PENGEMBANGAN SISTEM INFORMASI KEHADIRAN SISWA BERBASIS WEB
(STUDI KASUS : SMA SULUH JAKARTA)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Ade Ayu Puspitasari
NIM: 155150401111012

PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
TAHUN
2019

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI KEHADIRAN SISWA BERBASIS WEB
(STUDI KASUS : SMA SULUH JAKARTA)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Ade Ayu Puspitasari
NIM: 155150401111012

Skripsi ini telah diuji dan dinyatakan lulus pada
16 Mei 2019
Telah diperiksa dan disetujui oleh:

Pembimbing I	Pembimbing II
Ismiarta Aknurandra, S.T., M.Sc., Ph.D.	Nanang Yudi Setiawan, S.T., M.Kom.
NIK: 2010067407191001	NIP: 197606192006041001

Mengetahui
Ketua Jurusan Sistem Informasi

Dr. Eng. Herman Tolle, S.T., M.T.
NIP: 197408232000121001

Scanned with CamScanner

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 didalam 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, 17 Mei 2019

Ade Ayu Puspitasari

NIM: 155150401111012

PRAKATA

Segala puji bagi Allah SWT atas segala limpahan rahmat dan karunianya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Pengembangan Sistem Informasi Kehadiran Siswa Berbasis Web (Studi kasus : SMA Suluh Jakarta)".

Dari penelitian ini, penulis mendapatkan banyak ilmu dan pengetahuan terkait dengan pengembangan sistem informasi yang sebelumnya penulis tidak ketahui. Dengan adanya penelitian ini, penulis juga belajar berkomunikasi dengan pemangku kepentingan untuk mengetahui permasalahan dan solusi yang dibutuhkan. Dalam penulisan dan pelaksanaan penelitian ini, penulis tidak dapat menyelesaikannya tanpa adanya bantuan dan dukungan dari orang-orang terdekat. Oleh karena itu penulis ingin berterimakasih kepada:

1. Bapak Dr. Eng. Herman Tolle, S.T., M.T. selaku Ketua Jurusan Sistem Informasi Universitas Brawijaya Malang.
2. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB. selaku Ketua Program Studi Sistem Informasi Universitas Brawijaya Malang.
3. Bapak Ismiarta Aknuranda, S.T., M.Sc., Ph.D. selaku pembimbing satu yang telah memberikan penulis kesempatan untuk menjadi mahasiswi bimbingannya, serta memberikan bimbingan dengan penuh kesabaran, memberikan solusi, dan memberikan contoh kepada mahasiswa bimbingannya sebagai pribadi yang baik, sabar dan profesional saat memberikan bimbingan.
4. Bapak Nanang Yudi Setiawan, S.T., M.Kom. selaku pembimbing dua yang mengajarkan penulis ketelitian, pengarahan untuk menyelesaikan masalah dan pemberian saran, serta memberikan contoh kepada mahasiswa bimbingannya sebagai pribadi yang baik, sabar, dan profesional saat memberikan bimbingan.
5. Seluruh Bapak dan Ibu dosen, serta karyawan di Fakultas Ilmu Komputer Universitas Brawijaya yang telah mendidik dan membantu penulis selama menempuh pendidikan.
6. Ayah penulis, Agus Wiharta dan ibu penulis, Mariyati yang selalu memberikan dukungan berupa moril dan materil serta doa yang senantiasa dipanjatkan untuk penulis. Serta kakak penulis, Dian Agus Pratama dan Indana Zulfa yang selalu memberikan dukungan dan doanya.
7. Bapak Saifudin, S. Ag, selaku kepala sekolah SMA Suluh Jakarta yang memberikan penulis izin untuk melakukan penelitian di SMA Suluh Jakarta.
8. Ibu Dina Apriliyati, S.Pd, Ibu Dessy Akhriani, S.Pd, dan seluruh guru serta karyawan SMA Suluh Jakarta yang memberikan penulis

kemudahan dalam memperoleh informasi dan melakukan penelitian di SMA Suluh Jakarta.

9. Dwi Nadya Latimar, Liana Andriyani, dan Nurhilaliyah yang memberikan dukungan dan doanya.
10. Teman-teman yang tidak dapat penulis sebutkan satu persatu di jurusan Sistem Informasi, Fakultas Ilmu Komputer yang telah menjadi teman dan sahabat yang selalu mendukung secara moril.

Semoga pihak-pihak yang telah membantu penulis dalam melakukan penelitian ini diberikan limpahan rahmat serta pahala yang berlipat dari Allah SWT. Dalam penelitian yang penulis lakukan, penulis menyadari memiliki banyak kekurangan, sehingga penulis mengharapkan adanya diskusi penelitian selanjutnya serta masukkan berupa saran dan kritik terkait penelitian yang telah penulis lakukan.

Malang, 17 Mei 2019

Penulis

ayupuspitasari4@gmail.com

repository.ub.ac.id

ABSTRAK

Ade Ayu Puspitasari, Pengembangan Sistem Informasi Kehadiran Siswa Berbasis Web (Studi Kasus: SMA Suluh Jakarta)

Pembimbing: Ismiarta Aknurandra, S.T., M.Sc., Ph.D. dan Nanang Yudi Setiawan, S.T., M.Kom.

