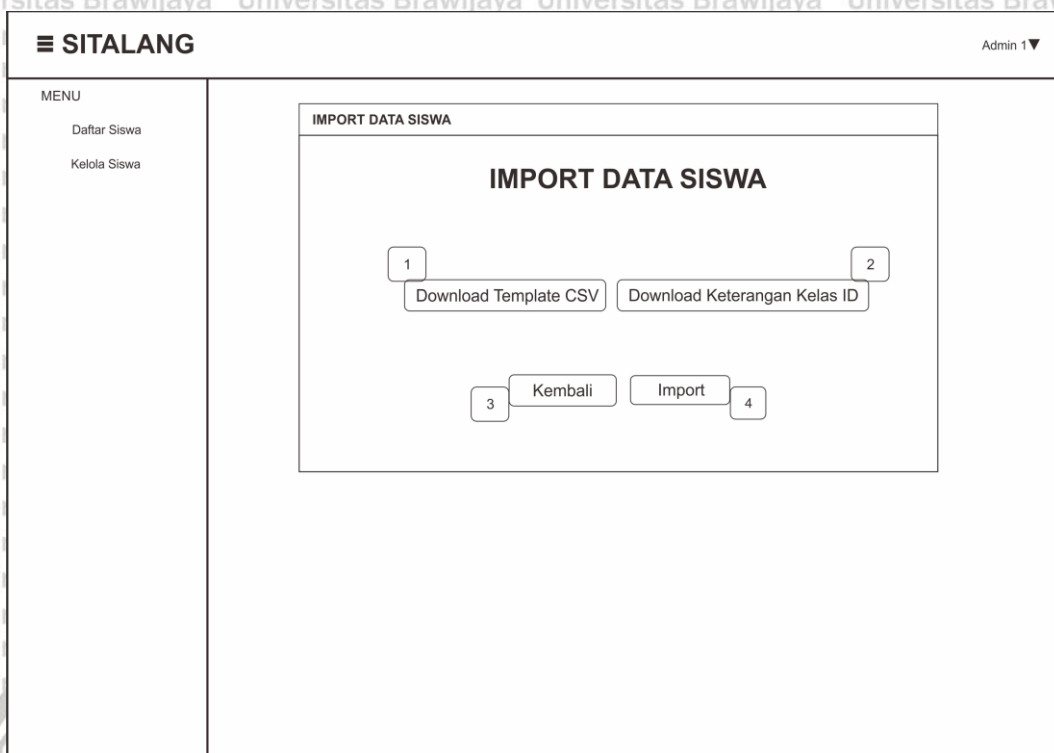


Gambar 4.44 Wireframe Konfirmasi Tambah Akun Siswa

Wireframe pada Gambar 4.44 merupakan wireframe dari halaman konfirmasi tambah akun siswa. Pada halaman konfirmasi tambah akun siswa tersebut, sistem akan menampilkan data siswa sesuai dengan masukan NIS yang telah dimasukkan oleh aktor pada halaman tambah akun siswa. sistem juga akan menampilkan keterangan akun siswa yang akan ditambahkan sesuai dengan data masukan aktor pada halaman tambah akun siswa. Seluruh data siswa dan data akun siswa tersebut ditunjukkan pada bagian nomor 1 di wireframe Gambar 4.44.

Apabila aktor ingin menambahkan akun siswa, aktor perlu melakukan konfirmasi dengan cara menekan tombol konfirmasi yang ditandai dengan nomor 3 dan sistem akan memproses tambah akun siswa serta menyimpan keterangan data akun siswa pada database. Jika aktor ingin membatalkan tambah akun siswa, aktor dapat menekan tombol kembali yang ditandai dengan nomor 2 untuk kembali ke halaman tambah akun siswa.

4.2.5.13 Desain Wireframe Tambah Data Siswa



Gambar 4.45 Wireframe Tambah Data Siswa

Gambar 4.45 merupakan gambaran *wireframe* dari halaman tambah data siswa. Pada halaman tambah data siswa tersebut terdapat beberapa tombol seperti yang ditandai pada nomor 1 hingga 4. Pada tombol *download template CSV* yang ditandai dengan nomor 1, memiliki fungsi untuk mengunduh *file* CSV yang telah disediakan oleh sistem. Pada *file* tersebut sudah terdapat keterangan kolom sehingga aktor hanya perlu mengisikan data siswa yang akan dimasukkan saja. Pada tombol *download keterangan kelas id* yang ditandai dengan nomor 2, memiliki fungsi agar aktor dapat mengunduh keterangan kelas id sebagai pedoman ketika akan menambahkan kelas id pada *file* CSV yang akan diunggah sehingga aktor dapat memasukkan siswa sesuai dengan kelasnya. Tombol *import* yang ditandai pada nomor 4 memiliki fungsi untuk menampilkan modal *import data siswa* agar aktor dapat mengunggah *file* data siswa yang akan dimasukkan ke sistem atau *database* sekaligus memproses *file* yang telah diunggah tersebut. Aktor juga dapat kembali ke halaman kelola siswa dengan cara menekan tombol kembali yang ditandai pada nomor 3.

4.2.5.14 Desain Wireframe Lihat Pelanggaran

SITALANG

2 Ubah Password Hallo, Muhammad Sodik ▼
3 Keluar

MENU
1 Pelanggaran

DATA POIN, MUHAMMAD SODIQ

PROFIL SISWA

NIS (Nomor Induk Siswa) : 87346212 4
Nama : Muhammad Sodik
Kelas : 10 Multimedia 3
Predikat Tingkah Laku : A - Amat Baik
Motivasi dari Guru : Sudah baik, pertahankan!

POIN PELANGGARAN SISWA

Kategori 5 ▼

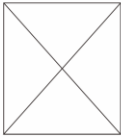

No.	Tanggal	Pelanggaran	Kategori	Poin
6	7	8	9	10
Total Poin				11

Pertama Terakhir 12

Gambar 4.46 Wireframe Lihat Pelanggaran

Gambar 4.46 merupakan desain *wireframe* dari fungsional lihat pelanggaran di mana siswa menjadi aktor dari sistem. Tampilan dari desain tersebut tidak jauh berbeda dari tampilan fungsional lihat pelanggaran siswa yang aktornya adalah guru. Yang menjadi perbedaan yaitu pada tampilan lihat pelanggaran Gambar 4.46, siswa hanya dapat melihat informasi dari profil siswa itu sendiri dengan poin pelanggaran siswanya dan tidak dapat untuk menambah, menghapus, mengedit, maupun melakukan aksi yang lain seperti yang dilakukan oleh aktor guru. Penjelasan singkat yaitu terdapat *sidebar* dengan menu yang bertuliskan pelanggaran yang ditandai dengan nomor 1. Terdapat menu pada *navigation bar* dengan menu yaitu ubah *password* dan keluar yang ditandai dengan nomor 2 dan 3. Terdapat profil siswa yang ditandai dengan nomor 4 diikuti dengan *dropdown* untuk melakukan *sorting* sesuai kategori pelanggaran yang ditandai dengan nomor 5. Setelah itu terdapat keterangan poin pelanggaran yang ditandai dengan nomor 6 hingga 11 serta terdapat tombol *pagination* jika poin pelanggaran cukup banyak yang ditandai dengan nomor 12.

4.2.5.15 Desain Wireframe Hasil Cetak Poin Pelanggaran

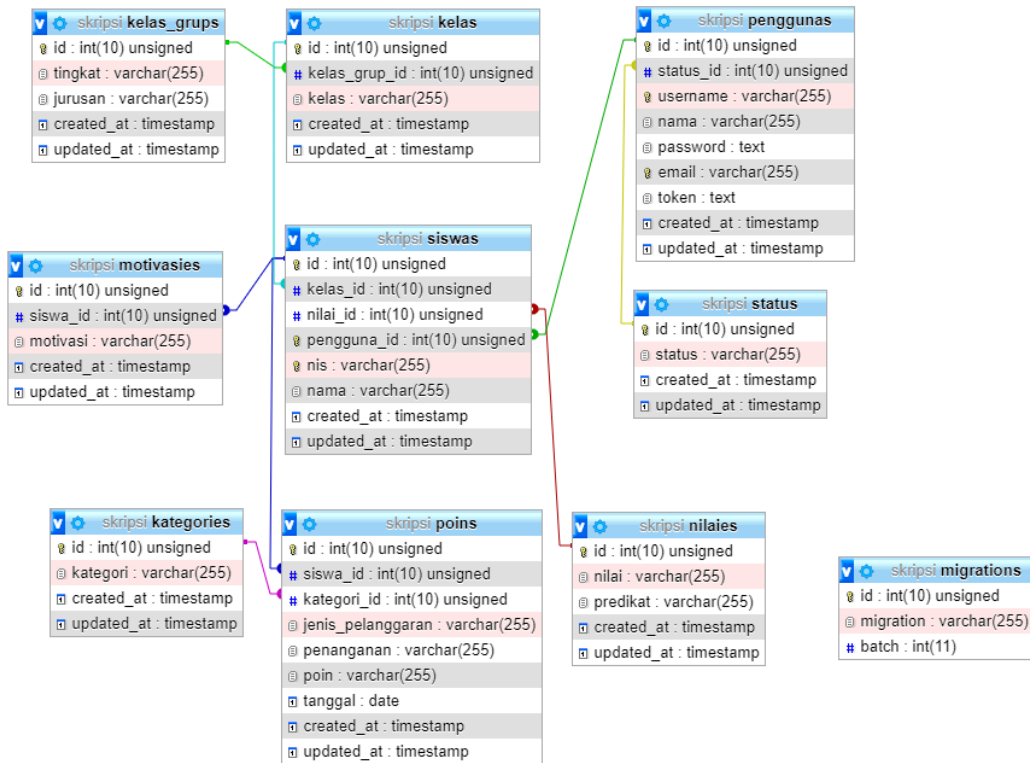
		PEMERINTAH KOTA PASURUAN DINAS PENDIDIKAN SMK NEGERI 1 PASURUAN Jalan Veteran, Kota Malang Telp. (0341) 421380 Website: http://www.smkn1-pasuruan.sch.id				
POIN PELANGGARAN SISWA						
NIS: 236427682						
Nama: Muhammad Sodik						
Kelas: 10 Multimedia 3						
Predikat Tingkah Laku: A - Amat Baik						
No.	Tanggal	Pelanggaran	Kategori	Poin		
Total Poin						
Catatan:						
Pasuruan, 26 Januari 2021 Kepala Kedisiplinan SMKN 1 Pasuruan,						
Muhammad Sodik, S.Pd. NIP/NIK: 746536453273272375						

Gambar 4.47 Wireframe Hasil Cetak Poin Pelanggaran

Pada Gambar 4.47 merupakan tampilan dari hasil fungsional cetak poin pelanggaran siswa. Gambar 4.47 memiliki beberapa komponen seperti profil siswa, pelanggaran-pelanggaran yang sudah dilakukan oleh siswa, tanggal, kategori, dan poin yang telah didapatkan oleh siswa. Pada bagian akhir dokumen terdapat catatan dari guru serta tanda tangan pihak sekolah sebagai bentuk validitas dokumen sebelum diberikan kepada orang tua siswa nantinya.

4.3 Hasil Implementasi Sistem

4.3.1 Hasil Implementasi *Physical Data Model (PDM)*



Gambar 4.48 *Physical Data Model (PDM)*

Gambar 4.48 merupakan gambar dari *Physical Data Model (PDM)* dari aplikasi SiTalang. PDM dapat disebut juga sebagai hasil implementasi dari ERD (*Entity Relationship Diagram*) yang sudah dibuat pada tahap perancangan sistem menjadi bentuk tabel yang terdapat pada *database* serta dapat diakses oleh sistem beserta relasi yang telah didefinisikan. Terdapat perbedaan dari penamaan tabel pada *database* dengan nama pendefinisian ERD dikarenakan pada implementasi PDM menggunakan fitur migrasi Laravel yang penamaan tabel secara otomatis diubah menjadi bentuk jamak.

4.3.2 Batasan Implementasi Sistem

Batasan implementasi sistem pada proses pengembangan aplikasi sistem tata tertib pelanggaran tersebut yaitu:

1. Data Siswa

Data siswa yang terdapat pada aplikasi merupakan data buatan pengembang dan bukan data asli dari pihak sekolah.

2. Akun Pengguna

Data akun pengguna yang terdapat pada aplikasi merupakan data yang ditambahkan oleh pengembang.



4.3.3 Spesifikasi Implementasi Sistem

Pada proses pengembangan aplikasi Sistem Tata Tertib Pelanggaran (SiTalang), terdapat dua jenis perangkat yang digunakan yaitu perangkat lunak (*software*) dan perangkat keras (*hardware*). Spesifikasi dari perangkat keras yang digunakan dalam proses pengembangan aplikasi Sistem Tata Tertib Pelanggaran (SiTalang) yang ditunjukkan pada Tabel 4.41.

Tabel 4.41 Spesifikasi Perangkat Keras

Komponen	Spesifikasi
Processor	AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx (8 CPUs), ~2.1GHz
Memory RAM (Random Access Memory)	8192MB RAM
Chip Type	GeForce GTX 1050

Selain terdapat spesifikasi perangkat keras, juga terdapat spesifikasi perangkat lunak yang ditunjukkan pada Tabel 4.42.

Tabel 4.42 Spesifikasi Perangkat Lunak

Perangkat Lunak	Deskripsi
Visual Studio Code	Sebagai <i>code editor</i> bahasa pemrograman dalam mengembangkan aplikasi.
XAMPP	Sebagai <i>server</i> sementara yang digunakan pada saat proses pengembangan aplikasi.
Google Chrome	Sebagai <i>web browser</i> yang digunakan untuk menampilkan hasil dari setiap proses pengembangan aplikasi.
PhpMyAdmin	Bagian dari XAMPP yang merupakan perangkat lunak serta berfungsi untuk melakukan pengolahan <i>database</i> pada saat proses pengembangan aplikasi.

4.3.4 Hasil Implementasi Model

Penggunaan dari *framework* Laravel sangat memudahkan pengembang dalam mengembangkan sebuah aplikasi atau *website*. Kerangka dalam penyusunan kode program *framework* Laravel yaitu dengan menggunakan MVC (*Model View Controller*) sehingga penulisan kode program dapat menjadi lebih terstruktur.

Penggunaan *model* dalam Laravel ini disebut dengan *Eloquent* Laravel sehingga pengembang hanya perlu memanggil nama *file* dari *model* yang telah dibuat ketika ingin mengakses data pada *database*. Hal tersebut dapat mempersingkat *query* atau kode program sehingga menjadikan pemanggilan data dari *database* menjadi lebih efektif. Berikut merupakan penulisan kode program *model* dalam pengembangan aplikasi SiTalang.

4.3.4.1 Model Pengguna

Tabel 4.43 Source Code Model Pengguna.php

Implementasi Model Pengguna	
1	class Pengguna extends Model{
2	protected \$table = 'penggunas';
3	protected \$fillable = ['username', 'status_id', 'nama',
4	'password', 'email', 'token'];
5	public function status(){}
6	public function siswa(){}
7	}

Tabel 4.43 merupakan kode program pada *file model* Pengguna. Pada *file model* tersebut terdapat *class* bernama Pengguna yang di *extend* dari *library Model* Laravel yang ditunjukkan pada baris nomor 1. Baris nomor 2 memberitahukan kepada sistem bahwa *file model* tersebut berhubungan dengan tabel bernama *penggunas* yang terdapat pada *database*. Baris nomor 3 hingga 4 memiliki fungsi agar kolom pada tabel pengguna seperti *username*, *status_id*, *nama*, *password*, *email*, dan *token* dapat diisi atau menjalankan perintah *create* atau *insert*. Baris nomor 5 dan 6 berfungsi untuk memberikan perintah relasi.

4.3.4.2 Model Siswa

Tabel 4.44 Source Code Model Siswa

Implementasi Model Siswa	
1	class Siswa extends Model
2	{
3	protected \$table = 'siswas';
4	protected \$fillable = ['nilai_id', 'pengguna_id', 'kelas_id',
5	'nis', 'nama'];
6	public function poin(){}
7	public function nilai(){}
8	public function motivasi(){}
9	public function kelas(){}
10	public function pengguna(){}
11	}

Tabel 4.44 merupakan kode program pada *file model* Siswa. Pada *file model* tersebut terdapat *class* bernama Siswa yang di *extend* dari *library Model* Laravel yang ditunjukkan pada baris nomor 1. Baris nomor 3 memiliki fungsi untuk menghubungkan *file model* dengan tabel bernama siswas yang terdapat pada *database*. Baris nomor 4 hingga 5 memiliki fungsi agar kolom pada tabel siswa yang bernama nilai_id, pengguna_id, kelas_id, nis, dan nama dapat diisi atau menjalankan perintah *create* atau *insert*. Baris nomor 6 hingga 10 memiliki fungsi untuk menjalankan fungsi relasi dengan tabel poin, nilai, motivasi, kelas, dan pengguna yang terdapat pada *database*.

4.3.4.3 Model Poin

Tabel 4.45 Source Code Model Poin

Implementasi Model Poin	
1	<code>class Poin extends Model</code>
2	<code>{</code>
3	<code>protected \$table = "poin";</code>
4	<code>protected \$fillable = ['siswa_id', 'kategori_id',</code>
5	<code>'jenis_pelanggaran', 'poin', 'tanggal'];</code>
6	<code>public function siswa() {}</code>
7	<code>public function kategori() {}</code>
	<code>}</code>

Tabel 4.45 merupakan kode program pada *file model* Poin. Pada *file model* tersebut terdapat *class* bernama Poin yang di *extend* dari *library Model* Laravel yang ditunjukkan pada baris nomor 1. Baris nomor 3 memiliki fungsi untuk menghubungkan *file model* dengan tabel bernama poin yang terdapat pada *database*. Baris nomor 4 hingga 5 memiliki fungsi agar kolom pada tabel poin yang bernama siswa_id, kategori_id, jenis_pelanggaran, poin, dan tanggal dapat diisi atau menjalankan perintah *create* atau *insert*. Baris nomor 6 dan 7 memiliki fungsi untuk menjalankan relasi dengan tabel siswas dan kategories yang terdapat pada *database*.

4.3.4.4 Model Status

Tabel 4.46 Source Code Model Status

Implementasi Model Status	
1	<code>class Status extends Model</code>
2	<code>{</code>
3	<code>protected \$table = 'status';</code>
4	<code>public function pengguna() {}</code>
5	<code>}</code>

Tabel 4.46 merupakan kode program pada *file model* Status. Pada *file model* tersebut terdapat *class* bernama Status yang di *extend* dari *library Model* Laravel

yang ditunjukkan pada baris nomor 1. Baris nomor 3 memiliki fungsi untuk menghubungkan *file model* dengan tabel bernama status yang terdapat pada *database*. Baris nomor 4 memiliki fungsi untuk menjalankan relasi dengan tabel pengguna yang terdapat pada *database*.

4.3.4.5 Model Nilai

Tabel 4.47 Source Code Model Nilai

Implementasi Model Nilai	
1	class Nilai extends Model
2	{
3	protected \$table = 'nilaies';
4	public function siswa(){}
5	}

Tabel 4.47 merupakan kode program pada *file model* Nilai. Pada *file model* tersebut terdapat *class* bernama Nilai yang di *extend* dari *library Model* Laravel yang ditunjukkan pada baris nomor 1. Baris nomor 3 memiliki fungsi untuk menghubungkan *file model* dengan tabel bernama nilaies yang terdapat pada *database*. Baris nomor 4 memiliki fungsi untuk menjalankan relasi dengan tabel siswas yang terdapat pada *database*.

4.3.4.6 Model Motivasi

Tabel 4.48 Source Code Model Motivasi

Implementasi Model Motivasi	
1	class Motivasi extends Model
2	{
3	protected \$table = "motivasi";
4	protected \$fillable = ['siswa_id', 'motivasi'];
5	public function siswa(){}
6	}

Tabel 4.48 merupakan kode program pada *file model* Motivasi. Pada *file model* tersebut terdapat *class* bernama Motivasi yang di *extend* dari *library Model* Laravel yang ditunjukkan pada baris nomor 1. Baris nomor 3 memiliki fungsi untuk menghubungkan *file model* dengan tabel bernama motivasi yang terdapat pada *database*. Baris nomor 4 memiliki fungsi agar kolom pada tabel motivasi yang bernama siswa_id dan motivasi dapat diisi atau menjalankan perintah *create* atau *insert*. Baris nomor 5 memiliki fungsi untuk menjalankan relasi dengan tabel siswas yang terdapat pada *database*.

4.3.4.7 Model Kelas

Tabel 4.49 Source Code Model Kelas

Implementasi Model Kelas	
1	class Kelas extends Model
2	{
3	protected \$table = "kelas";
4	public function siswa(){}
5	public function kelasGrup(){}
6	}

Tabel 4.49 merupakan kode program pada *file model* Kelas. Pada *file model* tersebut terdapat *class* bernama Kelas yang di *extend* dari *library Model* Laravel yang ditunjukkan pada baris nomor 1. Baris nomor 3 memiliki fungsi untuk menghubungkan *file model* dengan tabel bernama kelas yang terdapat pada *database*. Baris nomor 4 dan 5 memiliki fungsi untuk menjalankan relasi dengan tabel siswas dan kelas_grups yang terdapat pada *database*.

4.3.4.8 Model KelasGrup

Tabel 4.50 Source Code Model KelasGrup

Implementasi Model KelasGrup	
1	class KelasGrup extends Model
2	{
3	protected \$table = 'kelas grups';
4	public function kelas(){}
5	}

Tabel 4.50 merupakan kode program pada *file model* KelasGrup. Pada *file model* tersebut terdapat *class* bernama KelasGrup yang di *extend* dari *library Model* Laravel yang ditunjukkan pada baris nomor 1. Baris nomor 3 memiliki fungsi untuk menghubungkan *file model* dengan tabel bernama kelas_grups yang terdapat pada *database*. Baris nomor 4 memiliki fungsi untuk menjalankan relasi dengan tabel kelas yang terdapat pada *database*.

4.3.4.9 Model Kategori

Tabel 4.51 Source Code Model Kategori

Implementasi Model Kategori	
1	class Kategori extends Model
2	{
3	protected \$table = "kategories";
4	public function poin(){}
5	}

Tabel 4.51 merupakan kode program pada *file model* Kategori. Pada *file model* tersebut terdapat *class* bernama Kategori yang di *extend* dari *library Model* Laravel yang ditunjukkan pada baris nomor 1. Baris nomor 3 memiliki fungsi untuk menghubungkan *file model* dengan tabel bernama *categories* yang terdapat pada *database*. Baris nomor 4 memiliki fungsi untuk menjalankan relasi dengan tabel *poins* yang terdapat pada *database*.

4.3.5 Hasil Implementasi Controller

Setelah pada sub bab 4.3.3 menjelaskan tentang implementasi *Model*, pada bab ini akan dijelaskan bagaimana proses implementasi dari aplikasi SiTalang pada bagian *controller*. *Controller* merupakan jembatan penghubung antara data yang terdapat pada *database* dengan *view* atau *interface* aplikasi sehingga data dapat ditampilkan kepada aktor dan aplikasi dapat menjadi *website* yang dinamis. Berikut merupakan penulisan dari *controller* dalam proses implementasi pengembangan aplikasi SiTalang.

4.3.5.1 Controller AuthController

Tabel 4.52 merupakan kode program dari *controller* *AuthController*. *Controller* *AuthController* merupakan *controller* yang bertugas untuk mengurus jalannya program pada bagian autentikasi sistem. Pada *controller* *AuthController* ini menjelaskan beberapa fungsional dari masuk, reset *password*, ubah *password*, tambah akun siswa, dan keluar.

Tabel 4.52 Source Code Controller AuthController

Implementasi Controller AuthController	
1	<code>use Illuminate\Http\Request;</code>
2	<code>use App\Pengguna;</code>
3	<code>use App\Siswa;</code>
4	<code>use Illuminate\Support\Facades\Notification;</code>
5	<code>use App\Notifications\NewMessage;</code>
6	<code>use Faker\Factory as Faker;</code>
7	<code>use Illuminate\Support\Facades\Crypt;</code>
8	<code>use Illuminate\Support\Facades\Hash;</code>
9	
10	<code>class AuthController extends Controller</code>
11	<code>{</code>
12	<code>public function index(Request \$request){}</code>
13	<code>public function masuk(Request \$request){}</code>
14	<code>public function prosesMasuk(Request \$request){}</code>
15	<code>public function verifikasi_email(Request \$request){}</code>
16	<code>public function prosesVerifikasiEmail(Request</code>
17	<code>\$request){}</code>
18	<code>public function verifikasi_kode(Request \$request){}</code>

Tabel 4.52 Source Code Controller AuthController (lanjutan)

```

19 public function prosesVerifikasiKode(Request
20 $request){}
21 public function reset_password(Request $request){}
22 public function prosesResetPassword(Request $request){}
23 public function ubah_password(Request $request){}
24 public function prosesUbahPassword(Request $request,
25 $id){}
26 public function tambah_akun(Request $request){}
27 public function prosesTambahAkun(Request $request){}
28 public function prosesKonfirmasiAkun(Request $request){}
29 public function prosesKeluar(Request $request){}
30 }
    
```

Pada Tabel 4.52 baris ke 1 hingga 8 merupakan bagian yang berfungsi untuk mengakses *library* yang akan digunakan pada *controller* AuthController seperti *Request* untuk menerima masukan pengguna, *Notifications* untuk mengirimkan *email*, *Faker* untuk membuat data secara acak, *Crypt* untuk melakukan enkripsi dan dekripsi, dan *Hash* untuk menampilkan data dalam bentuk kalimat acak serta memanggil *file* model bernama Pengguna dan Siswa yang berfungsi untuk pengolahan data. Baris nomor 10 terdapat pendeklarasian *class controller* bernama AuthController yang di *extend* dari *library Controller* Laravel. Di dalam *class controller* AuthController tersebut terdapat fungsi-fungsi dari masing-masing fungsional autentikasi aplikasi yang dibangun seperti masuk, ubah *password*, reset *password*, tambah akun siswa, dan keluar.

Baris nomor 12 merupakan fungsi bernama *index* yang bertujuan untuk menyeleksi *session* pada saat aplikasi pertama kali diakses. Jika telah terdapat *session* pada aplikasi, maka sistem akan menampilkan halaman utama aplikasi. Jika pada aplikasi masih belum terdapat *session*, maka sistem akan menampilkan halaman masuk.

Baris nomor 13 merupakan fungsi bernama *masuk* yang berfungsi untuk menampilkan halaman masuk. Baris nomor 14 berfungsi untuk memproses masukan aktor yang telah dilakukan pada halaman masuk. Fungsi *prosesMasuk* akan melakukan verifikasi terhadap masukan pengguna dan menyeleksi status pengguna apakah guru atau siswa serta membuat *session*. Jika status pengguna adalah guru maka sistem akan menampilkan halaman utama guru. Jika status pengguna adalah siswa maka sistem akan menampilkan halaman utama siswa.

Baris nomor 15 merupakan fungsi bernama *verifikasi_email* yang berfungsi untuk menampilkan halaman verifikasi *email*. Baris nomor 16 hingga 17 kemudian memproses masukan dari pengguna pada halaman verifikasi *email* tersebut. Fungsi *prosesVerifikasiEmail* akan melakukan verifikasi terhadap masukan *email* pengguna dan akan mengirimkan kode pada *email* yang telah dimasukkan oleh pengguna tersebut. Jika terverifikasi, sistem akan menampilkan halaman verifikasi

kode yang ditunjukkan pada fungsi baris nomor 18. Jika tidak terverifikasi, sistem akan tetap menampilkan halaman verifikasi *email* dan menampilkan pesan bahwa *email* salah.

Baris nomor 18 merupakan fungsi bernama verifikasi_kode yang berfungsi untuk menampilkan halaman verifikasi kode. Baris nomor 19 hingga 20 kemudian memproses masukan dari pengguna pada halaman verifikasi kode tersebut. Fungsi prosesVerifikasiKode akan melakukan verifikasi terhadap masukan kode yang telah diterima pengguna melalui *email*. Jika terverifikasi, sistem akan menampilkan halaman reset *password* yang ditunjukkan pada fungsi baris nomor 21. Jika tidak terverifikasi, sistem akan tetap menampilkan halaman verifikasi kode dan menampilkan pesan bahwa kode yang dimasukkan tidak sesuai.

Baris nomor 21 merupakan fungsi yang berfungsi untuk menampilkan halaman reset *password*. Masukan dari pengguna pada halaman reset *password* tersebut kemudian diproses pada fungsi yang ditunjukkan pada nomor 22 yang bernama prosesResetPassword. Pada fungsi prosesResetPassword tersebut sistem akan melakukan validasi terhadap masukan pengguna apakah valid atau tidak valid. Jika valid, sistem akan menyimpan masukan dari pengguna tersebut ke *database* serta menampilkan halaman masuk. Jika masukan tidak valid, sistem akan tetap menampilkan halaman reset *password* serta akan menampilkan pesan bahwa masukan *password* baru tidak valid.