SMA Suluh Jakarta merupakan sekolah swasta yang berada di bawah naungan Yayasan Hortikultura. Salah satu subbagian yang ada pada SMA Suluh Jakarta yaitu BK. BK memiliki peranan untuk melakukan pengelolaan data kehadiran siswa, namun dalam melakukan pengelolaan data kehadiran siswa belum memanfaatkan teknologi secara optimal. Setiap minggunya BK perlu melakukan perekapan data kehadiran yang diperoleh dari presensi siswa setiap harinya dengan data yang dimasukkan ke dalam Microsoft Excel sebanyak 3.440 data. Saat memasukkan data ke dalam Microsoft Excel rawan terhadap kesalahan-kesalahan karena data yang dimasukkan cukup banyak. Permasalahan lainnya yang dihadapi adalah data tidak selalu tersedia ketika dibutuhkan dan data tidak saling terintegrasi dengan data kehadiran siswa yang terlambat atau tidak hadir dengan memberikan surat keterangan melalui guru piket, sehingga perlu adanya konfirmasi ke guru piket terlebih dahulu. Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah teknologi yang dapat membantu mengelola data kehadiran siswa yaitu sistem informasi. Sistem informasi yang dikembangkan dapat membantu pemangku kepentingan dalam mengelola data kehadiran siswa, sehingga tidak perlu melakukan perekapan data kehadiran siswa lagi karena data yang telah didata akan tersimpan kedalam penyimpanan data terpusat secara otomatis. Data yang tersimpan pada penyimpanan data terpusat akan terintegrasi dengan data yang dimasukkan oleh guru piket, sehingga data dapat terus terbarui dan dapat diakses kapanpun jika diperlukan. Metode yang digunakan untuk mengembangkan Sistem Informasi Kehadiran Siswa menggunakan model waterfall dengan tahapan yang dilakukan yaitu analisis persyaratan, perancangan, implementasi, dan pengujian. Dari hasil pengujian validasi sistem berdasarkan pada persyaratan fungsional didapatkan hasil valid dan pada pengujian kompatibilitas terdapat *critical issues* pada bagian keamanan sistem di peramban Chrome dan Firefox yang disebabkan penggunaan protokol HTTP.

Kata kunci: *sistem informasi kehadiran, waterfall, SMA Suluh Jakarta, pengujian kompatibilitas*

UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRACT

Ade Ayu Puspitasari, Development of Web-based Student Attendance Information System

***Supervisors*: Ismiarta Aknurandra, S.T, M.Sc, Ph.D *and* Nanang Yudi Setiawan, S.T., M.Kom**

Suluh Jakarta Senior High School is a private school under the auspices of the Horticulture Foundation. One of the subsections at SMA Suluh Jakarta is BK. BK has a role to manage student attendance data, but in managing attendance data students have not utilized technology optimally. Every week BK needs to do recording attendance data obtained from student attendance every day with data entered into Microsoft Excel as many as 3,440 data. When entering data into Microsoft Excel it is prone to errors because there is quite a lot of data entered. Another problem faced is that data is not always available when needed and the data is not integrated with the data on the presence of students who are late or absent by giving a certificate through the picket teacher, so it needs confirmation from the picket teacher first. To overcome these problems, we need a technology that can help manage student attendance data, namely information systems. The information system developed can help stakeholders manage student attendance data, so it does not require the recording of student attendance data anymore because the data that has been recorded will be stored in centralized data storage automatically. Data stored on centralized data storage will be integrated with the data entered by the picket teacher, so that data can continue to be updated and can be accessed whenever needed. The method used to develop a Student Attendance Information System uses the waterfall model with the steps taken, namely requirements analysis, design, implementation, and testing. From the results of the system validation test based on functional requirements, the results obtained are valid and in compatibility testing there are critical issues in the system security section of the Chrome browser and Firefox due to the use of the HTTP protocol.

Keywords: *attendance information system, waterfall, Suluh Jakarta Senior High School, compatibility testing*

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ..... **ii**
PERNYATAAN ORISINALITAS ..... **iii**
PRAKATA ..... **iv**
ABSTRAK ..... **vi**
ABSTRACT ..... **vii**
DAFTAR ISI ..... **viii**
DAFTAR TABEL ..... **xiii**
DAFTAR GAMBAR ..... **xvi**
DAFTAR LAMPIRAN ..... **xix**
BAB 1 PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.2.1 Rumusan Masalah Umum 3
1.2.2 Rumusan Masalah Khusus 3
1.3 Tujuan 3
1.3.1 Tujuan Umum 3
1.3.2 Tujuan Khusus 3
1.4 Manfaat 4
1.5 Batasan Masalah 4
1.6 Sistematika Pembahasan 4
BAB 2 LANDASAN KEPUSTAKAAN **6**
2.1 Kajian Pustaka 6
2.2 Profil SMA Suluh Jakarta 8
2.2.1 Visi 9
2.2.2 Misi 9
2.2.3 Struktur Organisasi 9
2.3 Sistem Informasi 10
2.4 *Software Development Life Cycle* (SDLC) 11
2.4.1 Model *Waterfall* 11
2.4.2 Model-View-Controller (MVC) 12