Baris nomor 23 merupakan fungsi yang berfungsi untuk menampilkan halaman ubah *password*. Pada fungsi yang bernama ubah_password tersebut, sistem akan melakukan seleksi apakah status dari aktor adalah guru atau siswa. Masukan aktor pada halaman ubah *password* kemudian diproses melalui fungsi bernama prosesUbahPassword yang ditunjukkan pada baris nomor 24 hingga 25. Pada fungsi prosesUbahPassword tersebut sistem akan melakukan validasi terhadap masukan pengguna apakah valid atau tidak valid. Jika valid, sistem akan menyimpan masukan dari pengguna tersebut ke *database* serta menampilkan halaman masuk. Jika masukan tidak valid, sistem akan tetap menampilkan halaman ubah *password* serta akan menampilkan pesan bahwa masukan *password* baru tidak valid.

Baris nomor 26 merupakan fungsi yang berfungsi untuk menampilkan halaman tambah akun siswa. Baris nomor 27 kemudian memproses masukan dari pengguna pada halaman tambah akun siswa tersebut. Fungsi prosesTambahAkun akan melakukan verifikasi dan validasi terhadap masukan aktor yang telah dilakukan pada halaman tambah akun siswa. Jika informasi masukan terverifikasi dan valid, sistem akan menampilkan halaman konfirmasi akun dengan menampilkan data siswa dan data akun siswa yang akan ditambahkan. Jika informasi data tambah akun tidak terverifikasi dan tidak valid, sistem akan tetap menampilkan halaman tambah akun siswa dan menampilkan pesan bahwa masukan aktor tidak valid.

Baris nomor 28 merupakan fungsi yang menjalankan proses konfirmasi tambah akun siswa yang telah dilakukan. Jika aktor melakukan konfirmasi tambah

akun siswa, sistem akan memproses melalui fungsi prosesKonfirmasiAkun tersebut dan akan menyimpan informasi akun siswa yang baru ditambahkan ke *database*.

Baris nomor 29 merupakan fungsi yang berfungsi untuk menjalankan proses keluar dari sistem agar aktor kembali menjadi pengguna yang belum diketahui identitasnya oleh sistem. Pada fungsi tersebut, sistem akan menghapus *session* yang telah dibuat pada saat proses masuk diikuti dengan menampilkan halaman masuk.

4.3.5.2 Controller SiswaController

Tabel 4.53 merupakan baris kode yang menjelaskan *controller* SiswaController yang di *extend* dari *library Controller* Laravel. *Controller* ini memiliki fungsi-fungsi yang berhubungan dengan siswa seperti menampilkan daftar siswa, menampilkan detail informasi, tambah data siswa, dan detail poin pelanggaran siswa serta memproses pemberian motivasi oleh aktor guru.

Tabel 4.53 Source Code Controller SiswaController

Implementasi Controller SiswaController	
1	<code>use Maatwebsite\Excel\Facades\Excel;</code>
2	<code>use App\Imports\SiswaImport;</code>
3	<code>use Illuminate\Http\Request;</code>
4	<code>use App\Siswa;</code>
5	<code>use Carbon\Carbon;</code>
6	<code>use Yajra\DataTables\DataTables;</code>
7	
8	<code>class SiswaController extends Controller{</code>
9	<code>public function getKelasByTingkat(\$tingkat){}</code>
10	<code>public function prosesLihatDaftarSiswa(Request \$request){}</code>
11	<code>public function prosesDetailSiswa(Request \$request, \$id){}</code>
12	<code>public function prosesDetailPoin(Request \$request){}</code>
13	<code>public function prosesInfoPoin(Request \$request, \$id){}</code>
14	<code>public function prosesTambahMotivasi(Request \$request, \$id){}</code>
15	<code>public function kelola_siswa(Request \$request){}</code>
16	<code>public function import_siswa(Request \$request){}</code>
17	<code>public function prosesImportSiswa(Request \$request){}</code>
18	<code>public function put(){}</code>
19	<code>}</code>

Pada Tabel 4.53 baris nomor 1 hingga 6 merupakan bagian yang berfungsi untuk mengakses *library* yang akan digunakan pada *controller* SiswaController seperti Maatwebsite Excel untuk *import* data melalui *file* CSV, SiswaImport untuk melakukan konfigurasi *import* CSV yang dilakukan, *Request* untuk menerima masukan pengguna, *Carbon* untuk mengkonversi format tanggal, dan DataTables

untuk menampilkan data dalam bentuk tabel serta memanggil *file model* bernama Siswa untuk pengelolaan data pada tabel Siswa. Baris nomor 8 terdapat pendeklarasian *class controller* bernama SiswaController yang di *extend* dari *library Controller* Laravel. Di dalam *class controller* SiswaController tersebut terdapat fungsi-fungsi yang menjalankan fungsional aplikasi yang dibangun.

Baris nomor 9 merupakan fungsi untuk menampilkan kelas sesuai tingkat yang telah dipilih pada halaman daftar siswa. Pada baris nomor 10, memiliki fungsi untuk menampilkan seluruh data siswa yang terdapat pada *database* untuk ditampilkan pada halaman daftar siswa. Baris nomor 11 merupakan fungsi untuk menampilkan informasi siswa pada halaman detail siswa atau lihat pelanggaran siswa diikuti oleh fungsi pada baris nomor 12 untuk menampilkan informasi poin pelanggaran yang akan ditampilkan pada halaman detail siswa atau lihat pelanggaran siswa. Proses penyeleksian data siswa dan poin pelanggaran siswa yaitu berdasarkan dengan id siswa yang terdapat pada *database*.

Baris nomor 13 merupakan fungsi untuk menjalankan fungsional lihat pelanggaran yang diakses oleh aktor siswa. Pada fungsi prosesInfoPoin tersebut sistem akan menampilkan data informasi siswa maupun data poin pelanggaran siswa sesuai dengan siswa yang melakukan masuk dan diseleksi melalui fungsi bernama prosesDetailPoin yang ditunjukkan pada baris nomor 12. Proses penyeleksian data siswa yaitu berdasarkan dengan id siswa yang terdapat pada *database*. Baris nomor 14 merupakan fungsi yang berfungsi agar aktor guru dapat menambahkan pesan motivasi kepada siswa.

Baris nomor 15 merupakan fungsi untuk melihat halaman kelola siswa. Pada halaman kelola siswa tersebut terdapat dua menu yaitu tambah data siswa dan tambah akun siswa. Fungsi yang menjelaskan terkait proses tambah akun siswa dijelaskan pada sub bab 4.3.4.1 atau pada penjelasan *controller* AuthController. Pada *controller* SiswaController ini hanya menjelaskan terkait proses dari tambah data siswa yang dijelaskan pada baris nomor 16 dan 17.

Baris nomor 16 pada *controller* SiswaController tersebut merupakan fungsi untuk menampilkan halaman tambah data siswa. Pada halaman tersebut aktor akan mengunggah *file* dengan format CSV yang kemudian akan diproses pada fungsi bernama prosesImportSiswa yang ditunjukkan pada baris nomor 17. Pada fungsi prosesImportSiswa tersebut sistem akan melakukan validasi data atau *file* yang telah diunggah oleh aktor. Jika *file* atau data yang diunggah oleh aktor valid, maka data di dalam *file* yang diunggah akan disimpan pada *database*. Akan tetapi jika *file* atau data yang diunggah oleh aktor tidak valid, maka sistem akan menampilkan informasi bahwa data atau *file* yang diunggah tidak valid dan sistem tetap menampilkan halaman tambah data siswa.

Baris nomor 18 merupakan fungsi untuk memproses *file* keterangan kelas id yang bisa di *download* oleh aktor pada halaman tambah data siswa. *File* keterangan kelas id tersebut bertujuan untuk memberikan informasi kepada aktor ketika akan melakukan *import* data siswa menggunakan *file* CSV dikarenakan pada

format *file* CSV terdapat kolom kelas id yang harus diisi sesuai dengan kelas id yang terdapat pada *file* keterangan kelas id.

4.3.5.3 Controller PoinController

Tabel 4.54 merupakan kode program dari *controller* bernama PoinController yang di *extend* dari *library Controller* Laravel. *Controller* PoinController memiliki fungsi-fungsi yang berhubungan dengan poin pelanggaran siswa. Beberapa fungsi tersebut yaitu menambah poin pelanggaran, mengedit poin pelanggaran, dan menghapus poin pelanggaran yang dipilih.

Tabel 4.54 Source Code Controller PoinController

Implementasi Controller PoinController	
1	<code>use Illuminate\Http\Request;</code>
2	<code>use App\Poin;</code>
3	<code>use Barryvdh\DomPDF\Facade as PDF;</code>
4	<code>use Carbon\Carbon;</code>
5	
6	<code>class PoinController extends Controller</code>
7	<code>{</code>
8	<code>public function tambah_poin(Request \$request, \$id){}</code>
9	<code>public function prosesTambahPoin(Request \$request, \$id){}</code>
10	<code>public function edit_poin(Request \$request, \$id){}</code>
11	<code>public function prosesEditPoin(Request \$request, \$id){}</code>
12	<code>public function prosesHapusPoin(Request \$request, \$id){}</code>
13	<code>public function prosesCetakPoin(\$id)</code>
14	<code>}</code>

Pada Tabel 4.54 baris nomor 1 hingga 4 merupakan bagian yang berfungsi untuk mengakses *library* yang akan digunakan pada *controller* PoinController seperti *Request* untuk menerima masukan pengguna, *DomPDF* untuk melakukan fungsi dokumen PDF, dan *Carbon* untuk mengkonversi format tanggal serta memanggil *file model* bernama Poin. Baris nomor 6 terdapat pendeklarasian *class controller* bernama PoinController yang di *extend* dari *library Controller* Laravel. Di dalam *class controller* PoinController tersebut terdapat fungsi-fungsi yang menjalankan fungsional aplikasi yang dibangun.

Baris nomor 8 merupakan fungsi yang berfungsi untuk menampilkan halaman tambah poin pelanggaran. Hasil dari masukan informasi poin pelanggaran yang akan ditambahkan kemudian diproses melalui fungsi bernama *prosesTambahPoin* yang ditunjukkan pada baris nomor 9. Pada fungsi *prosesTambahPoin* tersebut sistem akan melakukan validasi terhadap masukan aktor apakah valid atau tidak valid. Jika valid maka sistem akan menyimpan informasi poin pelanggaran ke *database* diikuti dengan menampilkan halaman lihat pelanggaran siswa. Jika tidak valid, sistem akan menampilkan pesan bahwa masukan informasi poin

pelanggaran tidak valid dan tetap menampilkan halaman tambah poin pelanggaran.

Baris nomor 10 merupakan fungsi yang berfungsi untuk menampilkan halaman edit poin pelanggaran. Hasil dari perubahan informasi poin pelanggaran tersebut kemudian akan diproses melalui fungsi bernama prosesEditPoin yang ditunjukkan pada baris nomor 11. Pada fungsi prosesEditPoin sistem akan melakukan validasi terhadap masukan aktor apakah valid atau tidak valid. Jika valid maka sistem akan menyimpan hasil perubahan informasi poin pelanggaran ke *database* diikuti dengan menampilkan halaman lihat pelanggaran siswa. Jika tidak valid, sistem akan menampilkan pesan bahwa perubahan informasi poin pelanggaran tidak valid dan tetap menampilkan halaman edit poin pelanggaran.

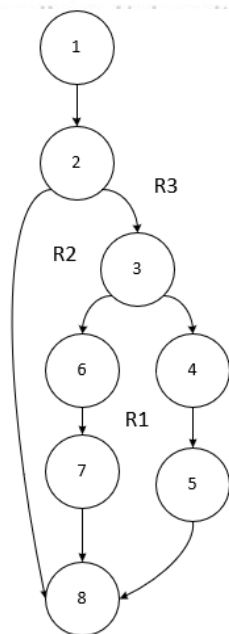
Baris nomor 12 merupakan fungsi yang menjalankan fungsional hapus poin pelanggaran. Pada fungsi prosesHapusPoin tersebut sistem akan menyeleksi poin pelanggaran yang akan dihapus berdasarkan id kemudian menghapusnya dari *database*. Setelah proses hapus berhasil, sistem menampilkan pesan bahwa proses hapus poin pelanggaran telah berhasil. Baris nomor 13 merupakan fungsi yang menjalankan fungsional cetak poin pelanggaran dalam format PDF.

4.3.6 Unit Testing

Tahap *unit testing* dilakukan dengan menggunakan *basis path testing* dengan metode *flow graph* dan menentukan *cyclomatic complexity* untuk menentukan kasus uji pada fungsional sistem. Langkah-langkah pengujian *unit testing* telah dijelaskan pada Bab 2 sub bab 2.6.3.2 terkait *unit testing*. Pengujian unit pada penelitian ini hanya dilakukan pada fungsi-fungsi utama yang terdapat pada sistem. Beberapa fungsi utama dari sistem diantaranya yaitu tambah poin pelanggaran, edit poin pelanggaran, dan hapus poin pelanggaran. Berikut merupakan hasil dari pengujian unit yang dilakukan pada sistem yang dikembangkan.

4.3.6.1 Unit Testing Fungsional Tambah Poin Pelanggaran

Gambar 4.49 merupakan gambar dari *flow graph* fungsi tambah poin pelanggaran. Susunan *flow graph* tambah poin pelanggaran didapatkan dari *pseudocode* yang telah dijelaskan pada Tabel 4.30 yang menjelaskan *pseudocode* tambah poin pelanggaran.



Gambar 4.49 Flow graph Fungsi Tambah Poin Pelanggaran

Setelah diperoleh *flow graph* dari fungsi tambah poin pelanggaran seperti yang ditunjukkan pada Gambar 4.49, langkah selanjutnya yaitu menentukan *cyclomatic complexity* dan menentukan jalur-jalur dasar sesuai dengan *cyclomatic complexity* yang telah diperoleh tersebut. Cara menentukan *cyclomatic complexity* yaitu dengan proses perhitungan yang telah dijelaskan pada Persamaan 2.1 hingga Persamaan 2.3 pada Bab 2 sub bab 2.6.3.2.

Sesuai dengan Gambar 4.49, dapat diketahui jumlah *region* pada *flow graph* fungsi tambah poin pelanggaran yaitu sebanyak 3 yang ditunjukkan dengan tanda R1, R2, dan R3. Jumlah dari predikat *node* yaitu sebanyak 2 yang ditunjukkan pada sampul 2 dan 3. Jumlah dari sampul atau *node* yaitu sebanyak 8 dan jumlah dari sisi atau *edge* yaitu sebanyak 9. Dari beberapa komponen tersebut dapat ditentukan *cyclomatic complexity* $V(G)$ dari fungsi tambah poin pelanggaran yaitu sebagai berikut.

$$V(G) = \text{jumlah area } (R) = 3 \tag{4.1}$$

$$V(G) = E - N + 2 = 9 - 8 + 2 = 3 \tag{4.2}$$

$$V(G) = P + 1 = 2 + 1 = 3 \tag{4.3}$$

Berdasarkan *cyclomatic complexity* yang telah didapat, jalur dasar yang dapat didefinisikan adalah sebagai berikut.

1. 1-2-3-4-5-8
2. 1-2-3-6-7-8
3. 1-2-8

Dari jalur dasar yang telah didapatkan, langkah selanjutnya yaitu mendefinisikan kasus uji sesuai dengan jalur-jalur dasar yang telah didapatkan

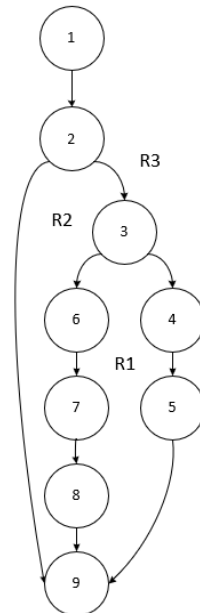
tersebut. Tabel 4.55 merupakan kasus uji yang didapatkan sesuai dengan masing-masing jalur dasar yang telah didapatkan.

Tabel 4.55 Kasus Uji Fungsi Tambah Poin Pelanggaran

No.	Kode Fungsional	Kasus Uji	Hasil yang Diharapkan
1.	SITALANG-F-O6	Menambahkan poin pelanggaran tanpa mengisi kategori poin pelanggaran.	Sistem menampilkan pesan <i>error</i> bahwa kategori harus diisi.
2.	SITALANG-F-O6	Menambahkan poin pelanggaran dengan data masukan yang valid	Sistem menyimpan poin pelanggaran yang telah ditambahkan pada <i>database</i> .
3.	SITALANG-F-O6	Menambahkan poin pelanggaran dengan data yang tidak valid.	Sistem menampilkan pesan <i>error</i> bahwa data poin pelanggaran harus diisi dengan lengkap dan benar.

4.3.6.2 Unit Testing Fungsional Edit Poin Pelanggaran

Gambar 4.50 merupakan gambar dari *flow graph* fungsi edit poin pelanggaran. Susunan *flow graph* edit poin pelanggaran didapatkan dari *pseudocode* yang telah dijelaskan pada Tabel 4.31 yang menjelaskan *pseudocode* edit poin pelanggaran.



Gambar 4.50 Flow graph Fungsi Edit Poin Pelanggaran

Setelah diperoleh *flow graph* dari fungsi edit poin pelanggaran seperti yang ditunjukkan pada Gambar 4.50, langkah selanjutnya yaitu menentukan *cyclomatic*

complexity dan menentukan jalur-jalur dasar sesuai dengan *cyclomatic complexity* yang telah diperoleh tersebut. Cara menentukan *cyclomatic complexity* yaitu dengan proses perhitungan yang telah dijelaskan pada Persamaan 2.1 hingga Persamaan 2.3 pada Bab 2 sub bab 2.6.3.2.

Sesuai dengan Gambar 4.50, dapat diketahui jumlah *region* pada *flow graph* fungsi edit poin pelanggaran yaitu sebanyak 3 yang ditunjukkan dengan tanda R1, R2, dan R3. Jumlah dari predikat *node* yaitu sebanyak 2 yang ditunjukkan pada sampel 2 dan 3. Jumlah dari sampel atau *node* yaitu sebanyak 9 dan jumlah dari sisi atau *edge* yaitu sebanyak 10. Dari beberapa komponen tersebut dapat ditentukan *cyclomatic complexity* $V(G)$ dari fungsi tambah poin pelanggaran yaitu sebagai berikut.

$$V(G) = \text{jumlah area } (R) = 3 \tag{4.4}$$

$$V(G) = E - N + 2 = 10 - 9 + 2 = 3 \tag{4.5}$$

$$V(G) = P + 1 = 2 + 1 = 3 \tag{4.6}$$

Berdasarkan *cyclomatic complexity* yang telah didapat, jalur dasar yang dapat didefinisikan adalah sebagai berikut.

1. 1-2-3-4-5-9
2. 1-2-3-6-7-8-9
3. 1-2-9

Dari jalur dasar yang telah didapatkan, langkah selanjutnya yaitu mendefinisikan kasus uji sesuai dengan jalur-jalur dasar yang telah didapatkan tersebut. Tabel 4.56 merupakan kasus uji yang didapatkan sesuai dengan masing-masing jalur dasar yang telah didapatkan.

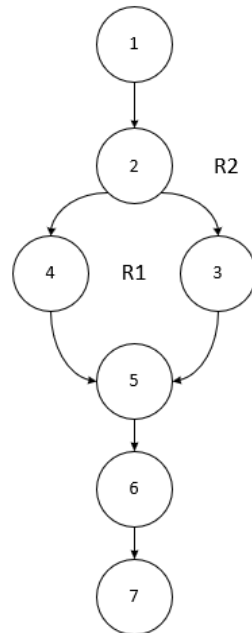
Tabel 4.56 Kasus Uji Fungsi Edit Poin Pelanggaran

No.	Kode Fungsional	Kasus Uji	Hasil yang Diharapkan
1.	SITALANG-F-07	Mengubah informasi poin pelanggaran tanpa mengisi kategori poin pelanggaran.	Sistem menampilkan pesan <i>error</i> bahwa kategori harus diisi.
2.	SITALANG-F-07	Mengubah informasi poin pelanggaran dengan data masukan yang valid	Sistem menyimpan poin pelanggaran yang telah diperbarui pada <i>database</i> .
3.	SITALANG-F-07	Mengubah informasi poin pelanggaran dengan data yang tidak valid.	Sistem menampilkan pesan <i>error</i> bahwa data poin pelanggaran harus diisi dengan benar.



4.3.6.3 Unit Testing Fungsional Hapus Poin Pelanggaran

Gambar 4.51 merupakan gambar dari *flow graph* fungsi hapus poin pelanggaran. Susunan *flow graph* hapus poin pelanggaran didapatkan dari *pseudocode* yang telah dijelaskan pada Tabel 4.32 yang menjelaskan *pseudocode* hapus poin pelanggaran.



Gambar 4.51 Flow graph Fungsi Hapus Poin Pelanggaran

Setelah diperoleh *flow graph* dari fungsi tambah poin pelanggaran seperti yang ditunjukkan pada Gambar 4.51, langkah selanjutnya yaitu menentukan *cyclomatic complexity* dan menentukan jalur-jalur dasar sesuai dengan *cyclomatic complexity* yang telah diperoleh tersebut. Cara menentukan *cyclomatic complexity* yaitu dengan proses perhitungan yang telah dijelaskan pada Persamaan 2.1 hingga Persamaan 2.3 pada Bab 2 sub bab 2.6.3.2.

Sesuai dengan Gambar 4.51, dapat diketahui jumlah *region* pada *flow graph* fungsi hapus poin pelanggaran yaitu sebanyak 2 yang ditunjukkan dengan tanda R1 dan R2. Jumlah dari predikat *node* yaitu sebanyak 1 yang ditunjukkan pada sampel 2. Jumlah dari sampel atau *node* yaitu sebanyak 7 dan jumlah dari sisi atau *edge* yaitu sebanyak 7. Dari beberapa komponen tersebut dapat ditentukan *cyclomatic complexity* $V(G)$ dari fungsi tambah poin pelanggaran yaitu sebagai berikut.

$$V(G) = \text{jumlah area } (R) = 2 \tag{4.1}$$

$$V(G) = E - N + 2 = 7 - 7 + 2 = 2 \tag{4.2}$$

$$V(G) = P + 1 = 1 + 1 = 2 \tag{4.3}$$

Berdasarkan *cyclomatic complexity* yang telah didapat, jalur dasar yang dapat didefinisikan adalah sebagai berikut.

1. 1-2-3-5-6-7

2. 1-2-4-5-6-7

Dari jalur dasar yang telah didapatkan, selanjutnya yaitu dengan mendefinisikan kasus uji sesuai dengan jalur-jalur dasar yang telah didapatkan tersebut. Tabel 4.57 merupakan kasus uji yang didapatkan sesuai dengan masing-masing jalur dasar yang telah didapatkan.

Tabel 4.57 Kasus Uji Fungsi Tambah Poin Pelanggaran

No.	Kode Fungsional	Kasus Uji	Hasil yang Diharapkan
1.	SITALANG-F-O8	Menghapus poin yang terdapat pada daftar pelanggaran siswa.	Sistem menghapus poin pelanggaran yang dipilih.
2.	SITALANG-F-O8	Menghapus poin pelanggaran yang tidak terdapat pada <i>database</i> , akan tetapi masih ada pada daftar pelanggaran siswa.	Sistem menampilkan pesan <i>error</i> bahwa kemungkinan poin pelanggaran sudah terhapus.

4.3.6.4 Analisis Hasil Pengujian Unit

Berdasarkan hasil pengujian unit yang telah dilakukan, peneliti melakukan pengujian unit terhadap 3 fungsi utama dari sistem yang dikembangkan. Tiga fungsi tersebut diantaranya yaitu tambah poin pelanggaran, edit poin pelanggaran, dan hapus poin pelanggaran. Hasil dari pengujian unit yang telah dilakukan didapatkan *flow graph* dari masing-masing fungsi serta didapatkan juga *cyclomatic complexity* dan jalur dasar dari masing-masing fungsi yang diuji sehingga dapat menentukan kasus uji dari masing-masing fungsi tersebut.

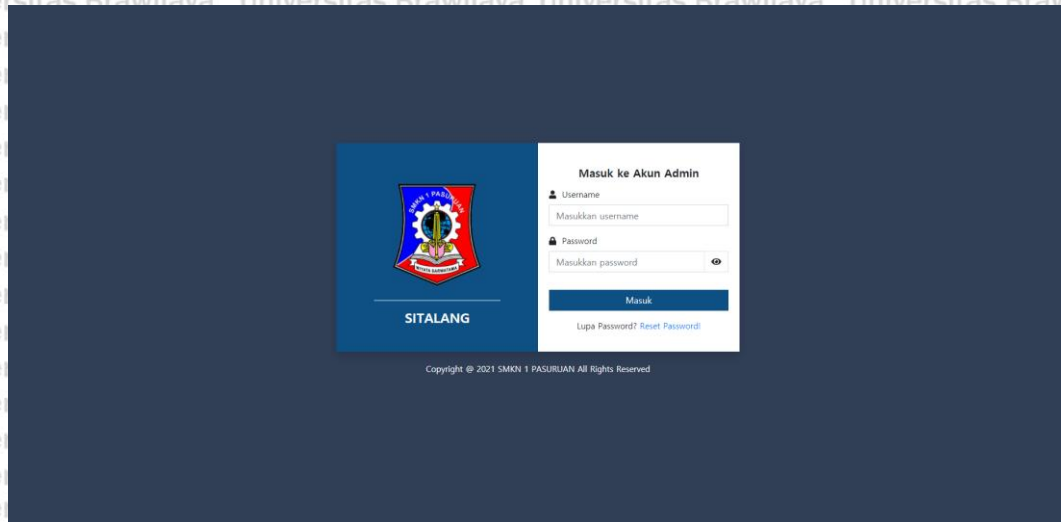
Terdapat total 8 kasus uji dari 3 fungsi yang diuji dengan rincian terdapat 3 kasus uji pada fungsi tambah poin pelanggaran, 3 kasus ui pada fungsi edit poin pelanggaran, dan 2 kasus uji pada fungsi hapus poin pelanggaran. Kasus uji yang telah didapatkan pada proses pengujian unit nantinya akan dijadikan dasar dalam penyusunan kasus uji pengujian validasi atau *validation testing* yang akan dilaksanakan pada tahap pengujian sistem.

4.3.7 Hasil Implementasi Antarmuka

Hasil implementasi antarmuka merupakan hasil representasi dari desain *wireframe* yang sudah dirancang pada tahap perancangan sistem. Hasil implementasi antarmuka aplikasi ini disusun dengan menggunakan *framework* Bootstrap dengan tambahan CSS dan Javascript. Terdapat penambahan elemen *breadcrumb* pada beberapa fungsional aktor guru untuk memberikan informasi kepada aktor alur berjalannya dari menu atau fungsional yang diakses. Berikut merupakan hasil dari implementasi antarmuka aplikasi Sistem Tata Tertib Pelanggaran (SiTalang).

4.3.7.1 Implementasi Antarmuka Masuk

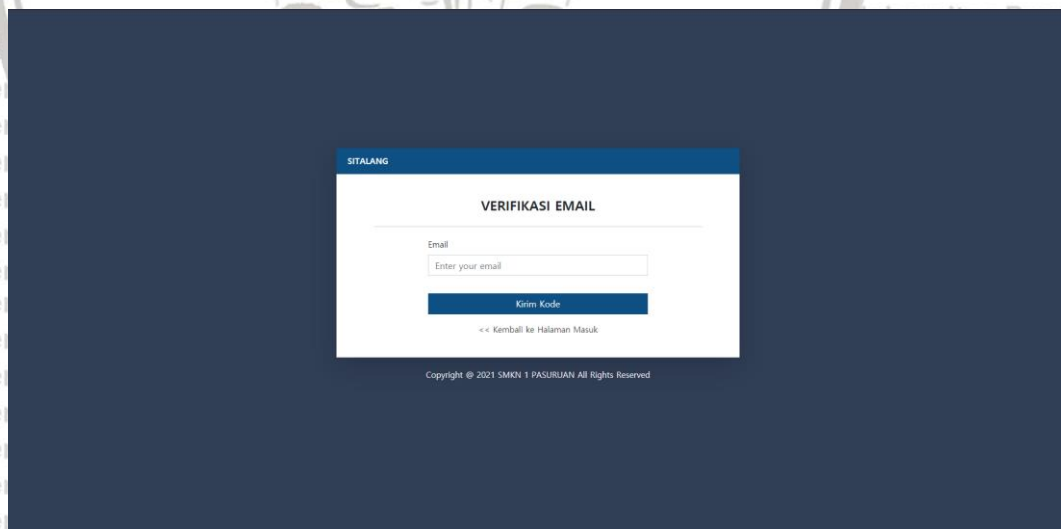
Gambar 4.52 merupakan hasil implementasi dari halaman masuk. Hasil implementasi Gambar 4.52 tersebut adalah bentuk representasi dari desain *wireframe* Gambar 4.33.



Gambar 4.52 Implementasi Antarmuka Masuk

4.3.7.2 Implementasi Antarmuka Verifikasi Email

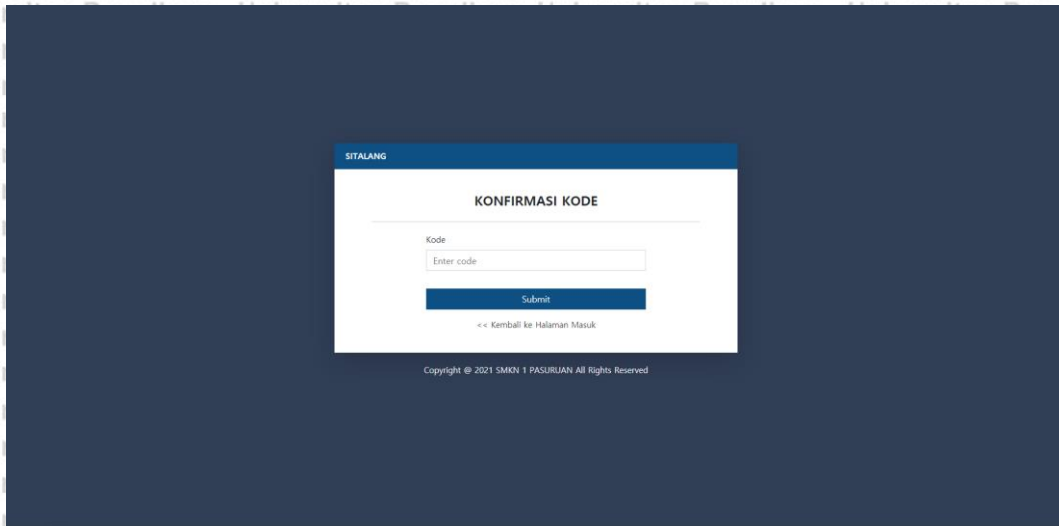
Gambar 4.53 merupakan hasil implementasi antarmuka dari halaman verifikasi *email*. Hasil implementasi antarmuka Gambar 4.53 tersebut adalah bentuk representasi dari *wireframe* gambar 4.34.



Gambar 4.53 Implementasi Antarmuka Verifikasi Email

4.3.7.3 Implementasi Antarmuka Verifikasi Kode

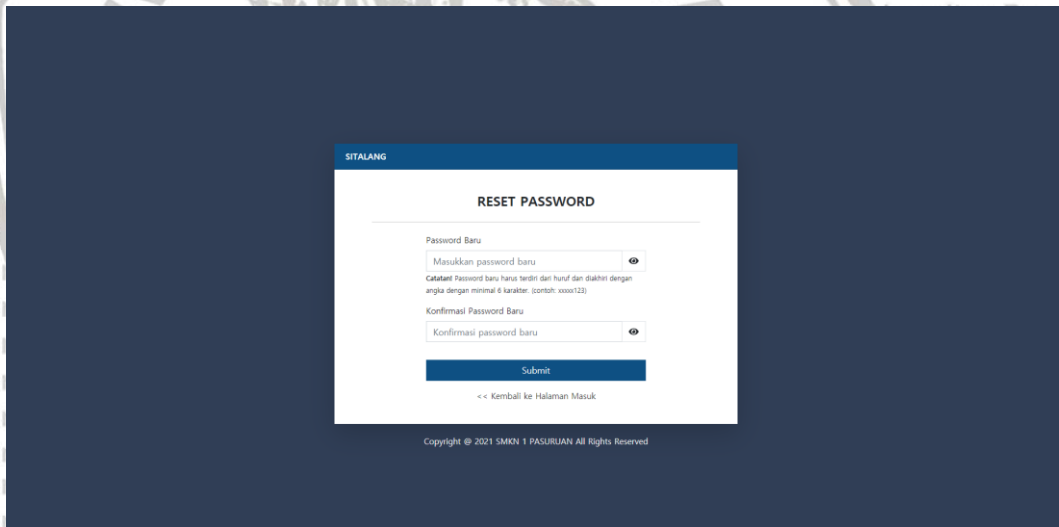
Gambar 4.54 merupakan bentuk representasi antarmuka dari *wireframe* Gambar 4.35 yang merupakan bentuk dari implementasi antarmuka dari halaman verifikasi kode.



Gambar 4.54 Implementasi Antarmuka Verifikasi Kode

4.3.7.4 Implementasi Antarmuka Reset *Password*

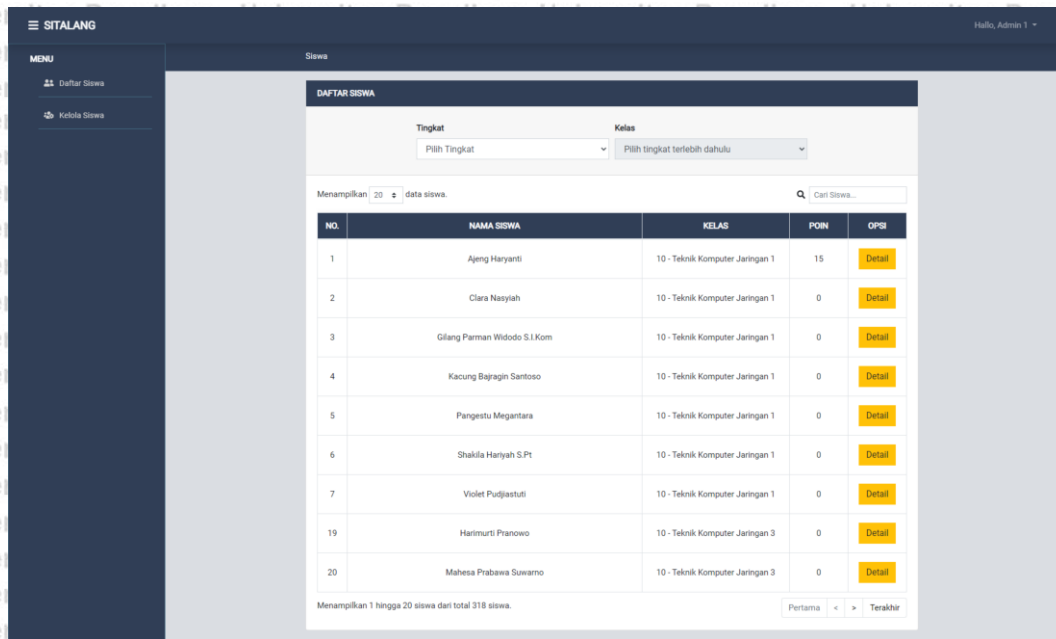
Gambar 4.55 merupakan hasil dari implementasi antarmuka Reset *Password*. Hasil implementasi Gambar 4.55 tersebut adalah bentuk representasi dari *wireframe* Gambar 4.36.



Gambar 4.55 Implementasi Antarmuka Reset *Password*

4.3.7.5 Implementasi Antarmuka Lihat Daftar Siswa

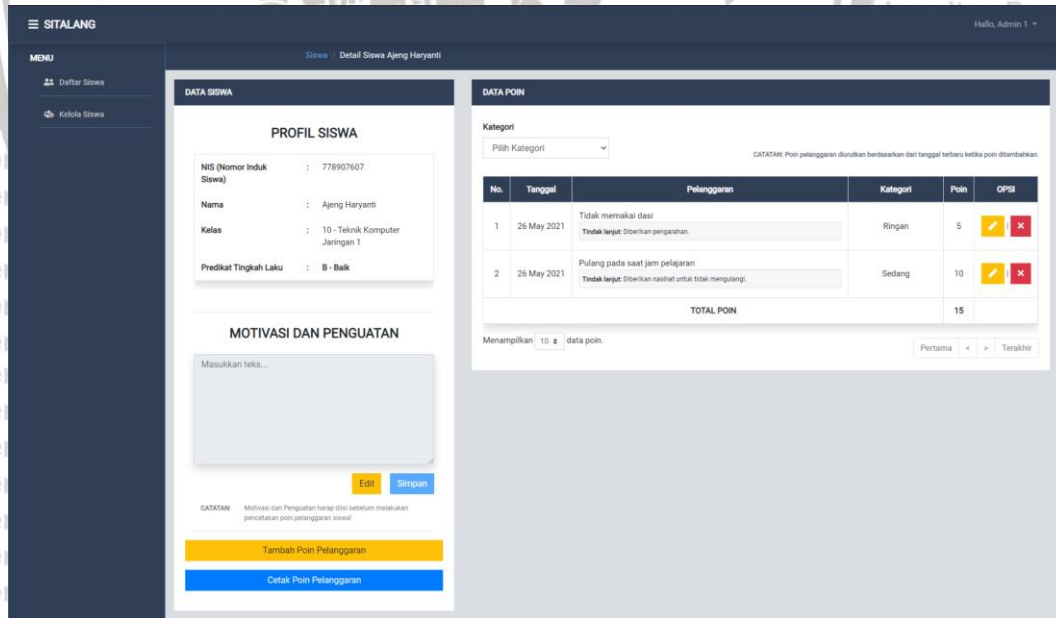
Gambar 4.56 merupakan hasil dari implementasi antarmuka halaman lihat daftar siswa. Hasil implementasi Gambar 4.56 tersebut adalah bentuk representasi dari *wireframe* Gambar 4.37.



Gambar 4.56 Implementasi Antarmuka Daftar Siswa

4.3.7.6 Implementasi Antarmuka Lihat Pelanggaran Siswa

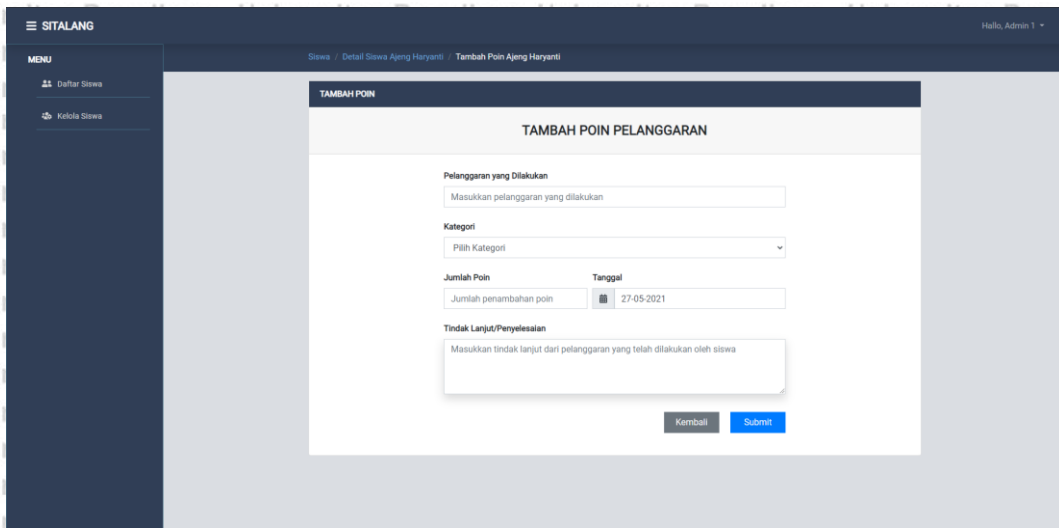
Gambar 4.57 merupakan bentuk representasi antarmuka dari *wireframe* Gambar 4.38 yang merupakan bentuk dari implementasi antarmuka halaman lihat pelanggaran siswa.



Gambar 4.57 Implementasi Antarmuka Lihat Pelanggaran Siswa

4.3.7.7 Implementasi Antarmuka Tambah Poin Pelanggaran

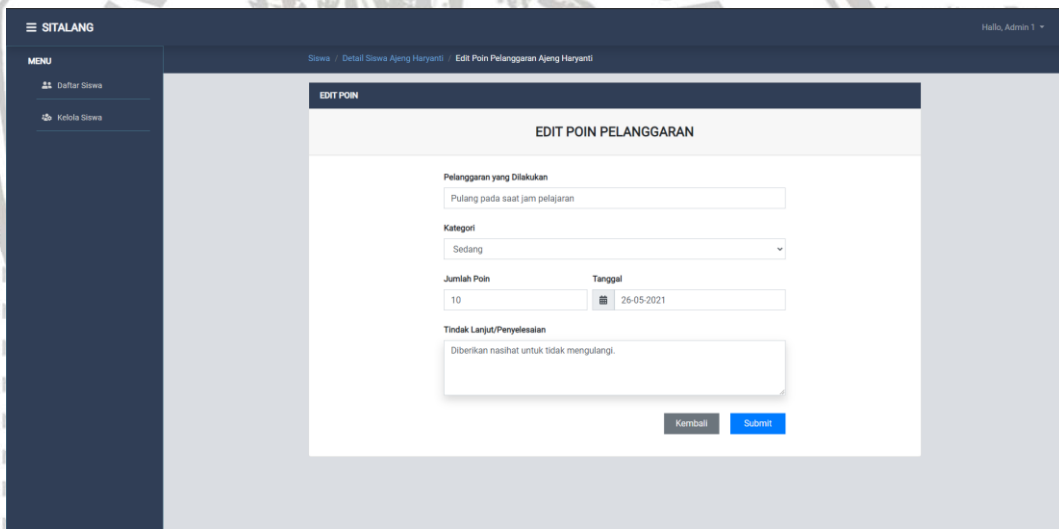
Gambar 4.58 merupakan bentuk representasi antarmuka dari *wireframe* Gambar 4.39 yang merupakan bentuk dari implementasi antarmuka halaman tambah poin pelanggaran.



Gambar 4.58 Implementasi Antarmuka Tambah Poin Pelanggaran

4.3.7.8 Implementasi Antarmuka Edit Poin Pelanggaran

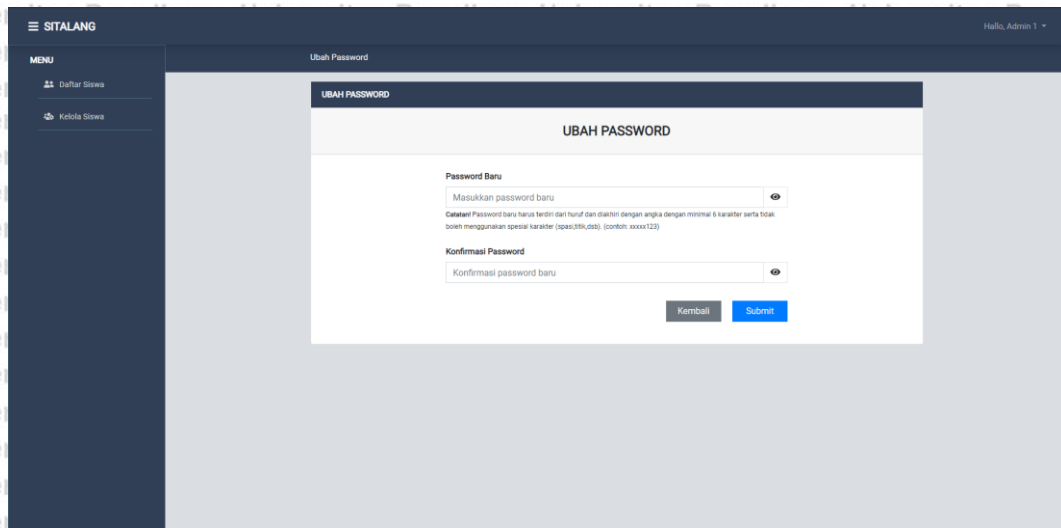
Gambar 4.59 merupakan hasil dari implementasi antarmuka edit poin pelanggaran. Hasil implementasi Gambar 4.59 tersebut adalah bentuk representasi dari *wireframe* Gambar 4.40.



Gambar 4.59 Implementasi Antarmuka Edit Poin Pelanggaran

4.3.7.9 Implementasi Antarmuka Ubah Password

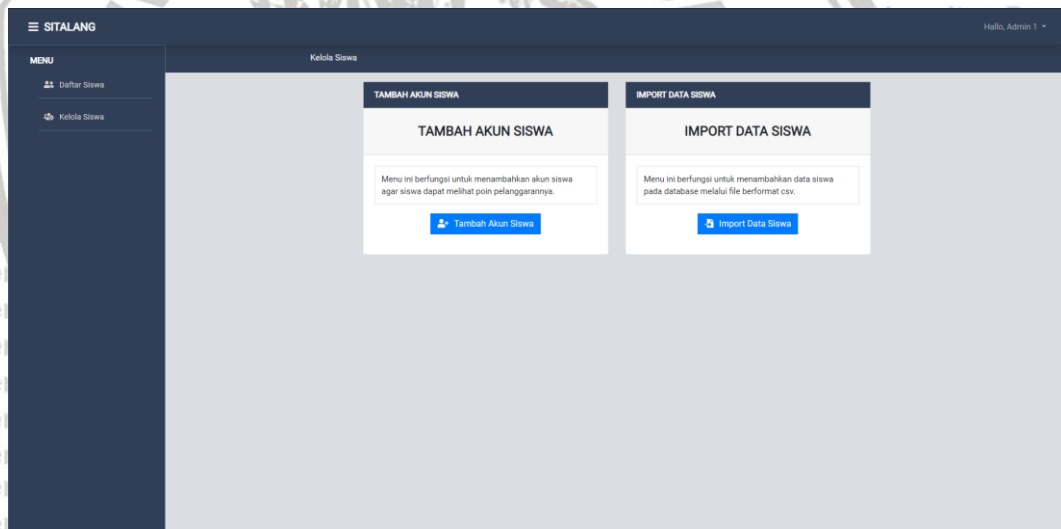
Gambar 4.60 merupakan hasil dari implementasi antarmuka halaman ubah password. Hasil implementasi Gambar 4.60 adalah bentuk representasi dari *wireframe* Gambar 4.41.



Gambar 4.60 Implementasi Antarmuka Ubah Password

4.3.7.10 Implementasi Antarmuka Halaman Kelola Siswa

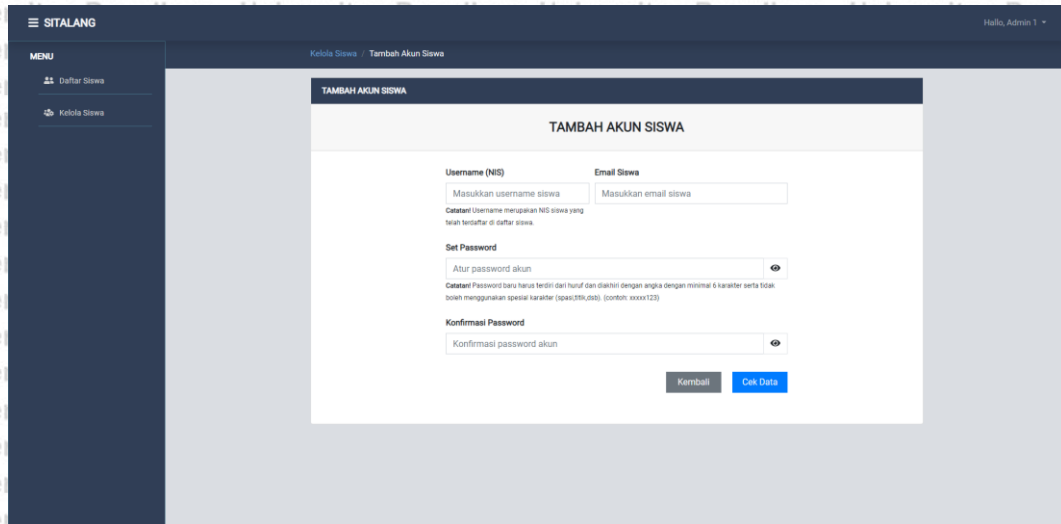
Gambar 4.61 merupakan bentuk representasi antarmuka dari *wireframe* Gambar 4.42 yang merupakan bentuk dari implementasi antarmuka halaman kelola siswa.



Gambar 4.61 Implementasi Antarmuka Halaman Kelola Siswa

4.3.7.11 Implementasi Antarmuka Tambah Akun Siswa

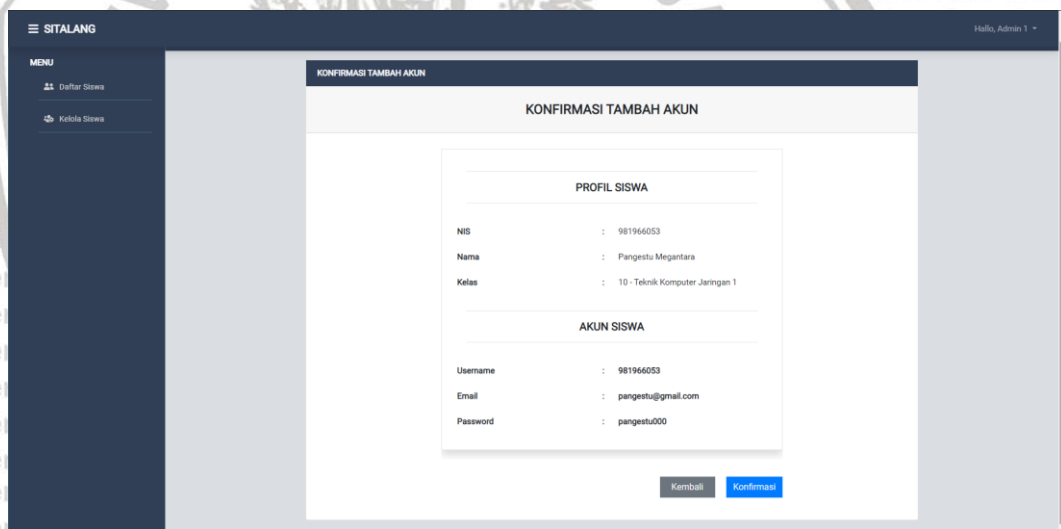
Gambar 4.62 merupakan hasil dari implementasi antarmuka halaman tambah akun siswa. Hasil implementasi Gambar 4.62 adalah bentuk representasi dari *wireframe* Gambar 4.43.



Gambar 4.62 Implementasi Antarmuka Tambah Akun Siswa

4.3.7.12 Implementasi Antarmuka Konfirmasi Tambah Akun Siswa

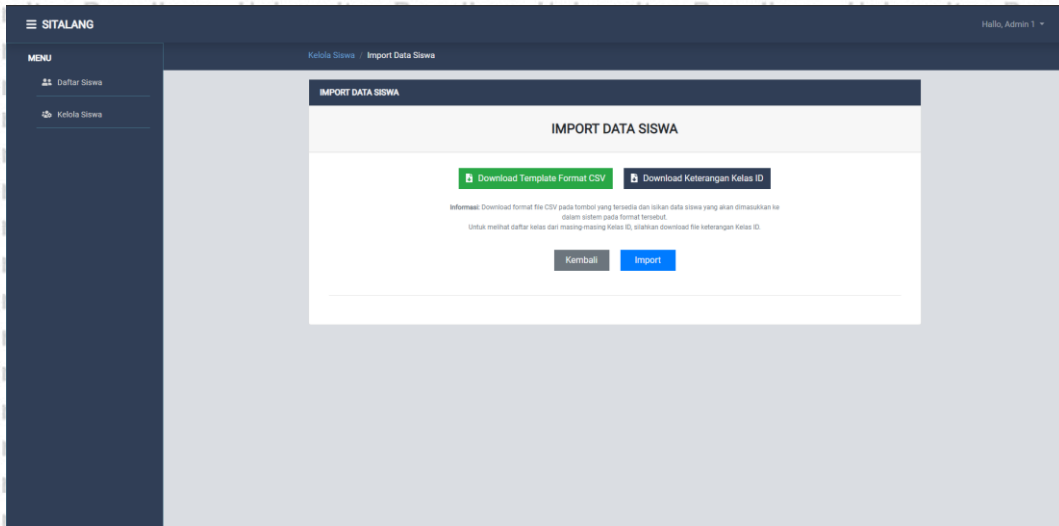
Gambar 4.63 merupakan bentuk representasi antarmuka dari *wireframe* Gambar 4.44 yang merupakan bentuk dari implementasi antarmuka halaman konfirmasi tambah akun siswa.



Gambar 4.63 Implementasi Antarmuka Konfirmasi Tambah Akun Siswa

4.3.7.13 Implementasi Antarmuka Tambah Data Siswa

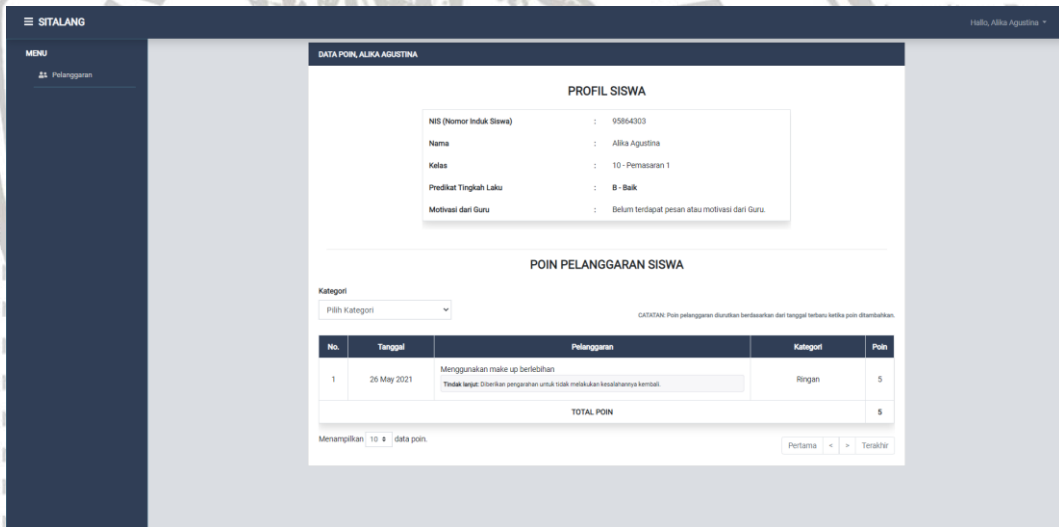
Gambar 4.64 merupakan hasil dari implementasi antarmuka halaman tambah data siswa. Hasil implementasi Gambar 4.64 adalah bentuk representasi dari *wireframe* Gambar 4.45.



Gambar 4.64 Implementasi Antarmuka Tambah Data Siswa

4.3.7.14 Implementasi Antarmuka Lihat Pelanggaran


Gambar 4.65 merupakan bentuk representasi antarmuka dari *wireframe* Gambar 4.46 yang merupakan bentuk dari implementasi antarmuka halaman lihat pelanggaran yang dapat diakses oleh aktor siswa.



Gambar 4.65 Implementasi Antarmuka Lihat Pelanggaran


4.3.7.15 Implementasi Antarmuka Hasil Cetak Poin Pelanggaran

Gambar 4.66 merupakan bentuk representasi dari desain *wireframe* yang terdapat Gambar 4.47. Gambar 4.66 merupakan hasil dari implementasi antarmuka dokumen menu cetak poin pelanggaran siswa dalam format pdf.



PEMERINTAH KOTA PASURUAN
DINAS PENDIDIKAN
SMK NEGERI 1 PASURUAN

Jalan Veteran, Kota Pasuruan Telp. (0341) 421380
 Website: <http://www.smkn1-pasuruan.sch.id>



POIN PELANGGARAN SISWA

NIS (Nomor Induk Siswa) : 778907807
Nama : Ajeng Haryanti
Kelas : 10 - Teknik Komputer Jaringan 1
Predikat Tingkah Laku : B - Baik

NO.	TANGGAL	PELANGGARAN	KATEGORI	POIN
1	26-05-2021	Tidak memakai dasi Tindak lanjut: Diberikan pengarahan.	Ringan	5
2	26-05-2021	Pulang pada saat jam pelajaran Tindak lanjut: Diberi nasihat untuk tidak mengulangi.	Sedang	10
TOTAL POIN				15

*Poin pelanggaran diurutkan berdasarkan dari tanggal terbaru ketika poin ditambahkan.

CATATAN!

Pasuruan, 17 Juni 2021
 Wakil Kepala Kesiswaan SMK N 1 Pasuruan

Setyo Wahyu Wicaksono., M.Pd.
 NIP. 19740829 200604 1 007

Halaman 1

<http://www.smkn1-pasuruan.sch.id>

Gambar 4.66 Implementasi Antarmuka Hasil Cetak Poin Pelanggaran

4.4 Hasil Pengujian Sistem

Pengujian dilakukan dengan menggunakan dua cara, yaitu pengujian validasi atau *validation testing* dan *user acceptance testing* (UAT). Pengujian validasi bertujuan untuk menguji fungsional-fungsional yang terdapat pada sistem apakah dapat dijalankan dengan baik atau tidak, sedangkan pengujian *user acceptance testing* (UAT) bertujuan untuk mengetahui apakah sistem yang dibangun (SiTalang) sudah sesuai dengan kebutuhan pengguna atau tidak. Berikut merupakan pembahasan dari pengujian yang dilakukan pada sistem yang dibangun (SiTalang).