2.5 Proses Bisnis 14
2.6 *Business Process Model and Notation* (BPMN) 14
2.6.1 Flow Objects 14
2.6.2 Data 16
2.6.3 Connecting Objects 17
2.6.4 Swimlanes 18
2.6.5 Artifacts 18
2.7 Pemodelan *Use Case* 18
2.7.1 Identifikasi Tipe Pemangku Kepentingan 19
2.7.2 Analisis Masalah 19
2.7.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna 20
2.7.4 Identifikasi Fitur 20
2.7.5 Identifikasi Persyaratan Fungsional dan NonFungsional 21
2.7.6 Pemodelan *Use Case* 21
2.8 *Unified Modeling Language* (UML) 22
2.8.1 Use Case Diagram 22
2.8.2 Class Diagram 23
2.8.3 Sequence Diagram 25
2.9 *Physical Data Model* 26
2.10 Pengujian Perangkat Lunak 27
2.10.2 Pengujian Validasi 27
2.10.3 Pengujian Kompatibilitas 28
BAB 3 METODOLOGI 29
3.1 Studi Literatur 29
3.2 Analisis Persyaratan Sistem 30
3.3 Perancangan Sistem 31
3.4 Implementasi Sistem 31
3.5 Pengujian Sistem 31
3.6 Penarikan Kesimpulan dan Saran 31
BAB 4 ANALISIS PERSYARATAN 32
4.1 Pemodelan Proses Bisnis 32

4.1.1 Proses Bisnis *As-Is* 32
4.1.2 Analisis Masalah 34
4.1.3 Proses Bisnis *To-Be* 36
4.2 Analisis Persyaratan 42
4.2.1 Identifikasi Tipe Pemangku Kepentingan 42
4.2.2 Identifikasi Kebutuhan Pemangku Kepentingan dan pengguna 42
4.2.3 Identifikasi Penguna 46
4.2.4 Identifikasi Fitur 47
4.2.5 Persyaratan Fungsional 48
4.2.6 Persyaratan Nonfungsional 50
4.3 Pemodelan *Use Case* 51
4.3.1 Use Case Diagram 51
4.3.2 Deskripsi Aktor 54
4.3.3 Spesifikasi *Use case* 55
BAB 5 PERANCANGAN 74
5.1 Perancangan Arsitektur Sistem 74
5.2 Pemodelan Objek 74
5.2.1 Diagram Kelas Analisis 75
5.2.2 Diagram Kelas Perancangan 76
5.3 Pemodelan Interaksi Antar Objek 80
5.3.1 Memonitor Kehadiran Siswa Secara Menyeluruh 80
5.3.2 Mendata Kehadiran Siswa 81
5.3.3 Mengelola Data Siswa 83
5.3.4 Memonitor Rekap Data Kehadiran 87
5.3.5 Mengelola Data *Master* 88
5.4 Perancangan Basis Data 91
5.5 Perancangan Algoritme 92
5.5.1 Mendata Kehadiran Siswa 92
5.5.2 Mengelola Data Siswa 93
5.5.3 Memonitor Rekap Data Kehadiran 93
5.6 Perancangan Antarmuka Pengguna 94

5.6.1 Antarmuka *Login* 94
5.6.2 Antarmuka Kehadiran Siswa Secara Menyeluruh 95
5.6.3 Antarmuka Pendataan Kehadiran Siswa 95
5.6.4 Antarmuka Mengelola Data kelas 96
5.6.5 Antarmuka Mengelola Data Siswa 96
5.6.6 Antarmuka Rekap Data Kehadiran 97
BAB 6 IMPLEMENTASI 99
6.1 Spesifikasi Lingkungan Implementasi 99
6.2 Struktur Artefak Sistem 100
6.3 Implementasi Algoritme 101
6.3.1 Implementasi Algoritme Mendata Kehadiran Siswa 101
6.3.2 Implementasi Algoritme Mengelola Data Siswa 101
6.3.3 Implementasi Algoritme Memonitor Rekap Data Kehadiran 102
6.4 Implementasi Antarmuka Pengguna 102
6.4.1 Implementasi Antarmuka *Login* 103
6.4.2 Implementasi Antarmuka Kehadiran Siswa Secara Menyeluruh 103
6.4.3 Implementasi Antarmuka Pendataan Kehadiran Siswa 104
6.4.4 Implementasi Antarmuka Mengelola Data Siswa 104
6.4.5 Implementasi Antarmuka Mengelola Data Kelas 105
6.4.6 Implementasi Antarmuka Rekap Data Kehadiran 105
BAB 7 PENGUJIAN 107
7.1 Pengujian Validasi 107
7.1.1 Pengujian *Black-Box Login* 107
7.1.2 Pengujian *Black-Box* Mendata Kehadiran Siswa 108
7.1.3 Pengujian *Black-Box* Memonitor Rekap Data Kehadiran 111
7.1.4 Pengujian *Black-Box* Mengelola Data Siswa 114
7.1.5 Pengujian *Black-Box* Memonitor Kehadiran Siswa Secara Menyeluruh 117
7.2 Pengujian Kompatibilitas 119
BAB 8 PENUTUP 121
8.1 Kesimpulan 121