4.4.1 Validation Testing

Pengujian validasi dilakukan dengan cara menguji seluruh fungsional yang terdapat pada sistem yang dilakukan oleh responden. Responden yang menguji aplikasi akan diberikan kuesioner yang berisi panduan pengujian dan validasi berupa *checklist* untuk memvalidasi apakah fungsional yang diuji telah sukses atau gagal. Pengujian validasi dilakukan kepada tujuh responden dengan rincian tiga guru dan empat siswa. Tiga guru tersebut merupakan guru yang bertugas pada bagian kedisiplinan yang menangani perilaku siswa di sekolah, sedangkan siswa yang menjadi responden merupakan siswa yang dipilih secara acak. Untuk detail hasil pengujian dapat dilihat pada Lampiran C dan Lampiran D. Hasil rekapitulasi pengujian validasi yang telah dilaksanakan adalah sebagai berikut.

4.4.1.1 Validation Testing Fungsional Masuk

- *Test Conditions:*
 1. Tidak dapat melakukan masuk jika *username* atau *password* guru salah.
 2. Dapat melakukan masuk jika *username* dan *password* guru benar.

Test Condition 1 – Tidak dapat melakukan masuk jika *username* dan *password* guru salah.

Tabel 4.58 Pengujian Validasi Fungsional Masuk Gagal

<i>Test Description</i>	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman <i>browser</i>.</p> <p><i>Actor:</i> Pengguna.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Mengakses alamat URL SiTalang (tatatertibpelanggaran.000webhostapp.com). 2. Memasukkan <i>username</i> atau <i>password</i> dengan data yang salah. 3. Menekan tombol masuk. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal masuk ke halaman utama aplikasi dan tetap berada di halaman masuk. 	Sukses

Tabel 4.58 Pengujian Validasi Fungsional Masuk Gagal (lanjutan)

<i>Test Description</i>	Sukses/Gagal
<p><i>Test Data:</i></p> <p>Guru:</p> <ul style="list-style-type: none"> • <i>Username:</i> admin1 • <i>Password:</i> admin111 <p>Siswa:</p> <ul style="list-style-type: none"> • <i>Username:</i> 95864303 • <i>Password:</i> salah123 	

Test Condition 2 - Dapat melakukan masuk jika *username* dan *password* guru benar.

Tabel 4.59 Pengujian Validasi Fungsional Masuk Berhasil

<i>Test Description</i>	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman <i>browser</i>.</p> <p><i>Actor:</i> Pengguna.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Mengakses alamat URL SiTalang (tatatertibpelanggaran.000webhostapp.com). 2. Memasukkan <i>username</i> atau <i>password</i> dengan data yang salah. 3. Menekan tombol masuk. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal masuk ke halaman utama aplikasi dan tetap berada di halaman masuk. <p><i>Test Data:</i></p> <p>Guru:</p> <ul style="list-style-type: none"> • <i>Username:</i> admin1 • <i>Password:</i> admin123 <p>Siswa:</p> <ul style="list-style-type: none"> • <i>Username:</i> 95864303 • <i>Password:</i> alika000 	Sukses

4.4.1.2 Validation Testing Reset Password

• **Test Conditions:**

1. Tidak dapat mengirimkan kode jika *email* pengguna salah.
2. Dapat mengirimkan kode jika *email* pengguna benar.
3. Tidak dapat mengakses halaman reset *password* jika kode konfirmasi salah.
4. Dapat mengakses halaman reset *password* jika kode konfirmasi benar.
5. Tidak dapat mengatur ulang informasi *password* jika masukan tidak valid.
6. Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.
7. Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru kurang dari 6 karakter.
8. Dapat mengatur ulang informasi *password* jika masukan valid.

Test Condition 1 – Tidak dapat mengirimkan kode jika *email* pengguna salah.

Tabel 4.60 Pengujian Validasi Verifikasi *Email* Salah

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman masuk.</p> <p><i>Actor:</i> Pengguna.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan <i>link</i> bertuliskan Reset Password!. 2. Memasukkan <i>email</i> dengan data yang salah. 3. Menekan tombol kirim kode. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal mengakses halaman konfirmasi kode dan tidak menerima kode konfirmasi melalui <i>email</i>. <p><i>Test Data:</i></p> <p>Guru:</p> <ul style="list-style-type: none"> • <i>Email:</i> email.salah@gmail.com <p>Siswa:</p> <ul style="list-style-type: none"> • <i>Email:</i> email.random@gmail.com 	<p>Sukses</p>

Test Condition 2 – Dapat mengirimkan kode jika *email* pengguna benar.

Tabel 4.61 Pengujian Validasi Verifikasi *Email* Benar

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman masuk.</p> <p><i>Actor:</i> Pengguna.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan <i>link</i> bertuliskan Reset Password!. 2. Memasukkan <i>email</i> dengan data yang benar. 3. Menekan tombol kirim kode. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor berhasil mengakses halaman konfirmasi kode dan menerima kode konfirmasi melalui <i>email</i>. <p><i>Test Data:</i></p> <p>Guru:</p> <ul style="list-style-type: none"> • <i>Email:</i> emailnya.sodiq@gmail.com <p>Siswa:</p> <ul style="list-style-type: none"> • <i>Email:</i> sodiq.jobseeker@gmail.com 	Sukses

Test Condition 3 – Tidak dapat mengakses halaman reset *password* jika kode konfirmasi salah.

Tabel 4.62 Pengujian Validasi Konfirmasi Kode Salah

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman konfirmasi kode.</p> <p><i>Actor:</i> Pengguna.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Memasukkan kode konfirmasi dengan data yang salah. 2. Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal mengakses halaman reset <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> • Kode: Tidak sesuai dengan kode yang diterima melalui <i>email (random)</i>. 	Sukses

Test Condition 4 – Dapat mengakses halaman reset *password* jika kode konfirmasi benar.

Tabel 4.63 Pengujian Validasi Konfirmasi Kode Benar

<i>Test Description</i>	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman konfirmasi kode.</p> <p><i>Actor:</i> Pengguna.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Memasukkan kode konfirmasi dengan data yang benar. Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor berhasil mengakses halaman reset <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> Kode: Sesuai dengan kode yang diterima melalui <i>email</i>. 	Sukses

Test Condition 5 – Tidak dapat mengatur ulang informasi *password* jika masukan tidak valid.

Tabel 4.64 Pengujian Validasi Reset *Password* Tidak Valid

<i>Test Description</i>	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman reset <i>password</i>.</p> <p><i>Actor:</i> Pengguna.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang tidak valid. Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal mengatur ulang informasi <i>password</i>. <p><i>Test Data:</i></p> <p>Guru:</p> <ul style="list-style-type: none"> <i>Password</i> baru: passwordbaru Konfirmasi <i>Password</i>: passwordbaru <p>Siswa:</p> <ul style="list-style-type: none"> <i>Password</i> baru: passwordtidakvalid Konfirmasi <i>Password</i>: passwordtidakvalid 	Sukses

Test Condition 6 – Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.

Tabel 4.65 Pengujian Validasi Reset Password Tidak Sesuai

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman reset <i>password</i>. <i>Actor:</i> Pengguna.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang berbeda. Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal mengatur ulang informasi <i>password</i>. <p><i>Test Data:</i></p> <p>Guru:</p> <ul style="list-style-type: none"> <i>Password</i> baru: passwordbaru123 Konfirmasi <i>Password</i>: passwordbaru456 <p>Siswa:</p> <ul style="list-style-type: none"> <i>Password</i> baru: passwordsiswa123 Konfirmasi <i>Password</i>: passwordsiswa456 	Sukses

Test Condition 7 – Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru kurang dari 6 karakter.

Tabel 4.66 Pengujian Validasi Reset Password 6 Karakter

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman reset <i>password</i>. <i>Actor:</i> Pengguna.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan kurang dari 6 karakter. Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal mengatur ulang informasi <i>password</i>. <p><i>Test Data:</i></p> <p>Guru & Siswa:</p> <ul style="list-style-type: none"> <i>Password</i> baru: tes12 Konfirmasi <i>Password</i>: tes12 	Sukses

Test Condition 8 – Dapat mengatur ulang informasi *password* jika masukan valid.

Tabel 4.67 Pengujian Validasi Reset *Password* Berhasil

Test Description	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman reset <i>password</i>.</p> <p>Actor: Pengguna.</p> <p>Scenario Test:</p> <ol style="list-style-type: none"> 1. Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang valid. 2. Menekan tombol <i>submit</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor berhasil mengatur ulang informasi <i>password</i>. <p>Test Data:</p> <p>Guru:</p> <ul style="list-style-type: none"> • <i>Password</i> baru: passwordbaru123 • Konfirmasi <i>Password</i>: passwordbaru123 <p>Siswa:</p> <ul style="list-style-type: none"> • <i>Password</i> baru: passwordbaru111 • Konfirmasi <i>Password</i>: passwordbaru111 	<p>Sukses</p>



4.4.1.3 Validation Testing Ubah Password

- **Test Conditions:**
 1. Tidak dapat mengubah informasi *password* jika masukan tidak valid.
 2. Tidak dapat mengubah informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.
 3. Tidak dapat mengubah informasi *password* jika masukan *password* baru kurang dari 6 karakter.
 4. Dapat mengubah informasi *password* jika masukan valid.

Test Condition 1 – Tidak dapat mengubah informasi *password* jika masukan tidak valid.

Tabel 4.68 Pengujian Validasi Ubah Password Tidak Valid

Test Description	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman utama aplikasi.</p> <p>Actor: Guru, Siswa.</p> <p>Scenario Test:</p> <ol style="list-style-type: none"> 1. Menekan tombol ubah <i>password</i> pada bagian <i>navigation bar</i>. 2. Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang tidak valid. 3. Menekan tombol <i>submit</i>. 4. Menekan tombol <i>submit</i> pada <i>modal</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal mengubah informasi <i>password</i>. <p>Test Data:</p> <p>Guru:</p> <ul style="list-style-type: none"> • <i>Password</i> baru: ubahpassword • Konfirmasi <i>Password</i>: ubahpassword <p>Siswa:</p> <ul style="list-style-type: none"> • <i>Password</i> baru: pwtidakvalid • Konfirmasi <i>Password</i>: pwtidakvalid 	<p>Sukses</p>

Test Condition 2 – Tidak dapat mengubah informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.

Tabel 4.69 Pengujian Validasi Ubah *Password* Tidak Sesuai

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman utama aplikasi. <i>Actor:</i> Guru, Siswa. <i>Scenario Test:</i></p> <ol style="list-style-type: none"> Menekan tombol ubah <i>password</i> pada bagian <i>navigation bar</i>. Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang berbeda. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal mengubah informasi <i>password</i>. <p><i>Test Data:</i></p> <p>Guru:</p> <ul style="list-style-type: none"> <i>Password</i> baru: ubahpassword000 Konfirmasi <i>Password:</i> ubahpassword123 <p>Siswa:</p> <ul style="list-style-type: none"> <i>Password</i> baru: ubahpassword000 Konfirmasi <i>Password:</i> ubahbedapassword123 	<p>Sukses</p>

Test Condition 3 – Tidak dapat mengubah informasi *password* jika masukan *password* baru kurang dari 6 karakter.

Tabel 4.70 Pengujian Validasi Ubah *Password* 6 Karakter

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman utama aplikasi. <i>Actor:</i> Guru, Siswa. <i>Scenario Test:</i></p> <ol style="list-style-type: none"> Menekan tombol ubah <i>password</i> pada bagian <i>navigation bar</i>. Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan kurang dari 6 karakter. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada <i>modal</i>. <p><i>Expected Result:</i></p>	<p>Sukses</p>

Tabel 4.70 Pengujian Validasi Ubah Password 6 Karakter (lanjutan)

Test Description	Sukses/Gagal
<p>• Aktor gagal mengubah informasi <i>password</i>.</p> <p>Test Data:</p> <p>Guru:</p> <ul style="list-style-type: none"> • Password baru: adm12 • Konfirmasi Password: adm12 <p>Siswa:</p> <ul style="list-style-type: none"> • Password baru: pwd12 • Konfirmasi Password: pwd12 	Sukses

Test Condition 4 – Dapat mengubah informasi *password* baru jika masukan valid.

Tabel 4.71 Pengujian Validasi Ubah Password Berhasil

Test Description	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman utama aplikasi.</p> <p>Actor: Guru, Siswa.</p> <p>Scenario Test:</p> <ol style="list-style-type: none"> 1. Menekan tombol ubah <i>password</i> pada bagian <i>navigation bar</i>. 2. Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang valid. 3. Menekan tombol <i>submit</i>. 4. Menekan tombol <i>submit</i> pada <i>modal</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor berhasil mengubah informasi <i>password</i>. <p>Test Data:</p> <p>Guru:</p> <ul style="list-style-type: none"> • Password baru: admin123 • Konfirmasi Password: admin123 <p>Siswa:</p> <ul style="list-style-type: none"> • Password baru: alika000 • Konfirmasi Password: alika000 	Sukses

4.4.1.4 Validation Testing Lihat Daftar Siswa

- **Test Conditions:**

1. Dapat melihat daftar siswa.

Test Condition 1 – Dapat melihat daftar siswa.

Tabel 4.72 Pengujian Validasi Lihat Daftar Siswa

Test Description	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman utama aplikasi.</p> <p>Actor: Guru.</p> <p>Scenario Test:</p> <ol style="list-style-type: none"> 1. Melakukan masuk. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor berhasil melihat daftar siswa. <p>Test Data: -</p>	Sukses

4.4.1.5 Validation Testing Lihat Pelanggaran Siswa

- **Test Conditions:**

1. Dapat melihat rincian pelanggaran siswa sesuai dengan siswa yang dipilih.

Test Condition 1 – Dapat melihat rincian pelanggaran siswa sesuai dengan siswa yang dipilih.

Tabel 4.73 Pengujian Validasi Lihat Pelanggaran Siswa

Test Description	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman daftar siswa.</p> <p>Actor: Guru.</p> <p>Scenario Test:</p> <ol style="list-style-type: none"> 1. Menekan tombol detail yang berada pada baris siswa. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor berhasil melihat rincian pelanggaran siswa. <p>Test Data: -</p>	Sukses

4.4.1.6 Validation Testing Tambah Poin Pelanggaran

• *Test Conditions:*

1. Tidak dapat menambah poin pelanggaran jika ada masukan tambah poin pelanggaran yang tidak valid.
2. Tidak dapat menambah poin pelanggaran jika ada masukan tambah poin pelanggaran yang tidak diisi atau kosong.
3. Dapat menambah poin pelanggaran jika masukan tambah poin pelanggaran valid.

Test Condition 1 – Tidak dapat menambah poin pelanggaran jika ada masukan tambah poin pelanggaran yang tidak valid.

Tabel 4.74 Pengujian Validasi Tambah Poin Tidak Valid

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman lihat pelanggaran siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan tombol tambah poin pelanggaran pada bagian data siswa. 2. Memasukkan jumlah poin dengan masukan yang tidak valid. 3. Menekan tombol <i>submit</i>. 4. Menekan tombol <i>submit</i> pada <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal menambah poin pelanggaran siswa. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Sepatu tidak sesuai dengan ketentuan • Kategori: Ringan • Jumlah Poin: 0 • Tanggal: 19 April 2021 • Tindak Lanjut: Diberikan pengarahan agar tidak mengulangi pelanggaran kembali. 	<p>Sukses</p>

Test Condition 2 – Tidak dapat menambah poin pelanggaran jika ada masukan tambah poin pelanggaran yang tidak diisi atau kosong.

Tabel 4.75 Pengujian Validasi Tambah Poin Kosong

<i>Test Description</i>	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman lihat pelanggaran siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Menekan tombol tambah poin pelanggaran pada bagian data siswa. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada bagian <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal menambah poin pelanggaran siswa. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> Pelanggaran yang Dilakukan: - Kategori: - Jumlah Poin: - Tindak Lanjut: - 	Sukses

Test Condition 3 – Dapat menambah poin pelanggaran jika masukan tambah poin pelanggaran valid.

Tabel 4.76 Pengujian Validasi Tambah Poin Berhasil

<i>Test Description</i>	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman lihat pelanggaran siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Menekan tombol tambah poin pelanggaran pada bagian data siswa. Memasukkan pelanggaran yang dilakukan, kategori, jumlah poin, tanggal, dan tindak lanjut dengan masukan yang valid. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada bagian <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor berhasil menambah poin pelanggaran siswa. <p><i>Test Data:</i></p>	Sukses

Tabel 4.76 Pengujian Validasi Tambah Poin Berhasil (lanjutan)

Test Description	Sukses/Gagal
<ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Tidak memakai atribut sekolah dengan lengkap • Kategori: Ringan • Jumlah Poin: 5 • Tanggal: 20 April 2021 • Tindak Lanjut: Diberikan pengarahan agar tidak mengulangi pelanggaran kembali. 	

4.4.1.7 Validation Testing Edit Poin Pelanggaran

- **Test Conditions:**
 1. Tidak dapat mengubah informasi poin pelanggaran jika perubahan informasi poin pelanggaran tidak valid.
 2. Tidak dapat mengubah poin pelanggaran jika perubahan informasi poin pelanggaran adalah menghilangkan nilai informasi poin pelanggaran.
 3. Dapat mengubah informasi poin pelanggaran jika perubahan informasi poin pelanggaran valid.

Test Condition 1 – Tidak dapat mengubah informasi poin pelanggaran jika perubahan informasi poin pelanggaran tidak valid.

Tabel 4.77 Pengujian Validasi Edit Poin Tidak Valid

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman lihat pelanggaran siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan tombol edit pada bagian daftar pelanggaran siswa. 2. Mengubah informasi jumlah poin dengan perubahan nilai yang tidak valid. 3. Menekan tombol <i>submit</i>. 4. Menekan tombol <i>submit</i> pada bagian <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal mengubah informasi poin pelanggaran. <p><i>Test Data:</i></p> <p>Data Awal:</p> <ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Tidak memakai atribut sekolah dengan lengkap • Kategori: Ringan 	Sukses

Tabel 4.77 Pengujian Validasi Edit Poin Tidak Valid (lanjutan)

<i>Test Description</i>	Sukses/Gagal
<ul style="list-style-type: none"> • Jumlah Poin: 5 • Tanggal: 20 April 2021 • Tindak Lanjut: Diberikan pengarahan agar tidak mengulangi pelanggaran kembali. <p>Data Edit:</p> <ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Tidak memakai atribut sekolah dengan lengkap • Kategori: Ringan • Jumlah Poin: 0 • Tanggal: 20 April 2021 • Tindak Lanjut: Diberikan pengarahan agar tidak mengulangi pelanggaran kembali. 	

Test Condition 2 – Tidak dapat mengubah poin pelanggaran jika perubahan informasi poin pelanggaran adalah menghilangkan nilai informasi poin pelanggaran.

Tabel 4.78 Pengujian Validasi Edit Poin Kosong

<i>Test Description</i>	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman lihat pelanggaran siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan tombol edit pada bagian daftar pelanggaran siswa. 2. Mengubah informasi atau menghapus nilai pada edit teks pelanggaran yang dilakukan dan jumlah poin. 3. Menekan tombol <i>submit</i>. 4. Menekan tombol <i>submit</i> pada bagian <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal mengubah informasi poin pelanggaran. <p><i>Test Data:</i></p> <p>Data Awal:</p> <ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Tidak memakai atribut sekolah dengan lengkap • Kategori: Ringan • Jumlah Poin: 5 • Tanggal: 20 April 2021 	Sukses

Tabel 4.78 Pengujian Validasi Edit Poin Kosong (lanjutan)

Test Description	Sukses/Gagal
<ul style="list-style-type: none"> • Tindak Lanjut: Diberikan pengarahan agar tidak mengulangi pelanggaran kembali. <p>Data Edit:</p> <ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: - • Kategori: Ringan • Jumlah Poin: - • Tanggal: 20 April 2021 • Tindak Lanjut: - 	

Test Condition 3 – Dapat mengubah informasi poin pelanggaran jika perubahan informasi poin pelanggaran valid.

Tabel 4.79 Pengujian Validasi Edit Poin Berhasil

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman lihat pelanggaran siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan tombol edit pada bagian daftar pelanggaran siswa. 2. Mengubah informasi pelanggaran yang dilakukan, kategori, jumlah poin, dan tanggal dengan perubahan informasi yang valid. 3. Menekan tombol <i>submit</i>. 4. Menekan tombol <i>submit</i> pada bagian <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor berhasil mengubah informasi poin pelanggaran. <p><i>Test Data:</i></p> <p>Data Awal:</p> <ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Tidak memakai atribut sekolah dengan lengkap • Kategori: Ringan • Jumlah Poin: 5 • Tanggal: 20 April 2021 • Tindak Lanjut: Diberikan pengarahan agar tidak mengulangi pelanggaran kembali. <p>Data Edit:</p>	<p>Sukses</p>

Tabel 4.79 Pengujian Validasi Edit Poin Berhasil (lanjutan)

Test Description	Sukses/Gagal
<ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Tidak memakai topi dan terlambat • Kategori: Sedang • Jumlah Poin: 10 • Tanggal: 21 April 2021 • Tindak Lanjut: Diperintahkan untuk membersihkan halaman sekolah. 	

4.4.1.8 Validation Testing Hapus Poin Pelanggaran

• *Test Conditions:*

1. Tidak dapat menghapus poin pelanggaran sesuai dengan data yang dipilih jika membatalkan proses hapus poin pelanggaran.
2. Dapat menghapus poin pelanggaran sesuai dengan data yang dipilih.

Test Condition 1 – Tidak dapat menghapus poin pelanggaran sesuai dengan data yang dipilih jika membatalkan proses hapus poin pelanggaran.

Tabel 4.80 Pengujian Validasi Hapus Poin Gagal

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman lihat pelanggaran siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan tombol hapus pada bagian daftar pelanggaran siswa. 2. Menekan tombol <i>close</i> pada bagian <i>modal</i> konfirmasi. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal menghapus poin pelanggaran. <p><i>Test Data:</i> -</p>	Sukses

Test Condition 2 – Dapat menghapus poin pelanggaran sesuai dengan data yang dipilih.

Tabel 4.81 Pengujian Validasi Hapus Poin Berhasil

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman lihat pelanggaran siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Menekan tombol hapus pada bagian daftar pelanggaran siswa. Menekan tombol submit pada bagian modal konfirmasi. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor berhasil menghapus poin pelanggaran. <p><i>Test Data:</i> -</p>	Sukses

4.4.1.9 Validation Testing Cetak Poin Pelanggaran

- Test Conditions:*
 - Dapat mencetak atau menyimpan poin pelanggaran.

Test Condition 1 – Dapat mencetak atau menyimpan poin pelanggaran.

Tabel 4.82 Pengujian Validasi Cetak Poin Pelanggaran

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman lihat pelanggaran siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Menekan tombol cetak poin pelanggaran pada bagian data siswa. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor berhasil mencetak atau menyimpan poin pelanggaran. <p><i>Test Data:</i> -</p>	Sukses

4.4.1.10 Validation Testing Tambah Motivasi

- Test Conditions:*
 - Tidak dapat menyimpan motivasi jika membatalkan proses tambah motivasi.
 - Dapat menambah motivasi.

Test Condition 1 – Tidak dapat menyimpan motivasi jika membatalkan proses tambah motivasi.

Tabel 4.83 Pengujian Validasi Tambah Motivasi Gagal

<i>Test Description</i>	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman lihat pelanggaran siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan tombol edit pada bagian motivasi dan penguatan. 2. Memasukkan teks motivasi. 3. Menekan tombol <i>cancel</i> pada bagian motivasi dan penguatan. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal menambah motivasi. <p><i>Test Data:</i> -</p>	Sukses

Test Condition 2 – Dapat menambah motivasi.

Tabel 4.84 Pengujian Validasi Tambah Motivasi Berhasil

<i>Test Description</i>	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman lihat pelanggaran siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan tombol edit pada bagian motivasi dan penguatan. 2. Memasukkan teks motivasi. 3. Menekan tombol simpan pada bagian motivasi dan penguatan. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor berhasil menambah motivasi. <p><i>Test Data:</i> -</p>	Sukses

4.4.1.11 Validation Testing Tambah Akun Siswa

• *Test Conditions:*

1. Tidak dapat menambah akun siswa jika NIS atau *email* tidak valid.
2. Tidak dapat menambah akun siswa jika *password* tidak valid.

3. Dapat mengakses halaman konfirmasi tambah akun siswa jika informasi tambah akun siswa valid.
4. Tidak dapat menambahkan akun siswa jika pada halaman konfirmasi akun aktor tidak mengkonfirmasi tambah akun siswa.
5. Dapat menambahkan akun siswa jika pada halaman konfirmasi akun siswa aktor mengkonfirmasi tambah akun siswa.

Test Condition 1 – Tidak dapat menambah akun siswa jika NIS atau *email* tidak valid.

Tabel 4.85 Pengujian Validasi Tambah Akun Siswa Username atau Email Tidak Valid

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman kelola siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan tombol tambah akun siswa pada bagian tambah akun siswa. 2. Memasukkan NIS atau <i>email</i> dengan data yang salah. 3. Memasukkan <i>set password</i> dan konfirmasi <i>password</i>. 4. Menekan tombol cek data. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal mengakses halaman konfirmasi tambah akun siswa. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> • NIS: 95864303 • <i>Email:</i> sodiq.jobseeker@gmail.com • <i>Set Password:</i> password123 • <i>Konfirmasi Password:</i> password123 	Sukses

Test Condition 2 – Tidak dapat menambah akun siswa jika *password* tidak valid.

Tabel 4.86 Pengujian Validasi Tambah Akun Siswa Password Tidak Valid

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman kelola siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan tombol tambah akun siswa pada bagian tambah akun siswa. 2. Memasukkan NIS atau <i>email</i>. 	Sukses

Tabel 4.86 Pengujian Validasi Tambah Akun Siswa Password Tidak Valid (lanjutan)

Test Description	Sukses/Gagal
<p>3. Memasukkan <i>set password</i> dan konfirmasi <i>password</i> dengan data yang salah.</p> <p>4. Menekan tombol cek data.</p> <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal mengakses halaman konfirmasi tambah akun siswa. <p>Test Data:</p> <ul style="list-style-type: none"> • NIS: 419392974 • Email: clara@gmail.com • Set Password: pwd12 • Konfirmasi Password: pwd12 	

Test Condition 3 – Dapat mengakses halaman konfirmasi tambah akun siswa jika informasi tambah akun siswa valid.

Tabel 4.87 Pengujian Validasi Tambah Akun Siswa Informasi Valid

Test Description	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman kelola siswa.</p> <p>Actor: Guru.</p> <p>Scenario Test:</p> <ol style="list-style-type: none"> 1. Menekan tombol tambah akun siswa pada bagian tambah akun siswa. 2. Memasukkan NIS atau <i>email</i>. 3. Memasukkan <i>set password</i> dan konfirmasi <i>password</i>. 4. Menekan tombol cek data. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal mengakses halaman konfirmasi tambah akun siswa. <p>Test Data:</p> <ul style="list-style-type: none"> • NIS: 419392974 • Email: clara@gmail.com • Set Password: clara000 • Konfirmasi Password: clara000 	<p>Sukses</p>

Test Condition 4 – Tidak dapat menambahkan akun siswa jika pada halaman konfirmasi akun aktor tidak mengkonfirmasi tambah akun siswa.

Tabel 4.88 Pengujian Validasi Konfirmasi Tambah Akun Siswa Gagal

<i>Test Description</i>	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman konfirmasi tambah akun siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Menekan tombol konfirmasi pada bagian konfirmasi tambah akun siswa. Menekan tombol <i>close</i> pada bagian <i>modal</i> konfirmasi. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal menambah akun siswa. <p><i>Test Data:</i> -</p>	Sukses

Test Condition 5 – Dapat menambahkan akun siswa jika pada halaman konfirmasi akun siswa aktor mengkonfirmasi tambah akun siswa.

Tabel 4.89 Pengujian Validasi Konfirmasi Tambah Akun Siswa Berhasil

<i>Test Description</i>	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman konfirmasi tambah akun siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Menekan tombol konfirmasi pada bagian konfirmasi tambah akun siswa. Menekan tombol <i>submit</i> pada bagian <i>modal</i> konfirmasi. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor berhasil menambah akun siswa. <p><i>Test Data:</i> -</p>	Sukses

4.4.1.12 Validation Testing Tambah Data Siswa

- Test Conditions:*

- Tidak dapat menambah data siswa jika format *file* tidak valid.
- Tidak dapat menambah data siswa jika data siswa tidak valid.
- Dapat menambah data siswa jika format *file* dan data siswa valid.
- Tidak dapat menambah data siswa jika terdapat data NIS yang sama pada *database*.

Test Condition 1 – Tidak dapat menambah data siswa jika format *file* tidak valid.

Tabel 4.90 Pengujian Validasi Tambah Data Siswa Format File Tidak Valid

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman kelola siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa. Menekan tombol <i>import</i> pada bagian <i>import</i> data siswa. Menekan tombol pilih <i>file</i> pada <i>modal import</i> data siswa. Memilih <i>file</i> berformat TXT. Menekan tombol <i>import</i> pada <i>modal import</i> data siswa. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal menambah data siswa. <p><i>Test Data:</i> -</p>	Sukses

Test Condition 2 – Tidak dapat menambah data siswa jika data siswa tidak valid.

Tabel 4.91 Pengujian Validasi Tambah Data Siswa Data Siswa Tidak Valid

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman kelola siswa.</p> <p><i>Actor:</i> Guru.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> Menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa. Menekan tombol <i>import</i> pada bagian <i>import</i> data siswa. Menekan tombol pilih <i>file</i> pada <i>modal import</i> data siswa. Memilih <i>file</i> bernama Data Siswa Tidak Valid. Menekan tombol <i>import</i> pada <i>modal import</i> data siswa. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal menambah data siswa. <p><i>Test Data:</i> -</p>	Sukses

Test Condition 3 – Dapat menambah data siswa jika format *file* dan data siswa valid.

Tabel 4.92 Pengujian Validasi Tambah Data Siswa Berhasil

Test Description	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman kelola siswa.</p> <p>Actor: Guru.</p> <p>Scenario Test:</p> <ol style="list-style-type: none"> 1. Menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa. 2. Menekan tombol <i>import</i> pada bagian <i>import</i> data siswa. 3. Menekan tombol pilih <i>file</i> pada <i>modal import</i> data siswa. 4. Memilih <i>file</i> bernama Data Siswa Valid. 5. Menekan tombol <i>import</i> pada <i>modal import</i> data siswa. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor berhasil menambah data siswa. <p>Test Data: -</p>	Sukses

Test Condition 4 – Tidak dapat menambah data siswa jika terdapat data NIS yang sama pada *database*.

Tabel 4.93 Pengujian Validasi Tambah Data Siswa Duplikasi Data

Test Description	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman kelola siswa.</p> <p>Actor: Guru.</p> <p>Scenario Test:</p> <ol style="list-style-type: none"> 1. Menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa. 2. Menekan tombol <i>import</i> pada bagian <i>import</i> data siswa. 3. Menekan tombol pilih <i>file</i> pada <i>modal import</i> data siswa. 4. Memilih <i>file</i> bernama Data Siswa Valid. 5. Menekan tombol <i>import</i> pada <i>modal import</i> data siswa. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal menambah data siswa. <p>Test Data: -</p>	Sukses

4.4.1.13 Validation Testing Keluar

- **Test Conditions:**

1. Dapat keluar dari sistem.

Test Condition 1 – Dapat keluar dari sistem.

Tabel 4.94 Pengujian Validasi Keluar

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman utama aplikasi. <i>Actor:</i> Guru, Siswa.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Menekan tombol keluar pada bagian <i>navigation bar</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor berhasil keluar dari sistem. <p><i>Test Data:</i> -</p>	Sukses

4.4.1.14 Validation Testing Lihat Pelanggaran

- **Test Conditions:**

1. Dapat melihat data siswa dan informasi poin pelanggaran siswa sesuai data siswa yang melakukan masuk.

Test Condition 1 – Dapat melihat data siswa dan informasi poin pelanggaran siswa sesuai data siswa yang melakukan masuk.

Tabel 4.95 Pengujian Validasi Lihat Pelanggaran

Test Description	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman utama aplikasi. <i>Actor:</i> Siswa.</p> <p><i>Scenario Test:</i></p> <ol style="list-style-type: none"> 1. Melakukan masuk. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor berhasil melihat data siswa dan informasi poin pelanggaran siswa yang sesuai dengan siswa yang melakukan masuk. <p><i>Test Data:</i> -</p>	Sukses

4.4.2 User Acceptance Testing

Pengujian *user acceptance testing* (UAT) bertujuan untuk mengetahui apakah aplikasi yang telah dibangun dapat diterima oleh pengguna atau tidak. Untuk mengetahui apakah aplikasi yang telah dibangun dapat diterima atau tidak oleh pengguna yaitu dengan menggunakan pernyataan-pernyataan yang akan

diberikan kepada responden pada saat pengujian dilaksanakan. Proses penyusunan pernyataan yang digunakan pada pengujian UAT ini didasarkan dari kriteria penerimaan menurut Hambling dan van Goethem (2014) yang telah dijelaskan pada Bab 2 sub bab 2.6.4.2.

Untuk mengetahui persepsi penerimaan dari pengguna, peneliti menggunakan skala Likert dengan empat skala yaitu sangat setuju, setuju, tidak setuju, dan sangat tidak setuju. Penggunaan dari empat skala Likert tersebut dilandasi karena pengembang ingin menghindari jawaban netral dari responden. Interpretasi dari skala Likert dengan empat skala tersebut dijelaskan pada Tabel 4.96 yang diadaptasi dari tabel interpretasi milik Wardhono (2015).

Tabel 4.96 Interpretasi Skala Likert

Skor Likert	Interpretasi Skor	Tingkat Persetujuan
1	0% - 24,99%	Sangat Tidak Setuju
2	25% - 49,99%	Tidak Setuju
3	50% - 74,99%	Setuju
4	75% - 100%	Sangat Setuju

Pengujian UAT dilakukan kepada tujuh responden dengan rincian empat siswa yang dipilih secara acak dan tiga guru tim kedisiplinan sekolah. Pengujian dilakukan dengan membagikan kuesioner kepada seluruh responden sesuai dengan posisi responden apakah sebagai siswa atau guru. Kuesioner yang telah diisi oleh responden dapat dilihat pada Lampiran E dan Lampiran F. Hasil pengujian UAT yang telah dilaksanakan adalah sebagai berikut.

4.4.2.1 Hasil Pengujian UAT Guru

Jumlah responden guru pada UAT yaitu sebanyak tiga responden. Masing-masing guru yang menjadi responden memiliki peran sebagai tim kedisiplinan sekolah. Terdapat 19 pernyataan yang diajukan kepada responden guru untuk dapat diisi sesuai dengan tingkat persetujuan responden. Pernyataan-pernyataan yang diajukan kepada guru ditunjukkan pada Tabel 4.97.

Tabel 4.97 Pernyataan UAT Guru

Pernyataan ke	Pernyataan
1.	Fungsional masuk dapat dijalankan dengan baik.
2.	Fungsional reset password dapat dijalankan dengan baik.
3.	Fungsional ubah <i>password</i> dapat dijalankan dengan baik.
4.	Fungsional lihat daftar siswa dapat dijalankan dengan baik.
5.	Fungsional lihat pelanggaran siswa dapat dijalankan dengan baik.

Tabel 4.97 Pernyataan UAT Guru (lanjutan)

Pernyataan ke	Pernyataan
6.	Fungsional tambah poin pelanggaran dapat dijalankan dengan baik.
7.	Fungsional edit poin pelanggaran dapat dijalankan dengan baik.
8.	Fungsional hapus poin pelanggaran dapat dijalankan dengan baik.
9.	Fungsional cetak poin pelanggaran dapat dijalankan dengan baik.
10.	Fungsional tambah motivasi dapat dijalankan dengan baik.
11.	Fungsional tambah akun siswa dapat dijalankan dengan baik.
12.	Fungsional tambah data siswa dapat dijalankan dengan baik.
13.	Fungsional keluar dapat dijalankan dengan baik.
14.	Sistem tata tertib pelanggaran dapat membantu sekolah dalam mengolah poin pelanggaran siswa.
15.	Sistem tata tertib pelanggaran mudah dimengerti dan mudah untuk digunakan.
16.	Sistem tata tertib pelanggaran memiliki performa yang baik dalam menanggapi respon pengguna.
17.	Sistem tata tertib pelanggaran sudah sesuai dengan kebutuhan sekolah.
18.	Sistem tata tertib pelanggaran dapat dijalankan pada platform <i>mobile</i> dan desktop dengan baik.
19.	Tampilan dari sistem tata tertib pelanggaran mudah dipahami.

Masing-masing pernyataan tersebut kemudian dihitung skor ideal, skor aktual, dan persentase dari skor aktual yang telah didapat. Hasil perhitungan dari masing-masing skor tersebut didapatkan dengan menggunakan rumus yang telah dijelaskan pada Persamaan 2.4 hingga Persamaan 2.6 di Bab 2 subbab 2.6.4.2. Hasil dari perhitungan skor pengujian UAT guru ditunjukkan pada Tabel 4.98.

Tabel 4.98 Perhitungan Skor Pengujian UAT Guru

Pernyataan Ke	Skor Aktual	Skor Ideal	% Skor Aktual
1	12	12	100%
2	12	12	100%

Tabel 4.98 Perhitungan Skor Pengujian UAT Guru (lanjutan)

Pernyataan Ke	Skor Aktual	Skor Ideal	% Skor Aktual
3	12	12	100%
4	12	12	100%
5	12	12	100%
6	12	12	100%
7	12	12	100%
8	12	12	100%
9	12	12	100%
10	12	12	100%
11	12	12	100%
12	12	12	100%
13	12	12	100%
14	11	12	91,66%
15	11	12	91,66%
16	12	12	100%
17	11	12	91,66%
18	12	12	100%
19	12	12	100%

Ketika sudah didapatkan persentase skor aktual dari masing-masing pernyataan yang diajukan kepada responden, nilai persentase skor aktual yang telah didapatkan tersebut kemudian dirata-rata untuk menentukan tingkat penerimaan pengguna terhadap aplikasi yang telah dikembangkan. Perhitungan rata-rata persentase skor aktual didapatkan dengan menggunakan Persamaan 2.7.

Dari hasil perhitungan sesuai dengan Persamaan 2.7, didapatkan persentase rata-rata skor aktual sebesar 98,68%. Hasil persentase rata-rata skor aktual tersebut masuk dalam interval 75% - 100% interpretasi skala Likert pada Tabel 4.96 yang menunjukkan tingkat penerimaan responden guru pada aplikasi Sistem Tata Tertib Pelanggaran (SiTalang) termasuk dalam kategori "Sangat Setuju".

4.4.2.2 Hasil Pengujian UAT Siswa

Jumlah responden siswa pada UAT yaitu sebanyak empat responden. Siswa yang menjadi responden dipilih secara acak. Terdapat 10 pernyataan yang diajukan kepada responden siswa untuk dapat diisi sesuai dengan tingkat

persetujuan responden. Pernyataan-pernyataan yang diajukan kepada siswa ditunjukkan pada Tabel 4.99.

Tabel 4.99 Pernyataan UAT Siswa

Pernyataan ke	Pernyataan
1.	Fungsional masuk dapat dijalankan dengan baik.
2.	Fungsional reset password dapat dijalankan dengan baik.
3.	Fungsional ubah <i>password</i> dapat dijalankan dengan baik.
4.	Fungsional lihat pelanggaran dapat dijalankan dengan baik.
5.	Fungsional keluar dapat dijalankan dengan baik.
6.	Sistem tata tertib pelanggaran dapat membantu siswa untuk mengetahui info poin pelanggaran siswa.
7.	Sistem tata tertib pelanggaran mudah dimengerti dan mudah untuk digunakan.
8.	Sistem tata tertib pelanggaran memiliki performa yang baik dalam menanggapi respon pengguna.
9.	Sistem tata tertib pelanggaran dapat dijalankan pada platform <i>mobile</i> dan <i>desktop</i> dengan baik.
10.	Tampilan dari sistem tata tertib pelanggaran mudah dipahami.

Masing-masing pernyataan tersebut kemudian dihitung skor ideal, skor aktual, dan persentase dari skor aktual yang telah didapat. Perhitungan dari masing-masing skor tersebut didapatkan dengan menggunakan rumus yang telah dijelaskan pada Persamaan 2.4 hingga Persamaan 2.6 pada Bab 2 Bab 2 subbab 2.6.4.2. Hasil dari perhitungan skor pengujian UAT siswa ditunjukkan pada Tabel 4.100.

Tabel 4.100 Perhitungan Skor Pengujian UAT Siswa

Pernyataan ke	Skor Aktual	Skor Ideal	% Skor Aktual
1	16	16	100%
2	16	16	100%
3	16	16	100%
4	16	16	100%
5	16	16	100%
6	16	16	100%

Tabel 4.100 Perhitungan Skor Pengujian UAT Siswa (lanjutan)

Pernyataan Ke	Skor Aktual	Skor Ideal	% Skor Aktual
1	16	16	100%
2	16	16	100%
3	16	16	100%
4	16	16	100%
5	16	16	100%
6	16	16	100%
7	16	16	100%
8	16	16	100%
9	16	16	100%
10	16	16	100%

Ketika sudah didapatkan persentase skor aktual dari masing-masing pernyataan yang diajukan kepada responden, nilai persentase skor aktual yang telah didapatkan tersebut kemudian dirata-rata untuk menentukan tingkat penerimaan pengguna terhadap aplikasi yang telah dikembangkan. Perhitungan rata-rata didapatkan dengan menggunakan Persamaan 2.7.

Dari hasil perhitungan sesuai dengan Persamaan 2.7, didapatkan persentase rata-rata skor aktual sebesar 100%. Hasil persentase rata-rata skor aktual tersebut masuk dalam interval 75% - 100% interpretasi skala Likert pada Tabel 4.96 yang menunjukkan tingkat penerimaan responden siswa pada aplikasi Sistem Tata Tertib Pelanggaran (SiTalang) termasuk dalam kategori "Sangat Setuju".

4.4.3 Analisis Hasil Pengujian Sistem

Berdasarkan hasil pengujian validasi (*validation testing*) dan *user acceptance testing* (UAT), dapat disimpulkan bahwa aplikasi tata tertib pelanggaran dapat berjalan sesuai dengan fungsional yang telah didefinisikan dan dapat diterima oleh pihak sekolah serta siswa sebagai pengguna. Hal tersebut ditunjukkan dengan hasil dari *validation testing* yang dijalankan oleh pengguna sebagai responden pengujian menunjukkan seluruh fungsional telah sukses berjalan sesuai dengan fungsinya. Hasil dari *validation testing* dapat dilihat pada Lampiran C dan Lampiran D.

Pengujian *user acceptance testing* yang telah dilaksanakan menghasilkan penerimaan pengguna terhadap sistem dengan persentase sebesar 98,68% oleh guru dan 100% oleh siswa. Hasil perolehan nilai tersebut didapatkan melalui proses pengujian yang dilakukan dengan responden mengisi kuesioner berisi pernyataan-pernyataan yang telah disediakan. Perhitungan skala Likert seperti

skor aktual, skor ideal, dan persentase skor aktual ditunjukkan pada persamaan yang dijelaskan pada Bab 2 subbab 2.6.4.2. Oleh karena itu jika terdapat hasil pada perolehan nilai persentase skor aktual yang sempurna atau 100%, hal tersebut dikarenakan responden mengisikan skor pada kuesioner yang diberikan dengan nilai maksimal atau sangat setuju sesuai dengan interpretasi skala Likert yang telah dijelaskan.

Pengujian terhadap non-fungsional sistem yaitu *portability* juga dilakukan pada saat pengujian *user acceptance testing* dilakukan. Untuk menguji non-fungsional tersebut, peneliti melakukan pengujian yang dilakukan oleh responden melalui dua perangkat yaitu *desktop* dan *mobile*. Pernyataan terkait pengaksesan sistem pada dua perangkat yang berbeda yaitu *desktop* dan *mobile*, diperoleh hasil sebesar 100% yang berarti aplikasi dapat dijalankan pada perangkat *desktop* maupun *mobile* dengan baik.



BAB 5 PENUTUP

5.1 Kesimpulan

Dari proses penelitian yang telah dilakukan, dapat diperoleh kesimpulan sebagai berikut:

1. Dengan permasalahan kurang efektifnya pengolahan poin pelanggaran di SMK Negeri 1 Pasuruan, peneliti mengembangkan aplikasi tata tertib pelanggaran dengan fungsional yang dapat menghitung, menambah, menghapus, mengedit serta mencetak poin pelanggaran siswa yang sesuai dengan mekanisme yang telah disetujui oleh sekolah agar dapat membantu memudahkan dalam pengolahan maupun pelaporan poin pelanggaran di SMK Negeri 1 Pasuruan. Aplikasi yang dikembangkan juga berbasis *website* sehingga dapat memudahkan dalam pengaksesan aplikasi yang dapat diakses melalui *mobile* maupun desktop.
2. Berdasarkan hasil pengujian menggunakan metode *validation testing* dan *user acceptance testing*, aplikasi tata tertib pelanggaran dapat berjalan sesuai dengan fungsional yang telah didefinisikan dan dapat diterima oleh pihak sekolah serta siswa sebagai pengguna. Hal tersebut ditunjukkan dengan hasil pengujian validasi (*validation testing*) yang telah dilakukan menunjukkan seluruh fungsional telah sukses berjalan sesuai dengan fungsinya serta memiliki nilai rata-rata penerimaan sebesar 98,68% yang didapat dari 3 responden guru dan 100% yang didapat dari 4 responden siswa dengan memberikan nilai pada pernyataan-pernyataan yang telah diberikan.

5.2 Saran

Setelah melaksanakan penelitian terhadap pengolahan pelanggaran siswa di sekolah, peneliti dapat memberikan saran untuk penelitian selanjutnya yang serupa yaitu sebagai berikut:

1. Dikarenakan pada penelitian ini hanya terfokus pada pengolahan pelanggaran siswa, diharapkan untuk penelitian selanjutnya dapat menambahkan fungsional untuk mengolah data dan akun siswa seperti ketika siswa naik kelas, menghapus, serta mengedit data akun siswa.
2. Menambahkan fungsional pada aplikasi pengolah pelanggaran siswa agar dapat melakukan cetak poin pelanggaran per kelas, pemberian grafik data pelanggaran siswa per kelas, serta melakukan improvisasi pada bagian tampilan sistem agar tampilan aplikasi menjadi lebih baik.
3. Melakukan pengembangan agar sistem dapat mengolah rekapitulasi prestasi siswa di sekolah.
4. Menambahkan aktor orang tua siswa agar dapat menggunakan sistem yang telah dikembangkan untuk dapat mengawasi perilaku siswa di sekolah.

DAFTAR REFERENSI

Aprianti, W. & Maliha, U., 2016. Sistem Informasi Kepadatan Penduduk Kelurahan Atau Desa Studi Kasus Pada Kecamatan Bati-Bati. [e-journal] 2(2013), pp. 21–28. Tersedia melalui: [Jurnal Sains dan Informatika <jsi.politala.ac.id>](http://jurnal.sainsdaninformatika.jsi.politala.ac.id) [Diakses 12 September 2020].

Ardiyanto, D. & Purwoto, B. H., 2014. KOMPRESI CITRA DENGAN MENGGUNAKAN METODE DELTA MODULATION. [e-journal]. Tersedia melalui: [Jurnal Teknik Elektro <journals.ums.ac.id>](http://jurnal.teknikelektro.ums.ac.id) [Diakses 12 Juli 2021].

Bella, M. R. M., Data, M. & Yahya, W., 2019. Implementasi Load Balancing Server Web Berbasis Docker Swarm Berdasarkan Penggunaan Sumber Daya Memory Host. [e-journal], Volume 3, pp. 3478-3487. Tersedia melalui: [Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer <j-ptiik.ub.ac.id>](http://jurnal.pengembangan.teknologi.informasi.dan.ilmu.komputer.ptiik.ub.ac.id) [Diakses 12 Juli 2021].

Erinton, R., Negara, R.M. & Sanjoyo, D.D., 2017. ANALISIS PERFORMASI FRAMEWORK CODEIGNITER DAN LARAVEL MENGGUNAKAN WEB SERVER APACHE (PERFORMANCE ANALYSIS FRAMEWORK CODEIGNITER AND LARAVEL USING APACHE WEB SERVER). [e-journal]. Tersedia melalui: [e-Proceeding of Engineering <openlibrarypublicatio.telkomuniversity.ac.id>](http://openlibrarypublicatio.telkomuniversity.ac.id) [Diakses 3 Maret 2020].

Firman, A., Wowor, H.F. & Najoan, X., 2016. Sistem Informasi Perpustakaan Online Berbasis Web. [e-journal]. Tersedia melalui: [E-Journal Teknik Elektro dan Komputer <ejournal.unsrat.ac.id>](http://ejournal.unsrat.ac.id) [Diakses 6 September 2020].

Habisal, S., Sinaga, A. & Saputra, A., 2018. SISTEM PENGOLAHAN DATA PERILAKU SISWA DENGAN PENERAPAN KUM DI SMAN 2 DUMAI BERBASIS WEB. [e-journal]. Tersedia melalui: [Jurnal Infomatika, Manajemen, dan Komputer <ejournal.stmikdumai.ac.id>](http://ejournal.stmikdumai.ac.id) [Diakses 11 September 2020].

Hambling, B. & van Goethem, P., 2014. User acceptance testing requires comprehensive, strategic training. [e-book]. *Computer Weekly*. Tersedia melalui: [Library Genesis <libgen.is>](http://libgen.is) [Diakses 6 November 2020].

Hendini, A., 2016. PEMODELAN UML SISTEM INFORMASI MONITORINGPENJUALAN DAN STOK BARANG (STUDI KASUS: DISTRO ZHEZHA PONTIANAK). [e-book]. Tersedia melalui: [JURNAL KHATULISTIWA INFORMATIKA <ejournal.bsi.ac.id>](http://ejournal.bsi.ac.id) [Diakses 1 Juni 2021].

Idris, M., Rahman, A.F. & Arsyad, O.A., 2018. Perancangan Sistem Print Online. [e-journal]. Tersedia melalui: [Seminar Nasional Aplikasi Teknologi Informasi \(SNATI\) <journal.uui.ac.id>](http://journal.uui.ac.id) [Diakses 16 April 2021].

Junus, I.S., Santoso, H.B., Isal, R.Y.K. & Utomo, A.Y., 2015. Usability evaluation of the student centered e-Learning environment. [e-journal]. Tersedia melalui: [International Review of Research in Open and Distance Learning <erudit.org>](http://erudit.org) [Diakses 24 Maret 2021].

Kurniawan, T. A., 2018. PEMODELAN USE CASE (UML): EVALUASI TERHADAP BEBERAPA. [e-journal], Volume 5, pp. 77-86. Tersedia melalui: *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)* <jtiik.ub.ac.id> [Diakses 25 Maret 2021].

Noor, A., 2018. Lab IT Security System Dengan QR Code Berbasis Web. [e-journal], Volume 4. Tersedia melalui: *Jurnal Sains dan Informatika* <jsi.politala.ac.id> [Diakses 25 Mei 2021].

Novita, R. & Sari, N., 2015. Sistem Informasi Penjualan Pupuk Berbasis E-Commerce. [e-journal]. Tersedia melalui: *Teknoif* <ejournal.itp.ac.id> [Diakses 13 September 2020].

Pasuruan, S. N. 1., 2017. *SMK Negeri 1 Pasuruan*. [online] Tersedia di: smkn1-pasuruan.sch.id [Diakses 07 2021].

Pranatawijaya, V.H., Widiatry, Priskila, R. & Putra, P.B.A.A, 2019. Pengembangan Aplikasi Kuesioner Survey Berbasis Web Menggunakan Skala Likert dan Guttman. [e-journal]. Tersedia melalui: *Jurnal Sains dan Informatika* <jsi.politala.ac.id> [Diakses 19 April 2021].

Pressman, S.P. & Maxim, B.R., 2015. *SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH, EIGHTH EDITION* Published. [e-book]. McGraw-Hill Education. Tersedia melalui: Library Genesis <libgen.is> [Diakses 5 Februari 2021].

Rahmad, B. & Purnama, B.E., 2013. Rancangan Pembangunan Web E-Library Pada Perpustakaan Aptikom Indonesia Berbasis Web. [e-journal]. Tersedia melalui: *Seminar Riset Unggulan Nasional Informatika dan Komputer FTI UNSA* <ijns.org> [Diakses 13 September 2020].

Rahmayu, M., 2016. Rancang Bangun Sistem Informasi Pada Rumah Sakit Dengan Layanan Intranet Menggunakan Metode Waterfall. [e-journal]. Tersedia melalui: *Jurnal Evolusi* <ejournal.bsi.ac.id> [Diakses 13 September 2020].

Rusbandi, M. D. A., Aknuranda, I. & Pramono, D., 2019. Pengembangan Sistem Informasi Penyambungan Baru Listrik Khusus. [e-journal] Volume 3, pp. 2286-2294. Tersedia melalui: *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* <jtiik.ub.ac.id> [Diakses 25 Mei 2021].

Sarjono, S.P., Wicaksono, S. A. & Pradana, F., 2018. Pengembangan Sistem Informasi Toko Bangunan. [e-journal], pp. 3445-3453. Tersedia melalui: *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* <jptiik.ub.ac.id> [Diakses 8 Oktober 2020].

Sommerville, I., 2016. *Software engineering (10th edition)*. [e-book]. Pearson Education Limited. Tersedia melalui: Library Genesis <libgen.is> [Diakses 5 Februari 2021].

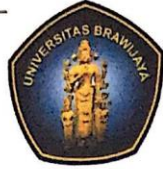
Sulistiyarini, D., Mustofa & Saputro, M., 2018. PENGEMBANGAN SISTEM INFORMASI PENJAMINAN. [e-journal]. Tersedia melalui: *Jurnal Pendidikan Informatika dan Sains* <journal.ikipgriptk.ac.id> [Diakses 21 Mei 2021].

Wahyudi, R. & Aristantia, D.A., 2017. Aplikasi Pengolahan Data Pelanggaran Siswa Pada Smk Yayasan Pendidikan Teknologi 1 Purbalingga Terintegrasi Dengan Sms Gateway. [e-journal]. Tersedia melalui: *Research Gate* <researchgate.net> [Diakses 30 April 2020].

Wardhono, W. S., Marji & Kusuma, L. P., 2015. EVALUASI USER ACCEPTANCE AUGMENTED REALITY TRIAGE MOBILE PADA SISTEM KEDARURATAN MEDIS. [e-journal]. Tersedia melalui: *Seminar Teknologi dan Rekayasa* <research-report.umm.ac.id> [Diakses 19 April 2021].



LAMPIRAN A HASIL PENGALIAN KEBUTUHAN DENGAN PIHAK SEKOLAH



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

FORM WAWANCARA KEBUTUHAN UNTUK PIHAK SEKOLAH

Aspek	Pertanyaan
Non-Behavioral	1. Bagaimana sistem pelanggaran/tata tertib yang dilakukan oleh SMKN 1 Pasuruan?
	<p>Sistem penanganan pelanggaran ! MOPBID diberi sosialisasi tatib SMKN 1</p> <p>2. Pelanggaran ringan (terlambat, tidak menggunakan atribut sekolah) diingatkan dan diberi motivasi</p> <p>3. Pelanggaran ringan yang dilakukan lebih dari 3 kali atau pelanggaran sedang, maka dilakukan pemanggilan orang tua</p> <p>4. Pelanggaran berat diadakan konferensi kasus.</p>
Non-Behavioral	2. Siapa sajakah pihak yang terlibat dalam penanganan siswa yang melanggar tata tertib di SMKN 1 Pasuruan?
	<p>1. Kepala Sekolah</p> <p>2. Wakil kepala sekolah</p> <p>3. Karomli</p> <p>4. Walas</p> <p>5. Guru BK</p> <p>6. Tim Kedisiplinan</p>





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Non-Behavioral	5. Jenis hukuman seperti apa yang akan diterima oleh siswa yang melanggar? Apakah hukuman disesuaikan dengan kategori-kategori yang ada?
	1. Ringan → peringatan & pengamatan 2. Sedang → pemanggilan orang tua 3. Berat → konferensi kasus (dipondokkan)
Non-Behavioral	6. Apakah siswa dapat mengajukan keringanan atau penghapusan poin? Bila ada, bagaimana alur dari proses keringanan atau penghapusan poin tersebut?
	Belum pernah ada keringanan / penghapusan point pelanggaran



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Non-Behavioral	<p>3. Bisa dijelaskan mengenai alur pemberian hukuman yang dilakukan oleh pihak sekolah kepada siswa yang bermasalah?</p> <p>1. Mengidentifikasi masalah yang dilakukan oleh BK</p> <p>2. Diberikan pengarahan</p> <p>3. Apabila pelanggaran lebih dari 2x pemanggilan orang tua</p> <p>4. Diadakan konferensi kasus melibatkan KS, waka kesiswaan, BK, walas dan tim kedisiplinan</p>
Non-Behavioral	<p>4. Apakah dalam pemberian sanksi terdapat kategori-kategori yang sesuai dengan pelanggaran yang dilakukan oleh siswa? Apabila ada apa saja kategori-kategori tersebut?</p> <p>1. Ringan</p> <p>2. Sedang</p> <p>3. Berat</p>



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Non-Behavioral	7. Apakah nantinya hasil rekam jejak poin yang diperoleh oleh siswa akan diberikan pada waktu pembagian rapor siswa?
	Selama ini belum direkap di rapor, untuk kedepannya diharapkan ada aplikasi untuk merekap point pelanggaran & point prestasi siswa
Behavioral	8. Hal apa saja yang diharapkan bisa dilakukan user pada aplikasi ini nantinya?
	<ul style="list-style-type: none"> - Merekap point pelanggaran - Penghitungan point pelanggaran - Memberikan laporan yang akurat pada ortu tercait pelanggaran yang dilafutan.



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Non-Behavioral	9. Siapakah yang nantinya akan menjadi stakeholder dan user dalam aplikasi?
	<ul style="list-style-type: none"> - Stakeholder → Waka Kesiswaan - User → Tim Kedisiplinan
Behavioral	10. Jika nantinya terdapat sebuah aplikasi untuk mengelola poin pelanggaran siswa, fitur apa saja yang diharapkan oleh SMKN 1 Pasuruan? Contoh : Login, Tambah Poin, Cetak Data Poin Siswa, dan sebagainya.
	Ditambah deskripsi yang berisi motivasi / penguatan.