8.2 Saran 122
DAFTAR REFERENSI 123
LAMPIRAN A HASIL WAWANCARA 125
LAMPIRAN B HASIL PENGUJIAN VALIDASI 128

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR TABEL

Tabel 2.1 Notasi-Notasi dari *Events* 15
Tabel 2.2 Notasi-Notasi dari *Activities* 15
Tabel 2.3 Notasi-Notasi dari *Gateways* 15
Tabel 2.4 Notasi-Notasi dari *Data* 17
Tabel 2.5 Notasi-Notasi dari *Connecting Objects* 17
Tabel 2.6 Notasi-Notasi dari *Swimlanes* 18
Tabel 2.7 Notasi-Notasi dari *Artifacts* 18
Tabel 2.8 Kerangka Penulisan untuk Analisis Masalah 19
Tabel 2.9 Kerangka Penulisan untuk Identifikasi Fitur 20
Tabel 2.10 Kerangka Penulisan untuk Spesifikasi *Use Case* 21
Tabel 2.11 Notasi-Notasi dari *Use Case Diagram* 23
Tabel 2.12 Notasi-Notasi dari *Class Diagram* 24
Tabel 2.13 Notasi-Notasi dari *Sequence Diagram* 26
Tabel 4.1 Analisis Permasalahan 1 35
Tabel 4.2 Analisis Permasalahan 2 35
Tabel 4.3 Perubahan Aktivitas Proses Bisnis *As-Is* Mendata Kehadiran Siswa Ke Proses Bisnis *To-Be* Mendata Kehadiran Siswa dan Merevisi Data Kehadiran Siswa 37
Tabel 4.4 Perubahan Aktivitas Proses Bisnis *As-Is* Mengelola Data Siswa Ke Proses Bisnis *To-Be* Mendata Siswa dan Memonitor Data Siswa 40
Tabel 4.5 Identifikasi Tipe Pemangku Kepentingan 42
Tabel 4.6 Hasil Identifikasi Kebutuhan Pengguna 43
Tabel 4.7 Hasil Identifikasi Pengguna 46
Tabel 4.8 Hasil Identifikasi Fitur 48
Tabel 4.9 Hubungan Fitur dengan Kebutuhan Pengguna 48
Tabel 4.10 Persyaratan Fungsional 49
Tabel 4.11 Persyaratan Nonfungsional 51
Tabel 4.12 Hubungan *Use Case* dengan Aktivitas pada Proses Bisnis *To-Be* 52
Tabel 4.13 Hubungan *Use case* dengan Pemangku Kepentingan 53
Tabel 4.14 Hubungan antara *Use Case* dengan Fitur 53
Tabel 4.15 Deskripsi Aktor 54

Tabel 4.16 Spesifikasi *Use Case Login* 55
Tabel 4.17 Spesifikasi *Use Case Logout* 56
Tabel 4.18 Spesifikasi Use Case Memonitor Kehadiran Siswa Secara Menyeluruh 57
Tabel 4.19 Spesifikasi Use Case Mendata Kehadiran Siswa 58
Tabel 4.20 Spesifikasi Use Case Merevisi Data Kehadiran Siswa 62
Tabel 4.21 Spesifikasi *Use Case* Mengelola Data Siswa 63
Tabel 4.22 Spesifikasi *Use Case* Memonitor Data Siswa 68
Tabel 4.23 Spesifikasi *Use Case* Memonitor Rekap Data Kehadiran 69
Tabel 4.24 Spesifikasi *Use Case* Mengelola Data *Master* 71
Tabel 5.1 *Pseudocode* Mendata Kehadiran Siswa 92
Tabel 5.2 *Pseudocode* Mengelola Data Siswa 93
Tabel 5.3 *Pseudocode* Memonitor Rekap Data Kehadiran 93
Tabel 6.1 Spesifikasi Perangkat Keras 99
Tabel 6.2 Spesifikasi Kebutuhan Perangkat Lunak 99
Tabel 6.3 Spesifikasi Lingkungan Pengembangan 100
Tabel 6.4 Implementasi Algoritme Mendata Kehadiran Siswa 101
Tabel 6.5 Implementasi Algoritme Mengelola Data Siswa 102
Tabel 6.6 Implementasi Algoritme Memonitor Rekap Data Kehadiran 102
Tabel 7.1 Matriks Skenario Pengujian *Black-Box Login* 107
Tabel 7.2 Rencana Pengujian dan Kasus Uji *Black-Box Login* 108
Tabel 7.3 Matriks Skenario Pengujian *Black-Box* Mendata Kehadiran Siswa 108
Tabel 7.4 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menyimpan Data Kehadiran 109
Tabel 7.5 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menyimpan Data Kehadiran Alternatif A5 (Menangani Kegagalan Penyimpan Data) 110
Tabel 7.6 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Mengubah Data Kehadiran 111
Tabel 7.7 Matriks Skenario Pengujian *Black-Box* Memonitor Rekap Data Kehadiran 112
Tabel 7.8 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Melihat Rekap Data Kehadiran 112
Tabel 7.9 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Melihat Rekap Data Kehadiran dengan Alternatif A1 113