Pasuruan, 16.10.2020

Nama Setyo Wahyu Wicaksono, M.Pd

NIP./NIK. 197406202006041007

LAMPIRAN B PANDUAN POIN PENANGANAN SISWA



PEMERINTAH PROVINSI JAWA TIMUR
DINAS PENDIDIKAN
**SEKOLAH MENENGAH KEJURUAN NEGERI 1
PASURUAN**
Jalan Veteran 11 Pasuruan, Telp/Fax (0343) 421380
Website : www.smkn1-pasuruan.sch.id, Email : smk1pasuruan@yahoo.com
PASURUAN KodePos 67122

ATURAN PENANGANAN SISWA

- A. RINGAN (KERAPIAN) (poin5)
1. Atribut sekolah tidak lengkap meliputi : bed, topi, dasi dan kaos kaki
 - a. Siswa dicatat dalam buku pelanggaran siswa
 - b. Siswa di suruh melengkapi dengan membeli di koperasi
 2. Seragam tidak sesuai dengan aturan yang ditentukan sekolah
 - a. Siswa dicatat dalam buku pelanggaran Siswa.
 - b. Siswa di suruh pulang untuk ganti seragam dan datang kesekolah kembali.
 3. Sepatu tidak sesuai dengan ketentuan
 - a. Siswa dicatat dalam buku pelanggaran siswa
 - b. Sepatu disita dan dicat oleh petugas
 4. Rambut panjang, diwarnai, model tidak standart
 - a. Siswa dicatat dalam buku pelanggaran siswa
 - b. dipotong atau disuruh memotong
 5. Kuku panjang, asesoris berlebihan
 - a. Asesoris disita
 - b. Kuku di potong atau disuruh memotong
 6. Siswa perempuan menggunakan make up berlebihan
 - a. Siswa diperingatkan dan di suruh membersihkan
- B. SEDANG (KEDISIPLINAN)
1. Alpha (tidak masuk tanpa keterangan 1 kali poin 10)
 - a. Siswa Alpha dicatat dalam buku pelanggaran siswa
 - b. Siswa alpha sampai 3 kali tatap muka, siswa dipanggil untuk klarifikasi, dicatat dalam buku pelanggaran. Dan dilakukan pembinaan
 - c. Membuat surat pernyataan untuk tidak mengulangi lagi.
 - d. Siswa Alpha 4 kali, dilakukan pemanggilan orang tua, jika pemanggilan orang tua sampai 2 kali tidak hadir, maka dilakukan home visit.
 - e. Dilakukan pembinaan, surat keterangan untuk tidak mengulangi lagi
 - f. Siswa mengulangi tidak masuk alpha sampai 5 kali atau lebih dan telah dilakukan proses pemanggilan atau pembinaan maka siswa dipanggil bersama orang tua, diajukan kedalam konferensi untuk pertimbangan kebijakan (dikembalikan ke orang tua atau masih perlu dibina).



2. Bolos/tidak mengikuti pelajaran pada saat jam pelajaran (poin 10)
 - a. Siswa Bolos dua kali di panggil dan dilakukan klarifikasi dan dicatat dalam buku pelanggaran.
 - b. Siswa mendapat pembinaan.
 - c. Siswa bolos 3 kali dipanggil dan diidentifikasi, dicatat dalam buku pelanggaran siswa.
 - d. Siswa mendapat pembinaan dan membuat surat pernyataan untuk tidak mengulangi lagi.
 - e. Siswa bolos lebih dari 3 kali panggilan orang tua untuk klarifikasi tentang keadaan siswanya.
 - f. Disamping langkah poin (diatas)
 3. Siswa terlambat
 - a. Siswa terlambat di catat di buku pelanggaran siswa
 - b. Siswa terlambat 5 kali dipanggil diidentifikasi, dan di catat dalam buku pelanggaran siswa
 - c. dilakukan bimbingan.
 - d. Siswa terlambat 10 kali, dilakukan panggilan orang tua
 - e. Dilakukan pembinaan dan membuat surat pernyataan
 - f. Terlambat sampai 20 kali lebih maka siswa diajukan ke konferensi
 - g. Siswa mendapat pembinaan, dilakukan panggilan orang tua
 - h. Mendapat sanksi bidang kebersihan.
 4. Membuat surat keterangan palsu (poin 10)
 - a. Siswa dipanggil diklarifikasi, dicatat dalam buku pelanggaran
 - b. Surat ijin tidak berlaku
 - c. Jika lebih dari 1 kali maka dilakukan pemanggilan orang tua
 - d. Dilakukan pembinaan, membuat surat pernyataan untuk tidak mengulangi lagi
- C. BERAT (KELAKUAN)
1. Obat terlarang/Narkoba (100 poin)
 - a. Siswa membawa, menggunakan, mengedarkan dan atau mempengaruhi teman atau orang lain
 - b. Siswa dipanggil, dicatat dalam buku pelanggaran, diidentifikasi dan panggilan orang tua
 - c. Diajukan pada konferensi untuk dikembalikan pada orang tua.
 2. Minuman keras(miras atau sejenisnya) poin 75
 - a. Siswa melakukan minum minuman keras, membawa, mendukung dalam lingkungan sekolah (menggunakan atribut sekolah).



3. Mencuri

- a. Siswa melakukan pencurian dan atau terlibat mengarah kejahatan kriminal baik lingkungan sekolah maupun diluar (point 100), siswa dicatat dalam buku pelanggaran siswa dan di kembalikan pada orang tua
- b. siswa melakukan pencurian terlibat dalam kejadian yang mengarah pada kenakalan (75 poin)
 - ✓ siswa dan orang tua di panggil diidentifikasi di catat dalam buku pelanggaran siswa
 - ✓ siswa diberi pembinaan dan membuat surat pernyataan untuk tidak mengulangi
 - ✓ diberi sanksi skorsing 3 hari
 - ✓ jika siswa mengulangi melakukan pencurian kembali lebih dari satu kali maka siswa di kembalikan pada orang tua

4. Tindakan asusila, pencemaran nama baik, hamil diluar nikah (100 poin)

- a. Siswa di panggil dan di catat dalam buku pelanggaran siswa, diidentifikasi dan diajukan pada konferensi
- b. Panggilan orang tua
- c. Dikembalikan pada orang tua

5. Perkelahian di lingkungan sekolah (50 Poin)

- a. Siswa dipanggil dan diidentifikasi di catat dalam buku pelanggaran siswa
- b. Panggilan orang tua
- c. Mendapat pembinaan dan membuat surat pernyataan untuk tidak mengulangi
- d. Dan jika siswa melakukan kembali perkelahian maka diberi sanksi skorsing 3 hari disamping langkah (a)

6. Membawa barang – barang terlarang (senjata tajam untuk kejahatan, CD pornografi) (50 Poin)

- a. Siswa dipanggil dan diidentifikasi , dicatat dalam buku pelanggaran siswa
- b. Panggilan orang tua dan membuat surat pernyataan untuk tidak mengulangi lagi
- c. Mendapat bimbingan
- d. Barang – barang disita oleh sekolah

7. Membawa rokok dilingkungan sekolah (50 Poin)

- a. Siswa membawa atau merokok dipanggil dan diidentifikasi, di catat dalam buku pelanggaran
- b. Panggilan orang tua dan membuat surat pernyataan untuk tidak mengulangi lagi
- c. Mendapat pembinaan
- d. Jika melakukan sampai tiga kali maka siswa dikenakan skorsing bidang kebersihan

8. Merusak fasilitas sekolah (sarana/prasarana dan atau mengotori, mencorat – coret atau tulisan tidak sopan) (poin 25)

- a. Siswa dipanggil untuk diidentifikasi, siswa dicatat dalam buku pelanggaran siswa
- b. Mendapat pembinaan dan membuat surat pernyataan untuk tidak mengulangi lagi

- c. Diberi sanksi untuk mengganti barang yang rusak, membersihkan barang yang dikotori
 - d. Jika tulisan mengarah pada pelecehan pada pihak lain maka pemanggilan orang tua dan diadakan klarifikasi dengan pihak yang bersangkutan
9. Menghabiskan uang bukan hak miliknya (poin 25)
- a. Siswa dipanggil dan dicatat dalam buku pelanggaran siswa dan diidentifikasi
 - b. Siswa dipanggil orang tuanya, memuat surat pernyataan untuk tidak mengulangi
 - c. Mengembalikan uang yang di habiskan
 - d. Mendapat pembinaan
10. Mengganggu pihak lain (poin 20)
- a. Siswa melakukan pemalakan/penargetan pada pihak lain baik dilingkungan sekolah maupun diluar sekolah
 - b. Siswa dipanggil diidentifikasi dan dicatat dalam buku pelanggaran siswa
 - c. Panggilan orang tua membuat surat pernyataan untuk tidak mengulangi.
 - d. Dilakukan pembinaan dan skorsing 3 hari
 - e. Pengembalian hasil penargetan

KRITERIA PENILAIAN

NO	POIN	NILAI	KETERANGAN
1	0	A	Amat baik
2	1 – 25	B	Baik
3	26 – 50	C	Cukup
4	51 – 75	D	Kurang
5	76 – 100	E	Amat Kurang

LAMPIRAN C HASIL PENGUJIAN VALIDASI OLEH GURU



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

PENGUJIAN VALIDASI APLIKASI TATA TERTIB PELANGGARAN KUESIONER UNTUK GURU

Pengujian validasi ini bertujuan untuk menguji fungsional/menu dari aplikasi tata tertib pelanggaran. Pada pengujian ini proses berhasil dan gagal akan diujikan. Berikut langkah-langkah yang harus diperhatikan selama proses pengujian dilakukan:

1. Baca *test conditions* terlebih dahulu pada setiap pengujian yang akan dilakukan. *Test conditions* memberikan informasi terkait tujuan dari pengujian yang dilakukan pada setiap fungsional yang diujikan.
2. Pada setiap *test conditions* terdapat tabel yang berisi informasi terkait aktor, skenario tes (langkah apa saja yang akan dilakukan), hasil yang diharapkan, data tes (jika ada), dan langkah-langkah melakukan pengujian.
3. Informasi data tes digunakan untuk mengisikan edit teks yang terdapat pada setiap fungsional yang diujikan.
4. Baca *test description* (langkah yang dilakukan) dan *expected result* (hasil yang diharapkan) yang terdapat pada setiap tabel *test conditions*.
5. Berikan simbol centang (✓) jika sukses dan (x) jika gagal pada setiap kolom berdasarkan *test description* dan *expected result*.



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Masuk

- *Test Conditions:*
 1. Tidak dapat melakukan masuk jika *username* atau *password* guru salah.
 2. Dapat melakukan masuk jika *username* dan *password* guru benar.

Test Condition 1 – Tidak dapat melakukan masuk jika *username* dan *password* guru salah.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman <i>browser</i>. <i>Actor:</i> Pengguna. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> • Mengakses alamat URL SiTalang (tatatertibpelanggaran.000webhostapp.com). • Memasukkan <i>username</i> atau <i>password</i> dengan data yang salah. • Menekan tombol masuk. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal masuk ke halaman utama aplikasi dan tetap berada di halaman masuk. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> • <i>Username:</i> admin1 • <i>Password:</i> admin111 			
1.	Aktor mengakses alamat URL SiTalang.	Sistem menampilkan halaman masuk aplikasi. Pada halaman masuk terdapat edit teks kosong <i>username</i> , edit teks kosong <i>password</i> , link reset <i>password</i> , tombol masuk, dan gambar.	✓
2.	Aktor memasukkan admin1 pada edit teks <i>username</i>	Masukan aktor tampil pada edit teks <i>username</i> .	✓
3.	Aktor memasukkan admin111 pada edit teks <i>password</i>	Masukan aktor tampil pada edit teks <i>password</i> dengan karakter *	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

4.	Aktor menekan tombol masuk.	Sistem menampilkan peringatan bahwa <i>username</i> atau <i>password</i> salah.	✓
----	-----------------------------	---	---

Test Condition 2 - Dapat melakukan masuk jika *username* dan *password* guru benar.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman <i>browser</i>. Actor: Pengguna. Scenario Test:</p> <ul style="list-style-type: none"> Mengakses alamat URL SiTalang (tatatertibpelanggaran.000webhostapp.com). Memasukkan <i>username</i> dan <i>password</i> dengan data yang benar. Menekan tombol masuk. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor berhasil masuk ke halaman utama aplikasi. <p>Test Data:</p> <ul style="list-style-type: none"> <i>Username:</i> admin1 <i>Password:</i> admin123 			
1.	Aktor mengakses alamat URL SiTalang.	Sistem menampilkan halaman masuk aplikasi. Pada halaman masuk terdapat edit teks kosong <i>username</i> , edit teks kosong <i>password</i> , link reset <i>password</i> , tombol masuk, dan gambar.	✓
2.	Aktor memasukkan admin1 pada edit teks <i>username</i>	Masukan aktor tampil pada edit teks <i>username</i> .	✓
3.	Aktor memasukkan admin123 pada edit teks <i>password</i>	Masukan aktor tampil pada edit teks <i>password</i> dengan karakter *.	✓
4.	Aktor menekan tombol masuk.	Sistem menampilkan halaman utama guru.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Lihat Daftar Siswa

- **Test Conditions:**
 1. Dapat melihat daftar siswa.

Test Condition 1 – Dapat melihat daftar siswa.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman utama aplikasi. <i>Actor:</i> Guru. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> • Melakukan masuk. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor berhasil melihat daftar siswa. <p><i>Test Data:</i> -</p>			
1.	Aktor telah melakukan masuk.	Sistem menampilkan halaman daftar siswa. Pada halaman daftar siswa terdapat tabel daftar siswa, <i>dropdown sorting</i> kelas, <i>dropdown sorting</i> jurusan, dan <i>pagination</i> .	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Keluar

- *Test Conditions:*
 1. Dapat keluar dari sistem.

Test Condition 1 – Dapat keluar dari sistem.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru/Siswa berada di halaman utama aplikasi. <i>Actor:</i> Guru, Siswa. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> • Menekan tombol keluar pada bagian <i>navigation bar</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor berhasil keluar dari sistem. <p><i>Test Data:</i> -</p>			
1.	Aktor menekan tombol keluar pada bagian <i>navigation bar</i> .	Sistem menampilkan halaman masuk.	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Reset Password

- **Test Conditions:**
 1. Tidak dapat mengirimkan kode jika *email* pengguna salah.
 2. Dapat mengirimkan kode jika *email* pengguna benar.
 3. Tidak dapat mengakses halaman reset *password* jika kode konfirmasi salah.
 4. Dapat mengakses halaman reset *password* jika kode konfirmasi benar.
 5. Tidak dapat mengatur ulang informasi *password* jika masukan tidak valid.
 6. Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.
 7. Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru kurang dari 6 karakter.
 8. Dapat mengatur ulang informasi *password* jika masukan valid.

Test Condition 1 – Tidak dapat mengirimkan kode jika *email* pengguna salah.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman masuk. Actor: Pengguna. Scenario Test:</p> <ul style="list-style-type: none"> • Menekan <i>link</i> bertuliskan Reset Password! • Memasukkan <i>email</i> dengan data yang salah. • Menekan tombol kirim kode. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal mengakses halaman konfirmasi kode dan tidak menerima kode konfirmasi melalui <i>email</i>. <p>Test Data:</p> <ul style="list-style-type: none"> • <i>Email:</i> email.salah@gmail.com 			
1.	Pengguna menekan <i>link</i> bertuliskan Reset Password!	Sistem menampilkan halaman verifikasi <i>email</i> . Pada halaman verifikasi <i>email</i> terdapat edit teks kosong <i>email</i> , tombol bertuliskan kirim kode,	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

		dan <i>link</i> bertuliskan kembali ke halaman masuk.	
2.	Pengguna memasukkan email.salah@gmail.com pada edit teks <i>email</i> .	Masukan aktor tampil pada edit teks <i>email</i> .	✓
3.	Pengguna menekan tombol kirim kode.	Sistem menampilkan peringatan bahwa <i>email</i> salah.	✓

Test Condition 2 - Dapat mengirimkan kode jika *email* pengguna benar.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman masuk. Actor: Pengguna. Scenario Test:</p> <ul style="list-style-type: none"> Menekan <i>link</i> bertuliskan Reset Password!. Memasukkan <i>email</i> dengan data yang benar. Menekan tombol kirim kode. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor berhasil mengakses halaman konfirmasi kode dan menerima kode konfirmasi melalui <i>email</i>. <p>Test Data:</p> <ul style="list-style-type: none"> <i>Email:</i> emailnya.sodiq@gmail.com 			
1.	Pengguna menekan <i>link</i> bertuliskan Reset Password!	Sistem menampilkan halaman verifikasi <i>email</i> . Pada halaman verifikasi <i>email</i> terdapat edit teks kosong <i>email</i> , tombol bertuliskan kirim kode, dan <i>link</i> bertuliskan kembali ke halaman masuk.	✓
2.	Pengguna memasukkan emailnya.sodiq@gmail.com pada edit teks <i>email</i> .	Masukan aktor tampil pada edit teks <i>email</i> .	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

3.	Pengguna menekan tombol kirim kode.	Sistem menampilkan halaman konfirmasi kode dan mengirimkan kode konfirmasi melalui <i>email</i> diikuti dengan pesan pemberitahuan untuk mengecek <i>email</i> aktor untuk melihat kode konfirmasi.	✓
----	-------------------------------------	---	---

Test Condition 3 – Tidak dapat mengakses halaman reset *password* jika kode konfirmasi salah.

Test Step	Test Description	Expected Result	Sukses/Gagal
Pre-Condition: Pengguna berada di halaman konfirmasi kode. Actor: Pengguna. Scenario Test: <ul style="list-style-type: none"> Memasukkan kode konfirmasi dengan data yang salah. Menekan tombol <i>submit</i>. Expected Result: <ul style="list-style-type: none"> Aktor gagal mengakses halaman reset <i>password</i>. Test Data: <ul style="list-style-type: none"> Kode: Tidak sesuai dengan kode yang diterima melalui <i>email</i> (<i>random</i>). 			
1.	Pengguna memasukkan kode yang tidak sesuai dengan kode yang diterima melalui <i>email</i> pada edit teks kode.	Masukan aktor tampil pada edit teks kode.	✓
2.	Pengguna menekan tombol <i>submit</i> .	Sistem menampilkan peringatan bahwa kode tidak sesuai.	✓

Test Condition 4 – Dapat mengakses halaman reset *password* jika kode konfirmasi benar.

Test Step	Test Description	Expected Result	Sukses/Gagal
-----------	------------------	-----------------	--------------



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

<p><i>Pre-Condition:</i> Pengguna berada di halaman konfirmasi kode. <i>Actor:</i> Pengguna. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Memasukkan kode konfirmasi dengan data yang benar. Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor berhasil mengakses halaman reset <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> Kode: Sesuai dengan kode yang diterima melalui <i>email</i>. 			
1.	Pengguna memasukkan kode yang diterima pada edit teks kode.	Masukan aktor tampil pada edit teks kode.	✓
2.	Pengguna menekan tombol <i>submit</i> .	Sistem menampilkan halaman reset <i>password</i> . Pada halaman reset <i>password</i> terdapat edit teks kosong <i>password</i> baru, <i>konfirmasi password</i> , tombol mata, tombol <i>submit</i> , dan <i>link</i> kembali ke halaman masuk diikuti dengan pesan apabila kode sesuai.	✓

Test Condition 5 – Tidak dapat mengatur ulang informasi *password* jika masukan tidak valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
-----------	------------------	-----------------	--------------





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

<p><i>Pre-Condition:</i> Pengguna berada di halaman reset <i>password</i>. <i>Actor:</i> Pengguna. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang tidak valid. Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal mengatur ulang informasi <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> <i>Password</i> baru: passwordbaru Konfirmasi <i>Password</i>: passwordbaru 			
1.	Pengguna memasukkan passwordbaru pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
2.	Pengguna memasukkan passwordbaru pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓
3.	Pengguna menekan tombol <i>submit</i> .	Sistem menampilkan peringatan bahwa masukan aktor tidak valid.	✓

Test Condition 6 – Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman reset <i>password</i>. <i>Actor:</i> Pengguna. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang berbeda. Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal mengatur ulang informasi <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> <i>Password</i> baru: passwordbaru123 Konfirmasi <i>Password</i>: passwordbaru456 			





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

1.	Pengguna memasukkan passwordbaru123 pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
2.	Pengguna memasukkan passwordbaru456 pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓
3.	Pengguna menekan tombol <i>submit</i> .	Sistem menampilkan peringatan bahwa masukan <i>password</i> baru dan konfirmasi <i>password</i> harus sama.	✓

Test Condition 7 – Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru kurang dari 6 karakter.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman reset <i>password</i>. <i>Actor:</i> Pengguna. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan kurang dari 6 karakter. Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal mengubah informasi <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> <i>Password</i> baru: tes12 Konfirmasi <i>Password</i>: tes12 			
1.	Aktor memasukkan tes12 pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
2.	Aktor memasukkan tes12 pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓
3.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan peringatan bahwa	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

		masukan <i>password</i> baru harus mempunyai minimal 6 karakter.	
--	--	--	--

Test Condition 8 – Dapat mengatur ulang informasi *password* jika masukan valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman reset <i>password</i>. Actor: Pengguna. Scenario Test:</p> <ul style="list-style-type: none"> Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang valid. Menekan tombol <i>submit</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor berhasil mengatur ulang informasi <i>password</i>. <p>Test Data:</p> <ul style="list-style-type: none"> <i>Password</i> baru: passwordbaru123 Konfirmasi <i>Password</i>: passwordbaru123 			
1.	Pengguna memasukkan passwordbaru123 pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
2.	Pengguna memasukkan passwordbaru123 pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓
3.	Pengguna menekan tombol <i>submit</i> .	Sistem menampilkan halaman masuk diikuti dengan pesan bahwa informasi <i>password</i> telah diperbarui.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Ubah Password

- **Test Conditions:**
 1. Tidak dapat mengubah informasi *password* jika masukan tidak valid.
 2. Tidak dapat mengubah informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.
 3. Tidak dapat mengubah informasi *password* jika masukan *password* baru kurang dari 6 karakter.
 4. Dapat mengubah informasi *password* jika masukan valid.

Test Condition 1 – Tidak dapat mengubah informasi *password* baru jika masukan tidak valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru/Siswa berada di halaman utama aplikasi. Actor: Guru, Siswa. Scenario Test:</p> <ul style="list-style-type: none"> • Menekan tombol ubah <i>password</i> pada bagian <i>navigation bar</i>. • Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang tidak valid. • Menekan tombol <i>submit</i>. • Menekan tombol <i>submit</i> pada modal. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal mengubah informasi <i>password</i>. <p>Test Data:</p> <ul style="list-style-type: none"> • <i>Password</i> baru: ubahpassword • Konfirmasi <i>Password</i>: ubahpassword 			
1.	Aktor memasukkan ubahpassword pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
2.	Aktor memasukkan ubahpassword pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

3.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan modal konfirmasi ubah <i>password</i> .	✓
4.	Aktor menekan tombol <i>submit</i> pada bagian <i>modal</i> .	Sistem menampilkan peringatan bahwa masukan aktor tidak valid.	✓

Test Condition 2 – Tidak dapat mengubah informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru/Siswa berada di halaman utama aplikasi. Actor: Guru, Siswa. Scenario Test:</p> <ul style="list-style-type: none"> Menekan tombol ubah <i>password</i> pada bagian <i>navigation bar</i>. Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang berbeda. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada <i>modal</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor gagal mengubah informasi <i>password</i>. <p>Test Data:</p> <ul style="list-style-type: none"> <i>Password</i> baru: ubahpassword000 Konfirmasi <i>Password</i>: ubahpassword123 			
1.	Aktor menekan menu ubah <i>password</i> pada bagian <i>navigation bar</i> .	Sistem menampilkan halaman ubah <i>password</i> . Pada halaman ubah <i>password</i> terdapat edit teks kosong <i>password</i> baru, konfirmasi <i>password</i> , tombol kembali, dan tombol <i>submit</i> .	✓
2.	Aktor memasukkan ubahpassword000 pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

3.	Aktor memasukkan ubahpassword123 pada edit teks konfirmasi password.	Masukan aktor tampil pada edit teks konfirmasi password dengan karakter *.	✓
4.	Aktor menekan tombol submit.	Sistem menampilkan modal konfirmasi ubah password.	✓
5.	Aktor menekan tombol submit pada bagian modal.	Sistem menampilkan peringatan bahwa masukan password baru dan konfirmasi password harus sama.	✓

Test Condition 3 – Tidak dapat mengubah informasi password jika masukan password baru kurang dari 6 karakter.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru/Siswa berada di halaman utama aplikasi. Actor: Guru, Siswa. Scenario Test:</p> <ul style="list-style-type: none"> Menekan tombol ubah password pada bagian navigation bar. Memasukkan password baru dan konfirmasi password dengan masukan kurang dari 6 karakter. Menekan tombol submit. Menekan tombol submit pada modal. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor gagal mengubah informasi password. <p>Test Data:</p> <ul style="list-style-type: none"> Password baru: adm12 Konfirmasi Password: adm12 			
1.	Aktor menekan menu ubah password pada bagian navigation bar.	Sistem menampilkan halaman ubah password. Pada halaman ubah password terdapat edit teks kosong password baru, konfirmasi password, tombol	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

		kembali, dan tombol <i>submit</i> .	
2.	Aktor memasukkan adm12 pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
3.	Aktor memasukkan adm12 pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓
4.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan modal konfirmasi ubah <i>password</i> .	✓
5.	Aktor menekan tombol <i>submit</i> pada bagian <i>modal</i> .	Sistem menampilkan peringatan bahwa masukan <i>password</i> baru harus mempunyai minimal 6 karakter.	✓

Test Condition 4 – Dapat mengubah informasi *password* baru jika masukan valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru/Siswa berada di halaman utama aplikasi. <i>Actor:</i> Guru, Siswa. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Menekan tombol ubah <i>password</i> pada bagian <i>navigation bar</i>. Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang valid. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor berhasil mengubah informasi <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> <i>Password</i> baru: admin123 Konfirmasi <i>Password:</i> admin123 			
1.	Aktor menekan menu ubah <i>password</i> pada bagian <i>navigation bar</i> .	Sistem menampilkan halaman ubah <i>password</i> . Pada halaman ubah <i>password</i> terdapat edit	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

		teks kosong <i>password</i> baru, <i>konfirmasi password</i> , tombol kembali, dan tombol <i>submit</i> .	
2.	Aktor memasukkan admin123 pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
3.	Aktor memasukkan admin123 pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓
4.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan modal konfirmasi ubah <i>password</i> .	✓
5.	Aktor menekan tombol <i>submit</i> pada bagian <i>modal</i> .	Sistem menampilkan halaman masuk diikuti pesan bahwa informasi <i>password</i> berhasil diubah.	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Lihat Pelanggaran Siswa

• *Test Conditions:*

1. Dapat melihat rincian pelanggaran siswa sesuai dengan siswa yang dipilih.

Test Condition 1 – Dapat melihat rincian pelanggaran siswa sesuai dengan siswa yang dipilih.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman daftar siswa. <i>Actor:</i> Guru. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> • Menekan tombol detail yang berada pada baris siswa. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor berhasil melihat rincian pelanggaran siswa. <p><i>Test Data:</i> -</p>			
1.	Aktor menekan tombol detail yang berada pada baris siswa.	Sistem menampilkan halaman lihat pelanggaran siswa sesuai dengan siswa yang dipilih. Pada halaman pelanggaran siswa terdapat informasi profil siswa, <i>textarea</i> motivasi, tombol tambah poin pelanggaran, tombol cetak poin pelanggaran, dan daftar poin pelanggaran yang pernah didapatkan siswa. Pada daftar poin pelanggaran siswa tersebut terdapat tombol edit poin pelanggaran dan hapus poin pelanggaran.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Tambah Poin Pelanggaran

• **Test Conditions:**

1. Tidak dapat menambah poin pelanggaran jika ada masukan tambah poin pelanggaran yang tidak valid.
2. Tidak dapat menambah poin pelanggaran jika ada masukan tambah poin pelanggaran yang tidak diisi atau kosong.
3. Dapat menambah poin pelanggaran jika masukan tambah poin pelanggaran valid.

Test Condition 1 – Tidak dapat menambah poin pelanggaran jika ada masukan tambah poin pelanggaran yang tidak valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman lihat pelanggaran siswa. Actor: Guru. Scenario Test:</p> <ul style="list-style-type: none"> • Menekan tombol tambah poin pelanggaran pada bagian data siswa. • Memasukkan jumlah poin dengan masukan yang tidak valid. • Menekan tombol <i>submit</i>. • Menekan tombol <i>submit</i> pada modal. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal menambah poin pelanggaran siswa. <p>Test Data:</p> <ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Sepatu tidak sesuai dengan ketentuan • Kategori: Ringan • Jumlah Poin: 0 • Tanggal: 19 April 2021 • Tindak Lanjut: Diberikan pengarahan agar tidak mengulangi pelanggaran kembali. 			
1.	Aktor menekan tombol tambah poin pelanggaran yang berada pada bagian data siswa.	Sistem menampilkan halaman tambah poin pelanggaran. Pada halaman tambah poin pelanggaran terdapat edit teks kosong	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

		pelanggaran yang dilakukan, <i>dropdown</i> kategori, edit teks kosong jumlah poin, edit teks kosong tanggal, tombol kembali dan tombol <i>submit</i> .	
2.	Aktor memasukkan Tidak memakai atribut sekolah dengan lengkap pada edit teks pelanggaran yang dilakukan.	Masukan aktor tampil pada edit teks pelanggaran yang dilakukan.	✓
3.	Aktor memilih Ringan pada <i>dropdown</i> kategori.	Pilihan aktor tampil pada <i>dropdown</i> kategori.	✓
4.	Aktor memasukkan 0 pada edit teks jumlah poin.	Masukan aktor tampil pada edit teks jumlah poin.	✓
5.	Aktor memilih 19 April 2021 pada edit teks tanggal.	Masukan aktor tampil pada edit teks tanggal.	✓
6.	Aktor memasukkan teks tindak lanjut pada edit teks tindak lanjut.	Masukan aktor tampil pada edit teks jumlah tindak lanjut.	✓
7.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan <i>modal</i> untuk mengkonfirmasi penambahan poin pelanggaran.	✓
8.	Aktor menekan tombol <i>submit</i> .	Sistem tetap menampilkan halaman tambah poin pelanggaran diikuti pesan bahwa masukan jumlah poin harus lebih dari 0.	✓

Test Condition 2 – Tidak dapat menambah poin pelanggaran jika ada masukan tambah poin pelanggaran yang tidak diisi atau kosong.





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman lihat pelanggaran siswa. <i>Actor:</i> Guru. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Menekan tombol tambah poin pelanggaran pada bagian data siswa. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada bagian <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal menambah poin pelanggaran siswa. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> Pelanggaran yang Dilakukan: - Kategori: - Jumlah Poin: - Tindak Lanjut: - 			
1.	Aktor menekan tombol tambah poin pelanggaran yang berada pada bagian data siswa.	Sistem menampilkan halaman tambah poin pelanggaran. Pada halaman tambah poin pelanggaran terdapat edit teks kosong pelanggaran yang dilakukan, <i>dropdown</i> kategori, edit teks kosong jumlah poin, tombol kembali dan tombol <i>submit</i> .	✓
2.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan <i>modal</i> untuk mengkonfirmasi penambahan poin pelanggaran.	✓
3.	Aktor menekan tombol <i>submit</i> .	Sistem tetap menampilkan halaman tambah poin pelanggaran diikuti pesan seluruh informasi tambah poin pelanggaran harus diisi.	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Test Condition 3 – Dapat menambah poin pelanggaran jika masukan tambah poin pelanggaran valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman lihat pelanggaran siswa. Actor: Guru. Scenario Test:</p> <ul style="list-style-type: none"> Menekan tombol tambah poin pelanggaran pada bagian data siswa. Memasukkan pelanggaran yang dilakukan, kategori, jumlah poin, dan tanggal dengan masukan yang valid. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada bagian <i>modal</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor berhasil menambah poin pelanggaran siswa. <p>Test Data:</p> <ul style="list-style-type: none"> Pelanggaran yang Dilakukan: Tidak memakai atribut sekolah dengan lengkap Kategori: Ringan Jumlah Poin: 5 Tanggal: 20 April 2021 Tindak Lanjut: Diberikan pengarahannya agar tidak mengulangi pelanggaran kembali. 			
1.	Aktor menekan tombol tambah poin pelanggaran yang berada pada bagian data siswa.	Sistem menampilkan halaman tambah poin pelanggaran. Pada halaman tambah poin pelanggaran terdapat edit teks kosong pelanggaran yang dilakukan, <i>dropdown</i> kategori, edit teks kosong jumlah poin, edit teks kosong tanggal, tombol kembali dan tombol <i>submit</i> .	✓
2.	Aktor memasukkan Tidak memakai atribut sekolah dengan lengkap pada edit teks pelanggaran yang dilakukan.	Masukan aktor tampil pada edit teks pelanggaran yang dilakukan.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

3.	Aktor memilih Ringan pada dropdown kategori.	Pilihan aktor tampil pada dropdown kategori.	✓
4.	Aktor memasukkan 5 pada edit teks jumlah poin.	Masukan aktor tampil pada edit teks jumlah poin.	✓
5.	Aktor memilih 20 April 2021 pada edit teks tanggal.	Masukan aktor tampil pada edit teks tanggal.	✓
6.	Aktor memasukkan informasi tindak lanjut pada edit teks tindak lanjut.	Masukan aktor tampil pada edit teks jumlah tindak lanjut.	✓
7.	Aktor menekan tombol submit.	Sistem menampilkan modal untuk mengkonfirmasi penambahan poin pelanggaran.	✓
8.	Aktor menekan tombol submit.	Sistem menampilkan halaman pelanggaran siswa diikuti dengan menampilkan pesan apabila tambah poin pelanggaran telah berhasil.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Edit Poin Pelanggaran

• **Test Conditions:**

1. Tidak dapat mengubah informasi poin pelanggaran jika perubahan informasi poin pelanggaran tidak valid.
2. Tidak dapat mengubah poin pelanggaran jika perubahan informasi poin pelanggaran adalah menghilangkan nilai informasi poin pelanggaran.
3. Dapat mengubah informasi poin pelanggaran jika perubahan informasi poin pelanggaran valid.

Test Condition 1 – Tidak dapat mengubah informasi poin pelanggaran jika perubahan informasi poin pelanggaran tidak valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman lihat pelanggaran siswa. Actor: Guru. Scenario Test:</p> <ul style="list-style-type: none"> • Menekan tombol edit pada bagian daftar pelanggaran siswa. • Mengubah informasi jumlah poin dengan perubahan nilai yang tidak valid. • Menekan tombol <i>submit</i>. • Menekan tombol <i>submit</i> pada bagian <i>modal</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal mengubah informasi poin pelanggaran. <p>Test Data:</p> <p>Data Awal:</p> <ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Tidak memakai atribut sekolah dengan lengkap • Kategori: Ringan • Jumlah Poin: 5 • Tanggal: 20 April 2021 • Tindak Lanjut: Diberikan pengarahan agar tidak mengulangi pelanggaran kembali. <p>Data Edit:</p> <ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Tidak memakai atribut sekolah dengan lengkap • Kategori: Ringan • Jumlah Poin: 0 • Tanggal: 20 April 2021 			





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

<ul style="list-style-type: none"> • Tindak Lanjut: Diberikan pengarahan agar tidak mengulangi pelanggaran kembali. 			
1.	Aktor menekan tombol edit pada bagian daftar pelanggaran siswa.	Sistem menampilkan halaman edit poin pelanggaran. Pada halaman edit poin pelanggaran terdapat edit teks yang sudah terisi nilai masing-masing yaitu pelanggaran yang dilakukan, <i>dropdown</i> kategori, edit teks jumlah poin, edit teks tanggal, tombol kembali dan tombol <i>submit</i> . Nilai dari masing-masing edit teks dan <i>dropdown</i> sesuai dengan <i>test data</i> , data awal.	✓
2.	Aktor mengubah informasi 5 pada edit teks jumlah poin dengan nilai 0.	Masukan aktor tampil pada edit teks jumlah poin.	✓
3.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan <i>modal</i> untuk mengkonfirmasi perubahan poin pelanggaran.	✓
4.	Aktor menekan tombol <i>submit</i> .	Sistem tetap menampilkan halaman edit poin pelanggaran diikuti dengan menampilkan pesan apabila masukan jumlah poin harus lebih dari 0.	✓

Test Condition 2 – Tidak dapat mengubah poin pelanggaran jika perubahan informasi poin pelanggaran adalah menghilangkan nilai informasi poin pelanggaran.





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman lihat pelanggaran siswa. Actor: Guru. Scenario Test:</p> <ul style="list-style-type: none"> Menekan tombol edit pada bagian daftar pelanggaran siswa. Mengubah informasi atau menghapus nilai pada edit teks pelanggaran yang dilakukan dan jumlah poin. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada bagian <i>modal</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor gagal mengubah informasi poin pelanggaran. <p>Test Data:</p> <p>Data Awal:</p> <ul style="list-style-type: none"> Pelanggaran yang Dilakukan: Tidak memakai atribut sekolah dengan lengkap Kategori: Ringan Jumlah Poin: 5 Tanggal: 20 April 2021 Tindak Lanjut: Diberikan pengarahan agar tidak mengulangi pelanggaran kembali. <p>Data Edit:</p> <ul style="list-style-type: none"> Pelanggaran yang Dilakukan: - Kategori: Ringan Jumlah Poin: - Tanggal: 20 April 2021 Tindak Lanjut: - 			
1.	Aktor menekan tombol edit pada bagian daftar pelanggaran siswa.	Sistem menampilkan halaman edit poin pelanggaran. Pada halaman edit poin pelanggaran terdapat edit teks yang sudah terisi nilai masing-masing yaitu pelanggaran yang dilakukan, <i>dropdown</i> kategori, edit teks jumlah poin, edit teks tanggal, tombol kembali dan tombol <i>submit</i> . Nilai	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

		dari masing-masing edit teks dan <i>dropdown</i> sesuai dengan test data, data awal.	
2.	Aktor mengubah informasi dengan menghapus informasi pelanggaran yang dilakukan Tidak memakai atribut sekolah dengan lengkap.	Edit teks pelanggaran yang dilakukan kosong.	✓
3.	Aktor mengubah informasi dengan menghapus informasi jumlah poin 5.	Edit teks jumlah poin kosong.	✓
4.	Aktor mengubah informasi dengan menghapus teks tindak lanjut.	Edit teks tindak lanjut yang dilakukan kosong.	✓
5.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan <i>modal</i> untuk mengkonfirmasi perubahan poin pelanggaran.	✓
6.	Aktor menekan tombol <i>submit</i> .	Sistem tetap menampilkan halaman edit poin pelanggaran diikuti dengan menampilkan pesan apabila informasi poin pelanggaran harus diisi dengan lengkap.	✓

Test Condition 3 – Dapat mengubah informasi poin pelanggaran jika perubahan informasi poin pelanggaran valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
-----------	------------------	-----------------	--------------





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS ILMU KOMPUTER
JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

<p>Pre-Condition: Guru berada di halaman lihat pelanggaran siswa.</p> <p>Actor: Guru.</p> <p>Scenario Test:</p> <ul style="list-style-type: none"> • Menekan tombol edit pada bagian daftar pelanggaran siswa. • Mengubah informasi pelanggaran yang dilakukan, kategori, jumlah poin, dan tanggal dengan perubahan informasi yang valid. • Menekan tombol <i>submit</i>. • Menekan tombol <i>submit</i> pada bagian <i>modal</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor berhasil mengubah informasi poin pelanggaran siswa. <p>Test Data:</p> <p>Data Awal:</p> <ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Tidak memakai atribut sekolah dengan lengkap • Kategori: Ringan • Jumlah Poin: 5 • Tanggal: 20 April 2021 • Tindak Lanjut: Diberikan pengarahan agar tidak mengulangi pelanggaran kembali. <p>Data Edit:</p> <ul style="list-style-type: none"> • Pelanggaran yang Dilakukan: Tidak memakai topi dan terlambat • Kategori: Sedang • Jumlah Poin: 10 • Tanggal: 21 April 2021 • Tindak Lanjut: Diperintahkan untuk membersihkan halaman sekolah. 			
1.	<p>Aktor menekan tombol edit pada bagian daftar pelanggaran siswa.</p>	<p>Sistem menampilkan halaman edit poin pelanggaran. Pada halaman edit poin pelanggaran terdapat edit teks yang sudah terisi nilai masing-masing yaitu pelanggaran yang dilakukan, <i>dropdown</i> kategori, edit teks jumlah poin, edit teks tanggal, tombol kembali dan tombol <i>submit</i>. Nilai dari masing-masing edit teks dan <i>dropdown</i></p>	



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

		sesuai dengan test data, data awal.	
2.	Aktor mengubah informasi Tidak memakai atribut sekolah dengan lengkap pada edit teks pelanggaran yang dilakukan dengan nilai Tidak memakai topi dan terlambat.	Masukan aktor tampil pada edit teks pelanggaran yang dilakukan.	✓
3.	Aktor mengubah informasi Ringan pada dropdown kategori dengan nilai Sedang.	Pilihan aktor tampil pada dropdown kategori.	✓
4.	Aktor mengubah informasi 5 pada edit teks jumlah poin dengan nilai 10.	Masukan aktor tampil pada edit teks jumlah poin.	✓
5.	Aktor mengubah informasi 20 April 2021 pada edit teks tanggal dengan nilai 21 April 2021.	Masukan aktor tampil pada edit teks tanggal.	✓
6.	Aktor mengubah informasi pada edit teks tindak lanjut sesuai dengan data tes edit tindak lanjut.	Masukan aktor tampil pada edit teks tindak lanjut.	✓
6.	Aktor menekan tombol submit.	Sistem menampilkan modal untuk mengkonfirmasi perubahan poin pelanggaran.	✓
7.	Aktor menekan tombol submit.	Sistem menampilkan halaman pelanggaran siswa diikuti dengan menampilkan pesan apabila telah berhasil memperbarui data poin pelanggaran.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Hapus Poin Pelanggaran

• *Test Conditions:*

1. Tidak dapat menghapus poin pelanggaran sesuai dengan data yang dipilih jika membatalkan proses hapus poin pelanggaran.
2. Dapat menghapus poin pelanggaran sesuai dengan data yang dipilih.

Test Condition 1 – Tidak dapat menghapus poin pelanggaran sesuai dengan data yang dipilih jika membatalkan proses hapus poin pelanggaran.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman lihat pelanggaran siswa. <i>Actor:</i> Guru. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> • Menekan tombol hapus pada bagian daftar pelanggaran siswa. • Menekan tombol <i>close</i> pada bagian <i>modal</i> konfirmasi. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal menghapus poin pelanggaran. <p><i>Test Data:</i> -</p>			
1.	Aktor menekan tombol hapus pada bagian daftar pelanggaran siswa.	Sistem menampilkan <i>modal</i> konfirmasi hapus poin pelanggaran. Pada <i>modal</i> tersebut terdapat tombol <i>submit</i> dan <i>close</i> .	✓
2.	Aktor menekan tombol <i>close</i> pada <i>modal</i> konfirmasi.	Sistem menampilkan halaman lihat pelanggaran siswa dan pelanggaran tidak terhapus.	✓

Test Condition 2 – Dapat menghapus poin pelanggaran sesuai dengan data yang dipilih.

Test Step	Test Description	Expected Result	Sukses/Gagal
-----------	------------------	-----------------	--------------





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

<p><i>Pre-Condition:</i> Guru berada di halaman lihat pelanggaran siswa. <i>Actor:</i> Guru. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Menekan tombol hapus pada bagian daftar pelanggaran siswa. Menekan tombol <i>submit</i> pada bagian <i>modal</i> konfirmasi. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor berhasil menghapus poin pelanggaran. <p><i>Test Data:</i> -</p>			
1.	Aktor menekan tombol hapus pada bagian daftar pelanggaran siswa.	Sistem menampilkan <i>modal</i> konfirmasi hapus poin pelanggaran. Pada <i>modal</i> tersebut terdapat tombol <i>submit</i> dan <i>close</i> .	✓
2.	Aktor menekan tombol <i>submit</i> pada <i>modal</i> konfirmasi.	Sistem menampilkan halaman lihat pelanggaran siswa diikuti dengan pesan bahwa proses hapus poin pelanggaran telah berhasil.	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

Pengujian Tambah Motivasi

- **Test Conditions:**
 1. Tidak dapat menyimpan motivasi jika membatalkan proses tambah motivasi.
 2. Dapat menambah motivasi.

Test Condition 1 – Tidak dapat menyimpan motivasi jika membatalkan proses tambah motivasi.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman lihat pelanggaran siswa. Actor: Guru. Scenario Test:</p> <ul style="list-style-type: none"> • Menekan tombol edit pada bagian motivasi dan penguatan. • Memasukkan teks motivasi. • Menekan tombol <i>cancel</i> pada bagian motivasi dan penguatan. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal menambah motivasi. <p>Test Data: -</p>			
1.	Aktor menekan tombol edit pada bagian motivasi dan penguatan.	Sistem mengubah teks area yang awalnya <i>disabled</i> menjadi <i>fillable</i> .	✓
2.	Aktor memasukkan motivasi pada teks motivasi.	Masukan aktor tampil pada teks area motivasi.	✓
3.	Aktor menekan tombol <i>cancel</i> pada bagian motivasi dan penguatan.	Sistem mengubah teks area yang <i>fillable</i> menjadi <i>disabled</i> kembali.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

Test Condition 2 – Dapat menambah motivasi.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman lihat pelanggaran siswa. Actor: Guru. Scenario Test:</p> <ul style="list-style-type: none"> Menekan tombol edit pada bagian motivasi dan penguatan. Memasukkan teks motivasi. Menekan tombol simpan pada bagian motivasi dan penguatan. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor berhasil menambah motivasi. <p>Test Data: -</p>			
1.	Aktor menekan tombol edit pada bagian motivasi dan penguatan.	Sistem mengubah teks area yang awalnya <i>disabled</i> menjadi <i>fillable</i> .	✓
2.	Aktor memasukkan motivasi pada teks motivasi.	Masukan aktor tampil pada teks area motivasi.	✓
3.	Aktor menekan tombol simpan pada bagian motivasi dan penguatan.	Sistem mengubah teks area yang <i>fillable</i> menjadi <i>disabled</i> kembali diikuti dengan menampilkan pesan bahwa motivasi berhasil ditambahkan.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

Pengujian Tambah Akun Siswa

- **Test Conditions:**
 1. Tidak dapat menambah akun siswa jika NIS atau *email* tidak valid.
 2. Tidak dapat menambah akun siswa jika *password* tidak valid.
 3. Dapat mengakses halaman konfirmasi tambah akun siswa jika informasi tambah akun siswa valid.
 4. Tidak dapat menambahkan akun siswa jika pada halaman konfirmasi akun aktor tidak mengonfirmasi tambah akun siswa.
 5. Dapat menambahkan akun siswa jika pada halaman konfirmasi akun siswa aktor mengonfirmasi tambah akun siswa.

Test Condition 1 – Tidak dapat menambah akun siswa jika NIS atau *email* tidak valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman kelola siswa. <i>Actor:</i> Guru. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> • Menekan tombol tambah akun siswa pada bagian tambah akun siswa. • Memasukkan NIS atau <i>email</i> dengan data yang salah. • Memasukkan <i>set password</i> dan konfirmasi <i>password</i>. • Menekan tombol cek data. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal mengakses halaman konfirmasi tambah akun siswa. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> • NIS: 95864303 • <i>Email:</i> sodiq.jobseeker@gmail.com • <i>Set Password:</i> password123 • <i>Konfirmasi Password:</i> password123 			
1.	Aktor menekan tombol tambah akun siswa pada bagian tambah akun siswa.	Sistem menampilkan halaman tambah akun siswa. Pada halaman tambah akun siswa terdapat edit teks kosong NIS, <i>email</i> , <i>set</i>	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS ILMU KOMPUTER
JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
JL. Veteran No.8, Malang, 65145, Indonesia
Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

		<i>password</i> , konfirmasi <i>password</i> , tombol kembali dan tombol cek data.	
2.	Aktor memasukkan 95864303 pada edit teks NIS.	Masukan aktor tampil pada edit teks NIS.	✓
3.	Aktor memasukkan sodiq.jobseeker@gmail.com pada edit teks <i>email</i> .	Masukan aktor tampil pada edit teks <i>email</i> .	✓
4.	Aktor memasukkan password123 pada edit teks <i>set password</i> .	Masukan aktor tampil pada edit teks <i>set password</i> .	✓
5.	Aktor memasukkan password123 pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> .	✓
6.	Aktor menekan tombol cek data.	Sistem tetap menampilkan halaman tambah akun siswa diikuti pesan NIS salah atau sudah terdaftar.	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

Test Condition 2 – Tidak dapat menambah akun siswa jika password tidak valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman kelola siswa. Actor: Guru. Scenario Test:</p> <ul style="list-style-type: none"> Menekan tombol tambah akun siswa pada bagian tambah akun siswa. Memasukkan NIS atau email. Memasukkan set password dan konfirmasi password dengan data yang salah. Menekan tombol cek data. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor gagal mengakses halaman konfirmasi tambah akun siswa. <p>Test Data:</p> <ul style="list-style-type: none"> NIS: 419392974 Email: clara@gmail.com Set Password: pwd12 Konfirmasi Password: pwd12 			
1.	Aktor menekan tombol tambah akun siswa pada bagian tambah akun siswa.	Sistem menampilkan halaman tambah akun siswa. Pada halaman tambah akun siswa terdapat edit teks kosong NIS, email, set password, konfirmasi password, tombol kembali dan tombol cek data.	✓
2.	Aktor memasukkan 419392974 pada edit teks NIS.	Masukan aktor tampil pada edit teks NIS.	✓
3.	Aktor memasukkan clara@gmail.com pada edit teks email.	Masukan aktor tampil pada edit teks email.	✓
4.	Aktor memasukkan pwd12 pada edit teks set password.	Masukan aktor tampil pada edit teks set password.	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

5.	Aktor memasukkan pwd12 pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> .	✓
6.	Aktor menekan tombol cek data.	Sistem menampilkan peringatan bahwa masukan <i>password</i> baru harus mempunyai minimal 6 karakter.	✓

Test Condition 3 – Dapat mengakses halaman konfirmasi tambah akun siswa jika informasi tambah akun siswa valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman kelola siswa. Actor: Guru. Scenario Test:</p> <ul style="list-style-type: none"> Menekan tombol tambah akun siswa pada bagian tambah akun siswa. Memasukkan NIS atau <i>email</i>. Memasukkan <i>set password</i> dan konfirmasi <i>password</i>. Menekan tombol cek data. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor gagal mengakses halaman konfirmasi tambah akun siswa. <p>Test Data:</p> <ul style="list-style-type: none"> NIS: 419392974 <i>Email:</i> clara@gmail.com <i>Set Password:</i> clara000 Konfirmasi <i>Password:</i> clara000 			
1.	Aktor menekan tombol tambah akun siswa pada bagian tambah akun siswa.	Sistem menampilkan halaman tambah akun siswa. Pada halaman tambah akun siswa terdapat edit teks kosong NIS, <i>email</i> , <i>set password</i> , konfirmasi <i>password</i> , tombol kembali dan tombol cek data.	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

2.	Aktor memasukkan 419392974 pada edit teks NIS.	Masukan aktor tampil pada edit teks NIS.	✓
3.	Aktor memasukkan clara@gmail.com pada edit teks email.	Masukan aktor tampil pada edit teks email.	✓
4.	Aktor memasukkan clara000 pada edit teks set password.	Masukan aktor tampil pada edit teks set password.	✓
5.	Aktor memasukkan clara000 pada edit teks konfirmasi password.	Masukan aktor tampil pada edit teks konfirmasi password.	✓
6.	Aktor menekan tombol cek data.	Sistem menampilkan halaman konfirmasi tambah akun siswa.	✓

Test Condition 4 – Tidak dapat menambahkan akun siswa jika pada halaman konfirmasi akun aktor tidak mengonfirmasi tambah akun siswa.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman konfirmasi tambah akun siswa. Actor: Guru. Scenario Test:</p> <ul style="list-style-type: none"> Menekan tombol konfirmasi pada bagian konfirmasi tambah akun siswa. Menekan tombol close pada bagian modal konfirmasi. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor gagal menambah akun siswa. <p>Test Data: -</p>			
1.	Aktor menekan tombol konfirmasi pada bagian konfirmasi tambah akun siswa.	Sistem menampilkan modal konfirmasi buat akun siswa. Pada modal tersebut terdapat tombol submit dan close.	✓
2.	Aktor menekan tombol close pada modal konfirmasi tambah akun siswa.	Sistem menampilkan halaman konfirmasi tambah akun siswa	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

		kembali dan akun siswa tidak disimpan.	
--	--	--	--

Test Condition 5 – Dapat menambahkan akun siswa jika pada halaman konfirmasi akun siswa aktor mengonfirmasi tambah akun siswa.

Test Step	Test Description	Expected Result	Sukses/Gagal
Pre-Condition: Guru berada di halaman konfirmasi tambah akun siswa. Actor: Guru. Scenario Test: <ul style="list-style-type: none"> Menekan tombol konfirmasi pada bagian konfirmasi tambah akun siswa. Menekan tombol <i>submit</i> pada bagian <i>modal</i> konfirmasi. Expected Result: <ul style="list-style-type: none"> Aktor berhasil menambah akun siswa. Test Data: -			
1.	Aktor menekan tombol konfirmasi pada bagian konfirmasi tambah akun siswa.	Sistem menampilkan <i>modal</i> konfirmasi buat akun siswa. Pada <i>modal</i> tersebut terdapat tombol <i>submit</i> dan <i>close</i> .	✓
2.	Aktor menekan tombol <i>submit</i> pada <i>modal</i> konfirmasi tambah akun siswa.	Sistem menampilkan halaman kelola siswa diikuti dengan menampilkan pesan bahwa proses tambah akun siswa telah berhasil.	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

Pengujian Tambah Data Siswa

- **Test Conditions:**
 1. Tidak dapat menambah data siswa jika format *file* tidak valid.
 2. Tidak dapat menambah data siswa jika data siswa tidak valid.
 3. Dapat menambah data siswa jika format *file* dan data siswa valid.
 4. Tidak dapat menambah data siswa jika terdapat data NIS yang sama pada *database*.