Tabel 7.10 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Mengunduh Rekap Data Kehadiran 114

Tabel 7.11 Matriks Skenario Pengujian *Black-Box* Mengelola Data Siswa 115

Tabel 7.12 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menambah Data Banyak Siswa 115

Tabel 7.13 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Mengubah Data Siswa 116

Tabel 7.14 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menghapus Data Siswa 117

Tabel 7.15 Matriks Skenario Pengujian *Black-Box* Memonitor Kehadiran Siswa Secara Menyeluruh 118

Tabel 7.16 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menampilkan Data Kehadiran Siswa Secara Menyeluruh 118

Tabel 7.17 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menampilkan Data Kehadiran Siswa Secara Menyeluruh dengan Alternatif A1 119

Tabel 7.18 Jenis Peramban Pada Pengujian Kompatibilitas 120

DAFTAR GAMBAR

Gambar 2.1 Bagan Organisasi SMA Suluh Jakarta 10
Gambar 2.2 Tahapan Fase pada Model *Waterfall* 11
Gambar 2.3 Pola *Model-View-Controller* (MVC) 13
Gambar 2.4 Contoh dari *Use Case Diagram* 22
Gambar 2.5 Contoh Relasi antar *Class* 24
Gambar 2.6 Contoh dari *Sequence Diagram* 25
Gambar 2.7 Contoh dari *Physical Data Model* 27
Gambar 2.8 Contoh dari Hasil Pengujian Kompatibilitas dengan Aplikasi SortSite 28
Gambar 3.1 Diagram Alir Pengerjaan Penelitian 29
Gambar 4.1 Proses Binis *As-Is* dari Mendata Kehadiran Siswa 33
Gambar 4.2 Proses Binis *As-Is* dari Merekap Data Kehadiran 33
Gambar 4.3 Proses Bisnis *As-Is* dari Mengelola Data Siswa 34
Gambar 4.4 Proses Bisnis *To-Be* dari Mendata Kehadiran Siswa 36
Gambar 4.5 Proses Bisnis *To-Be* dari Merevisi Data Kehadiran Siswa 37
Gambar 4.6 Proses Bisnis *To-Be* dari Mendata Siswa 40
Gambar 4.7 Proses Bisnis *To-Be* dari Memonitor Data Siswa 40
Gambar 4.8 Kode Identifikasi Kebutuhan Pengguna 42
Gambar 4.9 Kode Identifikasi dari Fitur 47
Gambar 4.10 Kode Persyaratan Fungsional 49
Gambar 4.11 Kode Persyaratan Nonfungsional 50
Gambar 4.12 *Use Case Diagram* Sistem Informasi Kehadiran Siswa 52
Gambar 5.1 Arsitektur Sistem 74
Gambar 5.2 Diagram Kelas Analisis 75
Gambar 5.3 Atribut pada Diagram Kelas Analisis 76
Gambar 5.4 Diagram Kelas Perancangan *Controller* 76
Gambar 5.5 Diagram Kelas Perancangan *Model* 77
Gambar 5.6 Diagram Kelas Perancangan S*tereotype View* 78
Gambar 5.7 Diagram Kelas Perancangan Relasi *Model* dan *Controller* 79
Gambar 5.8 Diagram Kelas Perancangan Relasi *Controller* dan *View* 79

Gambar 5.9 *Sequence Diagram* Menampilkan Data Kehadiran Siswa Secara Menyeluruh 80
Gambar 5.10 *Sequence Diagram* Menyimpan Data Kehadiran 81
Gambar 5.11 *Sequence Diagram* Menampilkan Data Kehadiran 82
Gambar 5.12 *Sequence Diagram* Mengubah Data Kehadiran 82
Gambar 5.13 *Sequence Diagram* Menampilkan Data Siswa 83
Gambar 5.14 *Sequence Diagram* Menambah Data Satu Siswa 84
Gambar 5.15 *Sequence Diagram* Menghapus Data Siswa 84
Gambar 5.16 *Sequence Diagram* Mengubah Data Siswa 85
Gambar 5.17 *Sequence Diagram* Menambah Data Banyak Siswa 86
Gambar 5.18 *Sequence Diagram* Mengunduh Data Siswa 86
Gambar 5.19 *Sequence Diagram* Menampilkan Detail Data Siswa 87
Gambar 5.20 *Sequence Diagram* Menampilkan Rekap Data Kehadiran 88
Gambar 5.21 *Sequence Diagram* Mengunduh Rekap Data Kehadiran 88
Gambar 5.22 *Sequence Diagram* Menampilkan Data Kelas 89
Gambar 5.23 *Sequence Diagram* Menambah Data Kelas 90
Gambar 5.24 *Sequence Diagram* Menghapus Data Kelas 90
Gambar 5.25 *Sequence Diagram* Mengubah Data Kelas 91
Gambar 5.26 *Physical Data Model* Sistem Informasi Kehadiran Siswa 91
Gambar 5.27 Perancangan Antamuka Login 94
Gambar 5.28 Perancangan Antamuka Kehadiran Siswa Secara Menyeluruh 95
Gambar 5.29 Perancangan Antamuka Pendataan Kehadiran Siswa 95
Gambar 5.30 Perancangan Antamuka Mengelola Data Kelas 96
Gambar 5.31 Perancangan Antamuka Mengelola Data Siswa 97
Gambar 5.32 Perancangan Antamuka Rekap Data Kehadiran 97
Gambar 6.1 Struktur Artefak Sistem Informasi Kehadiran Siswa 100
Gambar 6.2 Implementasi Antarmuka Login 103
Gambar 6.3 Implementasi Antarmuka Kehadiran Siswa Secara Menyeluruh 103
Gambar 6.4 Implementasi Antarmuka Pendataan Kehadiran Siswa 104
Gambar 6.5 Implementasi Antarmuka Mengelola Data Siswa 105
Gambar 6.6 Implementasi Antarmuka Mengelola Data Kelas 105
Gambar 6.7 Implementasi Antarmuka Rekap Data Kehadiran 106

Gambar 7.1 Kode Pengujian *Black-Box* dengan Teknik Validasi 107
Gambar 7.2 Hasil Pengujian Kompatibilitas 120

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA 125
A.1 Hasil Wawancara dengan Guru BK 125
A.2 Hasil Wawancara dengan Guru Piket 127
LAMPIRAN B HASIL PENGUJIAN VALIDASI 128
B.1 Hasil Pengujian Validasi Menyimpan Data Kehadiran 128
B.2 Hasil Pengujian Validasi Menyimpan Data Kehadiran Alternatif A5 128
B.3 Hasil Pengujian Validasi Mengubah Data Kehadiran 129

BAB 1 PENDAHULUAN

1.1 Latar Belakang

SMA Suluh Jakarta merupakan sekolah swasta yang berada di bawah naungan Yayasan Hortikultura dan berstatus sebagai Sekolah Kategori Mandiri (SKM) atau Sekolah Standar Nasional (SSN). Dalam struktur organisasi SMA Suluh Jakarta terdapat 6 bagian yang berada dibawah kepala sekolah yaitu wakil kepala sekolah bidang kurikulum, wakil kepala sekolah bidang kesiswaan, wakil kepala sekolah bidang sarana, guru, bendahara sekolah dan KTU. Setiap bagian memiliki perannya masing-masing dalam mencapai tujuan dari proses bisnis yang ingin dicapai. Pada bagian wakil kepala sekolah bidang kesiswaan terdapat dua subbagian didalamnya yaitu BK dan OSIS. BK memiliki kegiatan yang rutin dilakukan setiap satu minggu sekali yaitu melakukan perekapan data kehadiran siswa. Hasil dari perekapan yang dilakukan kemudian diberikan kepada kepala sekolah dan wali kelas. Kepala sekolah dan wali kelas kemudian akan memantau kehadiran siswa dari hasil perekapan data yang diberikan oleh BK.

Perekapan data kehadiran siswa menjadi salah satu aspek yang dipantau oleh pihak sekolah karena dari data tersebut dapat dilihat persentase kehadiran dari siswa. Hal ini berkaitan dengan peraturan SMA Suluh Jakarta yang terdapat pada bab 3 pasal 3 nomor 2 yang berbunyi "Siswa tidak naik kelas jika tidak hadir tanpa keterangan melebihi 15% dari jumlah hari efektif sekolah." Pemantauan kehadiran siswa juga dilakukan untuk mendisiplinkan siswa sesuai dengan tujuan dari SMA Suluh Jakarta yaitu peserta didik dapat menjadi manusia yang beriman, bertaqwa kepada Tuhan YME, dan berdisiplin tinggi.

Selain kepala sekolah dan wali kelas yang memantau kehadiran siswa, guru BK juga bertanggung jawab untuk memantau kehadiran siswa. Pada SMA Suluh Jakarta, BK juga memiliki peranan dalam bekerjasama dengan wali kelas untuk mengatasi persoalan peserta didik di kelasnya. Menurut Mulyadi (2016), dengan adanya bimbingan dan konseling di sekolah, individu yang dibantu dapat mampu menyesuaikan diri dengan lingkungannya, dapat mengatasi masalah atau hambatan yang dihadapi, dapat mengembangkan potensi di dalam dirinya, dan dapat merencanakan kegiatan untuk menyelesaikan studi serta mengembangkan karir dan kehidupannya dimasa mendatang. Ketika terdapat siswa yang tidak hadir secara terus-menerus atau tidak hadir tanpa keterangan dengan jumlah yang cukup banyak maka guru BK akan berperan dalam melakukan bimbingan dan konseling dengan siswa tersebut untuk mengetahui penyebab ketidakhadirannya. Dan diharapkan dengan adanya tatap muka dan komunikasi yang dilakukan oleh guru BK dengan siswa tersebut maka guru BK dapat mengetahui alasan mengapa siswa tersebut tidak hadir dan menemukan solusi dari permasalahan yang dihadapi. Data terkait kehadiran siswa juga terkadang diminta oleh wali kelas ketika ada orang tua siswa yang datang ke sekolah dan bertanya mengenai kehadiran anaknya di sekolah.