Test Condition 1 – Tidak dapat menambah data siswa jika format *file* tidak valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman kelola siswa. Actor: Guru. Scenario Test:</p> <ul style="list-style-type: none"> • Menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa. • Menekan tombol <i>import</i> pada bagian <i>import</i> data siswa. • Menekan tombol pilih <i>file</i> pada modal <i>import</i> data siswa. • Memilih <i>file</i> berformat TXT. • Menekan tombol <i>import</i> pada modal <i>import</i> data siswa. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal menambah data siswa. <p>Test Data: -</p>			
1.	Aktor menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa.	Sistem menampilkan halaman tambah data siswa. Pada halaman tambah data siswa terdapat tombol <i>download template CSV</i> , tombol <i>download</i> keterangan kelas id, tombol kembali dan tombol <i>import</i> .	✓
2.	Aktor menekan tombol <i>import</i> .	Sistem menampilkan modal <i>import</i> data siswa. Pada modal tersebut terdapat tombol pilih	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

		file, tombol <i>import</i> dan tombol <i>close</i> .	
3.	Aktor menekan tombol pilih file pada modal <i>import</i> data siswa.	Sistem menampilkan <i>window file</i> agar aktor dapat memilih file yang akan diunggah.	✓
4.	Aktor memilih file berformat TXT.	Sistem menampilkan modal <i>import</i> data siswa kembali.	✓
5.	Aktor menekan tombol <i>import</i> pada modal <i>import</i> data siswa.	Sistem menampilkan halaman tambah akun siswa diikuti dengan pesan bahwa terdapat kesalahan pada file yang diunggah.	✓

Test Condition 2 – Tidak dapat menambah data siswa jika data siswa tidak valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman kelola siswa. <i>Actor:</i> Guru. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa. Menekan tombol <i>import</i> pada bagian <i>import</i> data siswa. Menekan tombol pilih file pada modal <i>import</i> data siswa. Memilih file bernama Data Siswa Tidak Valid. Menekan tombol <i>import</i> pada modal <i>import</i> data siswa. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal menambah data siswa. <p><i>Test Data:</i> -</p>			
1.	Aktor menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa.	Sistem menampilkan halaman tambah data siswa. Pada halaman tambah data siswa terdapat tombol <i>download template CSV</i> , tombol <i>download</i> keterangan kelas id,	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

		tombol kembali dan tombol <i>import</i> .	
2.	Aktor menekan tombol <i>import</i> .	Sistem menampilkan <i>modal import</i> data siswa. Pada <i>modal</i> tersebut terdapat tombol pilih <i>file</i> , tombol <i>import</i> dan tombol <i>close</i> .	✓
3.	Aktor menekan tombol pilih <i>file</i> pada <i>modal import</i> data siswa.	Sistem menampilkan <i>window file</i> agar aktor dapat memilih <i>file</i> yang akan diunggah.	✓
4.	Aktor memilih <i>file</i> berformat bernama Data Siswa Tidak Valid.	Sistem menampilkan <i>modal import</i> data siswa kembali.	✓
5.	Aktor menekan tombol <i>import</i> pada <i>modal import</i> data siswa.	Sistem menampilkan halaman tambah akun siswa diikuti dengan pesan bahwa terdapat kesalahan data <i>file</i> bari ke ... dan kolom	✓

Test Condition 3 – Dapat menambah data siswa jika format *file* dan data siswa valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru berada di halaman kelola siswa. Actor: Guru. Scenario Test:</p> <ul style="list-style-type: none"> Menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa. Menekan tombol <i>import</i> pada bagian <i>import</i> data siswa. Menekan tombol pilih <i>file</i> pada <i>modal import</i> data siswa. Memilih <i>file</i> bernama Data Siswa Valid. Menekan tombol <i>import</i> pada <i>modal import</i> data siswa. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor berhasil menambah data siswa. <p>Test Data: -</p>			





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA
FAKULTAS ILMU KOMPUTER
JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

JL. Veteran No.8, Malang, 65145, Indonesia
Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

1.	Aktor menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa.	Sistem menampilkan halaman tambah data siswa. Pada halaman tambah data siswa terdapat tombol <i>download template CSV</i> , tombol <i>download</i> keterangan kelas id, tombol kembali dan tombol <i>import</i> .	✓
2.	Aktor menekan tombol <i>import</i> .	Sistem menampilkan <i>modal import</i> data siswa. Pada <i>modal</i> tersebut terdapat tombol pilih <i>file</i> , tombol <i>import</i> dan tombol <i>close</i> .	✓
3.	Aktor menekan tombol pilih <i>file</i> pada <i>modal import</i> data siswa.	Sistem menampilkan <i>window file</i> agar aktor dapat memilih <i>file</i> yang akan diunggah.	✓
4.	Aktor memilih <i>file</i> berformat bernama Data Siswa Valid.	Sistem menampilkan <i>modal import</i> data siswa kembali.	✓
5.	Aktor menekan tombol <i>import</i> pada <i>modal import</i> data siswa.	Sistem menampilkan halaman daftar siswa diikuti dengan pesan bahwa proses tambah data siswa telah berhasil.	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

Test Condition 4 – Tidak dapat menambah data siswa jika terdapat data NIS yang sama pada database.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman kelola siswa. <i>Actor:</i> Guru. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa. Menekan tombol <i>import</i> pada bagian <i>import</i> data siswa. Menekan tombol pilih <i>file</i> pada <i>modal import</i> data siswa. Memilih <i>file</i> bernama Data Siswa Valid. Menekan tombol <i>import</i> pada <i>modal import</i> data siswa. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal menambah data siswa. <p><i>Test Data:</i> -</p>			
1.	Aktor menekan tombol <i>import</i> data siswa pada bagian <i>import</i> data siswa.	Sistem menampilkan halaman tambah data siswa. Pada halaman tambah data siswa terdapat tombol <i>download template CSV</i> , tombol <i>download</i> keterangan kelas id, tombol kembali dan tombol <i>import</i> .	✓
2.	Aktor menekan tombol <i>import</i> .	Sistem menampilkan <i>modal import</i> data siswa. Pada <i>modal</i> tersebut terdapat tombol pilih <i>file</i> , tombol <i>import</i> dan tombol <i>close</i> .	✓
3.	Aktor menekan tombol pilih <i>file</i> pada <i>modal import</i> data siswa.	Sistem menampilkan <i>window file</i> agar aktor dapat memilih <i>file</i> yang akan diunggah.	✓
4.	Aktor memilih <i>file</i> berformat bernama Data Siswa Valid.	Sistem menampilkan <i>modal import</i> data siswa kembali.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS ILMU KOMPUTER
JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

JL. Veteran No.8, Malang, 65145, Indonesia
Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

5.	Aktor menekan tombol <i>import</i> pada <i>modal import</i> data siswa.	Sistem menampilkan halaman tambah akun siswa diikuti dengan pesan bahwa terdapat data yang sama pada <i>database</i> dengan data yang diunggah serta menampilkan daftar siswa yang memiliki data yang sama tersebut.	✓
----	---	--	---



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Cetak Poin Pelanggaran

- *Test Conditions:*
 1. Dapat mencetak atau menyimpan poin pelanggaran.

Test Condition 1 – Dapat mencetak atau menyimpan poin pelanggaran.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru berada di halaman lihat pelanggaran siswa. <i>Actor:</i> Guru. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> • Menekan tombol cetak poin pelanggaran pada bagian data siswa. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor berhasil mencetak atau menyimpan poin pelanggaran. <p><i>Test Data:</i> -</p>			
1.	Aktor menekan tombol cetak poin pelanggaran pada bagian data siswa.	Sistem menampilkan <i>preview</i> hasil dokumen pada <i>tab</i> baru <i>browser</i> diikuti dengan tetap menampilkan halaman pelanggaran siswa.	✓

Pasuruan, 23.11.2021

Nama Wahyu Widayat, S.Pd
 NIP./NIK. 19741024 200904 1 001



LAMPIRAN D HASIL PENGUJIAN VALIDASI OLEH SISWA



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

PENGUJIAN VALIDASI APLIKASI TATA TERTIB PELANGGARAN KUESIONER UNTUK SISWA

Pengujian validasi ini bertujuan untuk menguji fungsional/menu dari aplikasi tata tertib pelanggaran. Pada pengujian ini proses berhasil dan gagal akan diujikan. Berikut langkah-langkah yang harus diperhatikan selama proses pengujian dilakukan:

1. Baca *test conditions* terlebih dahulu pada setiap pengujian yang akan dilakukan. *Test conditions* memberikan informasi terkait tujuan dari pengujian yang dilakukan pada setiap fungsional yang diujikan.
2. Pada setiap *test conditions* terdapat tabel yang berisi informasi terkait aktor, skenario tes (langkah apa saja yang akan dilakukan), hasil yang diharapkan, data tes (jika ada), dan langkah-langkah melakukan pengujian.
3. Informasi data tes digunakan untuk mengisikan edit teks yang terdapat pada setiap fungsional yang diujikan.
4. Baca *test description* (langkah yang dilakukan) dan *expected result* (hasil yang diharapkan) yang terdapat pada setiap tabel *test conditions*.
5. Berikan simbol centang (✓) jika sukses dan (x) jika gagal pada setiap kolom berdasarkan *test description* dan *expected result*.





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Masuk

- *Test Conditions:*
 1. Tidak dapat melakukan masuk jika *username* atau *password* siswa salah.
 2. Dapat melakukan masuk jika *username* dan *password* siswa benar.

Test Condition 1 – Tidak dapat melakukan masuk jika *username* atau *password* siswa salah.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman <i>browser</i>. <i>Actor:</i> Pengguna. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> • Mengakses alamat URL SiTalang (tatatertibpelanggaran.000webhostapp.com). • Memasukkan <i>username</i> atau <i>password</i> dengan data yang salah. • Menekan tombol masuk. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor gagal masuk ke halaman utama aplikasi dan tetap berada di halaman masuk. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> • <i>Username:</i> 95864303 • <i>Password:</i> salah123 			
1.	Aktor mengakses alamat URL SiTalang.	Sistem menampilkan halaman masuk aplikasi. Pada halaman masuk terdapat edit teks kosong <i>username</i> , edit teks kosong <i>password</i> , link reset <i>password</i> , tombol masuk, dan gambar.	✓
2.	Aktor memasukkan 95864303 pada edit teks <i>username</i>	Masukan aktor tampil pada edit teks <i>username</i> .	✓
3.	Aktor memasukkan salah123 pada edit teks <i>password</i>	Masukan aktor tampil pada edit teks <i>password</i> dengan karakter *	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

4.	Aktor menekan tombol masuk.	Sistem menampilkan peringatan bahwa <i>username</i> atau <i>password</i> salah.	✓
----	-----------------------------	---	---

Test Condition 2 - Dapat melakukan masuk jika *username* dan *password* siswa benar.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman <i>browser</i>. <i>Actor:</i> Pengguna. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Mengakses alamat URL SiTalang (tatatertibpelanggaran.000webhostapp.com). Memasukkan <i>username</i> dan <i>password</i> dengan data yang benar. Menekan tombol masuk. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor berhasil masuk ke halaman utama aplikasi. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> <i>Username:</i> 95864303 <i>Password:</i> alika000 			
1.	Aktor mengakses alamat URL SiTalang.	Sistem menampilkan halaman masuk aplikasi. Pada halaman masuk terdapat edit teks kosong <i>username</i> , edit teks kosong <i>password</i> , <i>link</i> reset <i>password</i> , tombol masuk, dan gambar.	✓
2.	Aktor memasukkan 95864303 pada edit teks <i>username</i>	Masukan aktor tampil pada edit teks <i>username</i> .	✓
3.	Aktor memasukkan alika000 pada edit teks <i>password</i>	Masukan aktor tampil pada edit teks <i>password</i> dengan karakter *.	✓
4.	Aktor menekan tombol masuk.	Sistem menampilkan halaman utama siswa bernama Alika.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Lihat Pelanggaran

- **Test Conditions:**
 1. Dapat melihat data siswa dan informasi poin pelanggaran siswa sesuai data siswa yang melakukan masuk.

Test Condition 1 – Dapat melihat data siswa dan informasi poin pelanggaran siswa sesuai data siswa yang melakukan masuk.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Siswa berada di halaman utama aplikasi. Actor: Siswa. Scenario Test:</p> <ul style="list-style-type: none"> • Melakukan masuk. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor berhasil melihat data siswa dan informasi poin pelanggaran siswa yang sesuai dengan siswa yang melakukan masuk. <p>Test Data: -</p>			
1.	Aktor telah melakukan masuk.	Sistem menampilkan halaman lihat pelanggaran. Pada halaman lihat pelanggaran terdapat data profil siswa dan informasi poin pelanggaran siswa yang sesuai dengan siswa yang melakukan masuk.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Keluar

- *Test Conditions:*
 1. Dapat keluar dari sistem.

Test Condition 1 – Dapat keluar dari sistem.

Test Step	Test Description	Expected Result	Sukses/Gagal
<i>Pre-Condition:</i> Guru/Siswa berada di halaman utama aplikasi. <i>Actor:</i> Guru, Siswa. <i>Scenario Test:</i> <ul style="list-style-type: none"> • Menekan tombol keluar pada bagian <i>navigation bar</i>. <i>Expected Result:</i> <ul style="list-style-type: none"> • Aktor berhasil keluar dari sistem. <i>Test Data:</i> -			
1.	Aktor menekan tombol keluar pada bagian <i>navigation bar</i> .	Sistem menampilkan halaman masuk.	✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Reset Password

- **Test Conditions:**
 1. Tidak dapat mengirimkan kode jika *email* pengguna salah.
 2. Dapat mengirimkan kode jika *email* pengguna benar.
 3. Tidak dapat mengakses halaman reset *password* jika kode konfirmasi salah.
 4. Dapat mengakses halaman reset *password* jika kode konfirmasi benar.
 5. Tidak dapat mengatur ulang informasi *password* jika masukan tidak valid.
 6. Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.
 7. Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru kurang dari 6 karakter.
 8. Dapat mengatur ulang informasi *password* jika masukan valid.

Test Condition 1 – Tidak dapat mengirimkan kode jika *email* pengguna salah.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman masuk. Actor: Pengguna. Scenario Test:</p> <ul style="list-style-type: none"> • Menekan <i>link</i> bertuliskan Reset Password! • Memasukkan <i>email</i> dengan data yang salah. • Menekan tombol kirim kode. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal mengakses halaman konfirmasi kode dan tidak menerima kode konfirmasi melalui <i>email</i>. <p>Test Data:</p> <ul style="list-style-type: none"> • <i>Email:</i> email.random@gmail.com 			
1.	Pengguna menekan <i>link</i> bertuliskan Reset Password!	Sistem menampilkan halaman verifikasi <i>email</i> . Pada halaman verifikasi <i>email</i> terdapat edit teks kosong <i>email</i> , tombol bertuliskan kirim kode,	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

		dan <i>link</i> bertuliskan kembali ke halaman masuk.	
2.	Pengguna memasukkan email.random@gmail.com pada edit teks <i>email</i> .	Masukan aktor tampil pada edit teks <i>email</i> .	✓
3.	Pengguna menekan tombol kirim kode.	Sistem menampilkan peringatan bahwa <i>email</i> salah.	✓

Test Condition 2 - Dapat mengirimkan kode jika *email* pengguna benar.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman masuk. Actor: Pengguna. Scenario Test:</p> <ul style="list-style-type: none"> Menekan <i>link</i> bertuliskan Reset Password!. Memasukkan <i>email</i> dengan data yang benar. Menekan tombol kirim kode. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor berhasil mengakses halaman konfirmasi kode dan menerima kode konfirmasi melalui <i>email</i>. <p>Test Data:</p> <ul style="list-style-type: none"> <i>Email:</i> sodiq.jobseeker@gmail.com 			
1.	Pengguna menekan <i>link</i> bertuliskan Reset Password!	Sistem menampilkan halaman verifikasi <i>email</i> . Pada halaman verifikasi <i>email</i> terdapat edit teks kosong <i>email</i> , tombol bertuliskan kirim kode, dan <i>link</i> bertuliskan kembali ke halaman masuk.	✓
2.	Pengguna memasukkan sodiq.jobseeker@gmail.com pada edit teks <i>email</i> .	Masukan aktor tampil pada edit teks <i>email</i> .	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

3.	Pengguna menekan tombol kirim kode.	Sistem menampilkan halaman konfirmasi kode dan mengirimkan kode konfirmasi melalui <i>email</i> diikuti dengan pesan pemberitahuan untuk mengecek <i>email</i> aktor untuk melihat kode konfirmasi.	✓
----	-------------------------------------	---	---

Test Condition 3 – Tidak dapat mengakses halaman reset *password* jika kode konfirmasi salah.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman konfirmasi kode. Actor: Pengguna. Scenario Test:</p> <ul style="list-style-type: none"> Memasukkan kode konfirmasi dengan data yang salah. Menekan tombol <i>submit</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> Aktor gagal mengakses halaman reset <i>password</i>. <p>Test Data:</p> <ul style="list-style-type: none"> Kode: Tidak sesuai dengan kode yang diterima melalui <i>email</i> (<i>random</i>). 			
1.	Pengguna memasukkan kode yang tidak sesuai dengan kode yang diterima melalui <i>email</i> pada edit teks kode.	Masukan aktor tampil pada edit teks kode.	✓
2.	Pengguna menekan tombol <i>submit</i> .	Sistem menampilkan peringatan bahwa kode tidak sesuai.	✓

Test Condition 4 – Dapat mengakses halaman reset *password* jika kode konfirmasi benar.

Test Step	Test Description	Expected Result	Sukses/Gagal
-----------	------------------	-----------------	--------------





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

<p><i>Pre-Condition:</i> Pengguna berada di halaman konfirmasi kode. <i>Actor:</i> Pengguna. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> • Memasukkan kode konfirmasi dengan data yang benar. • Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> • Aktor berhasil mengakses halaman reset <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> • Kode: Sesuai dengan kode yang diterima melalui <i>email</i>. 			
1.	Pengguna memasukkan kode yang diterima pada edit teks kode.	Masukan aktor tampil pada edit teks kode.	✓
2.	Pengguna menekan tombol <i>submit</i> .	Sistem menampilkan halaman reset <i>password</i> . Pada halaman reset <i>password</i> terdapat edit teks kosong <i>password</i> baru, <i>konfirmasi password</i> , tombol mata, tombol <i>submit</i> , dan <i>link</i> kembali ke halaman masuk diikuti dengan pesan apabila kode sesuai.	✓

Test Condition 5 – Tidak dapat mengatur ulang informasi *password* jika masukan tidak valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
-----------	------------------	-----------------	--------------





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pre-Condition: Pengguna berada di halaman reset *password*.
Actor: Pengguna.
Scenario Test:

- Memasukkan *password* baru dan konfirmasi *password* dengan masukan yang tidak valid.
- Menekan tombol *submit*.

Expected Result:

- Aktor gagal mengatur ulang informasi *password*.

Test Data:

- Password* baru: *passwordtidakvalid*
- Konfirmasi *Password*: *passwordtidakvalid*

1.	Pengguna memasukkan <i>password</i> tidak valid pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
2.	Pengguna memasukkan <i>password</i> tidak valid pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓
3.	Pengguna menekan tombol <i>submit</i> .	Sistem menampilkan peringatan bahwa masukan aktor tidak valid.	✓

Test Condition 6 – Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman reset <i>password</i>. <i>Actor:</i> Pengguna. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang berbeda. Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal mengatur ulang informasi <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> <i>Password</i> baru: <i>passwordsiswa123</i> Konfirmasi <i>Password</i>: <i>passwordsiswa456</i> 			





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
<http://filkom.ub.ac.id> E-mail : jsi.filkom@ub.ac.id

1.	Pengguna memasukkan passwordsiswa123 pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
2.	Pengguna memasukkan passwordsiswa456 pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓
3.	Pengguna menekan tombol <i>submit</i> .	Sistem menampilkan peringatan bahwa masukan <i>password</i> baru dan konfirmasi <i>password</i> harus sama.	✓

Test Condition 7 – Tidak dapat mengatur ulang informasi *password* jika masukan *password* baru kurang dari 6 karakter.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Pengguna berada di halaman reset <i>password</i>. Actor: Pengguna. Scenario Test:</p> <ul style="list-style-type: none"> • Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan kurang dari 6 karakter. • Menekan tombol <i>submit</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal mengubah informasi <i>password</i>. <p>Test Data:</p> <ul style="list-style-type: none"> • <i>Password</i> baru: tes12 • Konfirmasi <i>Password</i>: tes12 			
1.	Aktor memasukkan tes12 pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
2.	Aktor memasukkan tes12 pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓
3.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan peringatan bahwa	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

		masukan <i>password</i> baru harus mempunyai minimal 6 karakter.	
--	--	--	--

Test Condition 8 – Dapat mengatur ulang informasi *password* jika masukan valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Pengguna berada di halaman reset <i>password</i>. <i>Actor:</i> Pengguna. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang valid. Menekan tombol <i>submit</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor berhasil mengatur ulang informasi <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> <i>Password</i> baru: passwordbaru111 Konfirmasi <i>Password:</i> passwordbaru111 			
1.	Pengguna memasukkan passwordbaru111 pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
2.	Pengguna memasukkan passwordbaru111 pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓
3.	Pengguna menekan tombol <i>submit</i> .	Sistem menampilkan halaman masuk diikuti dengan pesan bahwa informasi <i>password</i> telah diperbarui.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

Pengujian Fungsional Ubah Password

- **Test Conditions:**
 1. Tidak dapat mengubah informasi *password* jika masukan tidak valid.
 2. Tidak dapat mengubah informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.
 3. Tidak dapat mengubah informasi *password* jika masukan *password* baru kurang dari 6 karakter.
 4. Dapat mengubah informasi *password* jika masukan valid.

Test Condition 1 – Tidak dapat mengubah informasi *password* baru jika masukan tidak valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p>Pre-Condition: Guru/Siswa berada di halaman utama aplikasi. Actor: Guru, Siswa. Scenario Test:</p> <ul style="list-style-type: none"> • Menekan tombol ubah <i>password</i> pada bagian <i>navigation bar</i>. • Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang tidak valid. • Menekan tombol <i>submit</i>. • Menekan tombol <i>submit</i> pada <i>modal</i>. <p>Expected Result:</p> <ul style="list-style-type: none"> • Aktor gagal mengubah informasi <i>password</i>. <p>Test Data:</p> <ul style="list-style-type: none"> • <i>Password</i> baru: pwtidakvalid • Konfirmasi <i>Password</i>: pwtidakvalid 			
1.	Aktor memasukkan pwtidakvalid pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
2.	Aktor memasukkan pwtidakvalid pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

3.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan modal konfirmasi ubah <i>password</i> .	✓
4.	Aktor menekan tombol <i>submit</i> pada bagian <i>modal</i> .	Sistem menampilkan peringatan bahwa masukan aktor tidak valid.	✓

Test Condition 2 – Tidak dapat mengubah informasi *password* jika masukan *password* baru tidak sama dengan masukan konfirmasi *password*.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru/Siswa berada di halaman utama aplikasi. <i>Actor:</i> Guru, Siswa. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Menekan tombol ubah <i>password</i> pada bagian <i>navigation bar</i>. Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang berbeda. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal mengubah informasi <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> <i>Password</i> baru: ubahpassword000 Konfirmasi <i>Password:</i> ubahbedapassword123 			
1.	Aktor menekan menu ubah <i>password</i> pada bagian <i>navigation bar</i> .	Sistem menampilkan halaman ubah <i>password</i> . Pada halaman ubah <i>password</i> terdapat edit teks kosong <i>password</i> baru, konfirmasi <i>password</i> , tombol kembali, dan tombol <i>submit</i> .	✓
2.	Aktor memasukkan ubahpassword000 pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

3.	Aktor memasukkan ubahbedapassword123 pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	✓
4.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan modal konfirmasi ubah <i>password</i> .	✓
5.	Aktor menekan tombol <i>submit</i> pada bagian <i>modal</i> .	Sistem menampilkan peringatan bahwa masukan <i>password</i> baru dan konfirmasi <i>password</i> harus sama.	✓

Test Condition 3 – Tidak dapat mengubah informasi *password* jika masukan *password* baru kurang dari 6 karakter.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru/Siswa berada di halaman utama aplikasi. <i>Actor:</i> Guru, Siswa. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Menekan tombol ubah <i>password</i> pada bagian <i>navigation bar</i>. Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan kurang dari 6 karakter. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor gagal mengubah informasi <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> <i>Password</i> baru: pwd12 Konfirmasi <i>Password</i>: pwd12 			
1.	Aktor menekan menu ubah <i>password</i> pada bagian <i>navigation bar</i> .	Sistem menampilkan halaman ubah <i>password</i> . Pada halaman ubah <i>password</i> terdapat edit teks kosong <i>password</i> baru, konfirmasi <i>password</i> , tombol	✓





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

		kembali, dan tombol <i>submit</i> .	
2.	Aktor memasukkan <i>pwd12</i> pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	
3.	Aktor memasukkan <i>pwd12</i> pada edit teks konfirmasi <i>password</i> .	Masukan aktor tampil pada edit teks konfirmasi <i>password</i> dengan karakter *.	
4.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan modal konfirmasi ubah <i>password</i> .	
5.	Aktor menekan tombol <i>submit</i> pada bagian <i>modal</i> .	Sistem menampilkan peringatan bahwa masukan <i>password</i> baru harus mempunyai minimal 6 karakter.	

Test Condition 4 – Dapat mengubah informasi *password* baru jika masukan valid.

Test Step	Test Description	Expected Result	Sukses/Gagal
<p><i>Pre-Condition:</i> Guru/Siswa berada di halaman utama aplikasi. <i>Actor:</i> Guru, Siswa. <i>Scenario Test:</i></p> <ul style="list-style-type: none"> Menekan tombol ubah <i>password</i> pada bagian <i>navigation bar</i>. Memasukkan <i>password</i> baru dan konfirmasi <i>password</i> dengan masukan yang valid. Menekan tombol <i>submit</i>. Menekan tombol <i>submit</i> pada <i>modal</i>. <p><i>Expected Result:</i></p> <ul style="list-style-type: none"> Aktor berhasil mengubah informasi <i>password</i>. <p><i>Test Data:</i></p> <ul style="list-style-type: none"> <i>Password</i> baru: alika000 Konfirmasi <i>Password:</i> alika000 			
1.	Aktor menekan menu ubah <i>password</i> pada bagian <i>navigation bar</i> .	Sistem menampilkan halaman ubah <i>password</i> . Pada halaman ubah <i>password</i> terdapat edit	





KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

		teks kosong <i>password</i> baru, <i>konfirmasi password</i> , tombol kembali, dan tombol <i>submit</i> .	
2.	Aktor memasukkan alika000 pada edit teks <i>password</i> baru.	Masukan aktor tampil pada edit teks <i>password</i> baru dengan karakter *.	✓
3.	Aktor memasukkan alika000 pada edit teks <i>konfirmasi password</i> .	Masukan aktor tampil pada edit teks <i>konfirmasi password</i> dengan karakter *.	✓
4.	Aktor menekan tombol <i>submit</i> .	Sistem menampilkan modal <i>konfirmasi ubah password</i> .	✓
5.	Aktor menekan tombol <i>submit</i> pada bagian <i>modal</i> .	Sistem menampilkan halaman masuk diikuti pesan bahwa informasi <i>password</i> berhasil diubah.	✓

Pasuruan, 23 April, 2021

Nama
 Nama
 Nama

LAMPIRAN E HASIL PENGUJIAN UAT OLEH GURU



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

PENGUJIAN PENERIMAAN APLIKASI TATA TERTIB PELANGGARAN KUESIONER UNTUK GURU

Jawablah pernyataan dengan memberikan simbol centang (✓) pada kolom jawaban yang tersedia sesuai dengan tingkat persetujuan dan keyakinan Anda terhadap setiap pernyataan tentang Aplikasi Tata Tertib Pelanggaran (SiTalang).

- Keterangan :
- SS = Sangat Setuju
 - S = Setuju
 - TS = Tidak Setuju
 - STS = Sangat Tidak Setuju

No.	Pernyataan	Tingkat Persetujuan			
		STS	TS	S	SS
Fungsional					
1.	Fungsional masuk dapat dijalankan dengan baik.				✓
2.	Fungsional reset password dapat dijalankan dengan baik.				✓
3.	Fungsional ubah password dapat dijalankan dengan baik.				✓
4.	Fungsional lihat daftar siswa dapat dijalankan dengan baik.				✓
5.	Fungsional lihat pelanggaran siswa dapat dijalankan dengan baik.				✓
6.	Fungsional tambah poin pelanggaran dapat dijalankan dengan baik.				✓
7.	Fungsional edit poin pelanggaran dapat dijalankan dengan baik.				✓
8.	Fungsional hapus poin pelanggaran dapat dijalankan dengan baik.				✓
9.	Fungsional cetak poin pelanggaran dapat dijalankan dengan baik.				✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI

Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

10.	Fungsional tambah motivasi dapat dijalankan dengan baik.				✓
11.	Fungsional tambah akun siswa dapat dijalankan dengan baik.				✓
12.	Fungsional tambah data siswa dapat dijalankan dengan baik.				✓
13.	Fungsional keluar dapat dijalankan dengan baik.				✓
Penerimaan					
14.	Sistem tata tertib pelanggaran dapat membantu sekolah dalam mengolah poin pelanggaran siswa.			✓	
15.	Sistem tata tertib pelanggaran mudah dimengerti dan mudah untuk digunakan.			✓	
16.	Sistem tata tertib pelanggaran memiliki performa yang baik dalam menanggapi respon pengguna.				✓
17.	Sistem tata tertib pelanggaran sudah sesuai dengan kebutuhan sekolah.			✓	
18.	Sistem tata tertib pelanggaran dapat dijalankan pada platform <i>mobile</i> dan <i>desktop</i> dengan baik.				✓
19.	Tampilan dari sistem tata tertib pelanggaran mudah dipahami.				✓

KOMENTAR DAN SARAN

Sudah bagus

Pasuruan, 23/1/2021

Nama WAHYU WIDAYAT S.Pd

NIP./NIK. 19741024 200904 1 001



LAMPIRAN F HASIL PENGUJIAN UAT OLEH SISWA



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 JL. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

PENGUJIAN PENERIMAAN APLIKASI TATA TERTIB PELANGGARAN KUESIONER UNTUK SISWA

Jawablah pernyataan dengan memberikan simbol centang (✓) pada kolom jawaban yang tersedia sesuai dengan tingkat persetujuan dan keyakinan Anda terhadap setiap pernyataan tentang Aplikasi Tata Tertib Pelanggaran (SITalang).

Keterangan :

- SS = Sangat Setuju
- S = Setuju
- TS = Tidak Setuju
- STS = Sangat Tidak Setuju

No.	Pernyataan	Tingkat Persetujuan			
		STS	TS	S	SS
Fungsional					
1.	Fungsional masuk dapat dijalankan dengan baik.				✓
2.	Fungsional reset password dapat dijalankan dengan baik.				✓
3.	Fungsional ubah password dapat dijalankan dengan baik.				✓
4.	Fungsional lihat pelanggaran dapat dijalankan dengan baik.				✓
5.	Fungsional keluar dapat dijalankan dengan baik.				✓
Penerimaan					
6.	Sistem tata tertib pelanggaran dapat membantu siswa untuk mengetahui info poin pelanggaran siswa.				✓
7.	Sistem tata tertib pelanggaran mudah dimengerti dan mudah untuk digunakan.				✓



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
 UNIVERSITAS BRAWIJAYA
 FAKULTAS ILMU KOMPUTER
 JURUSAN SISTEM INFORMASI
 PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
 Jl. Veteran No.8, Malang, 65145, Indonesia
 Telp. : +62-341-577911; Fax : +62-341-577911
 http://filkom.ub.ac.id E-mail : jsi.filkom@ub.ac.id

8.	Sistem tata tertib pelanggaran memiliki performa yang baik dalam menanggapi respon pengguna.				✓
9.	Sistem tata tertib pelanggaran dapat dijalankan pada platform <i>mobile</i> dan desktop dengan baik.				✓
10.	Tampilan dari sistem tata tertib pelanggaran mudah dipahami.				✓

KOMENTAR DAN SARAN

.....

Pasuruan, 23 April, 2021

Nama Savitiana Lutfi