Dari hasil wawancara yang penulis lakukan dengan ibu Dina Apriliyati selaku guru BK SMA Suluh Jakarta, perekapan data kehadiran siswa diperoleh dari pengecekkan dan pencatatan kehadiran siswa yang dilakukan setiap paginya oleh guru yang berada di kelas. Setelah satu minggu BK akan meminta presensi siswa dari setiap kelas untuk dimasukkan ke dalam Microsoft Excel. Data yang telah direkap mingguan kemudian akan diberikan kepada wali kelas, sedangkan untuk kepala sekolah data diberikan setelah dilakukan perekapan untuk satu bulan. Dari pengelolaan perekapan data kehadiran siswa yang dilakukan tersebut masih memiliki beberapa permasalahan karena belum adanya pemanfaatan teknologi secara maksimal. Permasalahan yang dihadapi adalah data kehadiran siswa yang saat ini ada tidak terhubung dengan sebuah sistem untuk menampung datanya dan perlu adanya perekapan data kembali setiap minggunya. Perekapan data kehadiran yang perlu dimasukkan ke Microsoft Excel setiap minggunya sebanyak 3.440 data, sehingga perekapan data kehadiran menjadi rawan terhadap kesalahan-kesalahan. Data kehadiran siswa yang saat ini ada juga belum terintegrasi satu sama lain, sehingga ketika siswa datang terlambat maka data kehadiran siswa tersebut menjadi kosong dan kehadiran siswa tersebut perlu dikonfirmasi ke guru piket. Sedangkan untuk surat siswa yang sakit atau izin yang diberikan melalui guru piket akan dituliskan pada buku piket dan buku presensi kelas. Serta ketika data terkait kehadiran diperlukan oleh wali kelas atau kepala sekolah secara tiba-tiba, guru BK akan mengalami kesulitan untuk menampilkan datanya secara cepat karena memerlukan pencarian data secara manual terlebih dahulu atau justru data yang diperlukan belum tersedia karena belum dilakukan perekapan.

Dengan adanya permasalahan tersebut, SMA Suluh Jakarta memerlukan sebuah teknologi yang dapat membantu dalam mengelola data kehadiran berupa sistem yang terkomputerisasi yaitu sistem informasi. Sistem Informasi yang dikembangkan dapat membantu pihak sekolah dalam mengelola data kehadiran siswanya, yaitu data kehadiran yang ada dapat dimasukkan dalam satu penyimpanan yang saling terintegrasi datanya sehingga dapat dilihat oleh pengguna sesuai dengan hak akses yang dimiliki, data dapat terus terbarui dan data yang dimasukkan saat pendataan secara otomatis akan terekap sehingga proses perekapan tidak perlu lagi dilakukan.

Pengembangan sistem informasi terkait dengan manajemen sekolah juga pernah dilakukan oleh Pargiyani et al. (2019). Penelitian ini bertujuan untuk mengembangkan sistem informasi yang dapat meningkatkan pelayanan sekolah dalam mengelola nilai hasil belajar siswa dan pembayaran SPP, serta memberikan kemudahan kepada orang tua siswa untuk berinteraksi dengan wali kelas.

Berdasarkan uraian yang telah dijelaskan diatas, penulis tertarik untuk melakukan pengembangan Sistem Informasi Kehadiran Siswa pada SMA Suluh Jakarta. Salah satu model yang dapat digunakan dalam pengembangan sistem yaitu model *waterfall*. Menurut Alshamrani & Bahattab (2015) dengan penggunaan model *waterfall* dalam pengembangan, tahapan yang dilakukan tidak akan saling tumpang tindih karena jika tahapan pada tahap sebelumnya belum

terselesaikan maka tahap selanjutnya tidak dapat dilakukan. Kebutuhan yang diperlukan oleh pemangku kepentingan pada tahap analisis persyaratan dapat teridentifikasi dengan jelas sebelum dapat melakukan perancangan. Kemudian perlu dipastikan tidak adanya kesalahan pada perancangan untuk dapat dilakukan pengembangan. Metode yang digunakan untuk mengembangkan Sistem Informasi Kehadiran Siswa menggunakan model *waterfall*. Agar Sistem Informasi Kehadiran Siswa dapat diakses kapanpun diberbagai tempat oleh pihak-pihak yang memerlukan informasi tersebut, sistem yang dikembangkan berbasis web.

1.2 Rumusan Masalah

Berdasarkan pada latar belakang yang telah diuraikan, terdapat rumusan masalah umum dan rumusan masalah khusus yang perlu dijawab. Berikut ini adalah rumusan masalah dari penelitian yang akan dilakukan yaitu sebagai berikut:

1.2.1 Rumusan Masalah Umum

Dapatkah dikembangkan sebuah Sistem Informasi Kehadiran Siswa yang dapat mengelola data kehadiran siswa di SMA Suluh Jakarta?

1.2.2 Rumusan Masalah Khusus

1. Bagaimanakah hasil analisis persyaratan Sistem Informasi Kehadiran Siswa?
2. Bagaimanakah hasil perancangan Sistem Informasi Kehadiran Siswa?
3. Bagaimanakah hasil implementasi Sistem Informasi Kehadiran Siswa berdasarkan hasil analisis persyaratan dan perancangan sistem?
4. Bagaimanakah hasil pengujian Sistem Informasi Kehadiran Siswa ?

1.3 Tujuan

Tujuan yang diharapkan dari pelaksanaan penelitian yang dilakukan adalah sebagai berikut:

1.3.1 Tujuan Umum

Melakukan pengembangan Sistem Informasi Kehadiran Siswa yang dapat mengelola data kehadiran Siswa pada SMA Suluh Jakarta.

1.3.2 Tujuan Khusus

1. Menspesifikasi persyaratan Sistem Informasi Kehadiran Siswa.
2. Merancang Sistem Informasi Kehadiran Siswa.
3. Mengimplementasikan Sistem Informasi Kehadiran Siswa berdasarkan hasil identifikasi persyaratan dan perancangan sistem.
4. Menguji Sistem Informasi Kehadiran Siswa.

1.4 Manfaat

Manfaat dari penelitian yang dilakukan adalah Sistem Informasi Kehadiran Siswa yang dikembangkan dapat membantu pihak-pihak yang ada pada SMA Suluh Jakarta dalam melakukan pengelolaan data kehadiran siswanya.

1.5 Batasan Masalah

Berdasarkan latar belakang dan tujuan yang ada, maka terdapat batasan masalah untuk penelitian ini sebagai berikut :

1. Penelitian berfokus pada pendataan dan perekapan data kehadiran siswa di SMA Suluh Jakarta.
2. Perancangan *database* yang digunakan untuk menyimpan data secara keseluruhan menggunakan MySQL versi 4.8.3
3. Pengembangan sistem informasi berbasis web menggunakan pemrograman HTML dan PHP versi 5.6.28 serta menggunakan kerangka kerja Codeigniter.
4. Pengguna dari sistem informasi ini nantinya adalah kepala sekolah, guru, dan tata usaha.

1.6 Sistematika Pembahasan

Dari penelitian yang dilakukan terdapat sistematika pembahasan yang terdiri dari 8 bab. Berikut ini adalah rincian dari setiap bab nya :

BAB 1 : PENDAHULUAN

Bab pendahuluan menerangkan mengenai latar belakang dilakukannya penelitian terkait pengembangan Sistem Informasi Kehadiran Siswa, rumusan masalah berdasarkan latar belakang dari penilitian yang dilakukan, tujuan dan manfaat dari adanya penelitian pengembangan sistem informasi kehadiran siswa, batasan masalah dari penelitian, dan ringkasan penjelasan yang dibahas pada tiap bab berupa sistematika pembahasan penelitian.

BAB 2 : LANDASAN KEPUSTAKAAN

Bab landasan keputusan menerangkan mengenai konsep teori dari literatur ilmiah yang digunakan dalam mengembangkan Sistem Informasi Kehadiran Siswa di SMA Suluh Jakarta.

BAB 3 : METODOLOGI PENELITIAN

Bab metodologi penelitian menerangkan mengenai metode atau teknik untuk mengumpulkan data yang digunakan dalam penelitian, serta berisi alur pengerjaan dari penelitian yang dilakukan.

BAB 4 : ANALISIS PERSYARATAN

Bab analisis pesyaratan menerangkan mengenai kondisi yang saat ini ada pada SMA Suluh Jakarta berupa proses bisnis, permasalahan, kebutuhan, dan

pemangku kepentingan yang terlibat. Kemudian dari kondisi yang ada, ditawarkan solusi untuk mengatasi permasalahan yang ada beserta dengan kebutuhan dari pemangku kepentingan. Dari analisis persyaratan ini kemudian digunakan untuk melakukan perancangan Sistem Informasi Kehadiran Siswa.

BAB 5 : PERANCANGAN

Bab perancangan menerangkan mengenai hasil dari analisis persyaratan Sistem Informasi Kehadiran Siswa yang telah dilakukan kemudian dibuat perancangannya dengan melakukan pemodelan dengan menggunakan UML (*Unified Modeling Language*).

BAB 6 : IMPLEMENTASI

Bab implementasi menerangkan mengenai hasil implementasi dari perancangan Sistem Informasi Kehadiran Siswa yang telah dilakukan dengan menggunakan kerangka kerja Codeigniter dan MySQL sebagai penyimpanan datanya.

BAB 7 : PENGUJIAN

Bab pengujian menerangkan mengenai Sistem Informasi Kehadiran Siswa yang telah dikembangkan diuji dengan menggunakan pengujian validasi dan Kompatibilitas.

BAB 8 : PENUTUP

Bab penutup menerangkan mengenai ringkasan dari penelitian yang telah dilakukan beserta saran untuk penelitian selanjutnya terkait dengan pengembangan lebih lanjut dari sistem yang telah dikembangkan.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian Pustaka merupakan penelitian-penelitian sebelumnya yang telah dilakukan dan bersumber dari jurnal. Penelitian yang pertama merupakan penelitian yang berjudul "Pengembangan Sistem Informasi Akademik Pada Sekolah Dasar Islam Terpadu Insantama (SDIT) Malang Berbasis Web." Penelitian yang dilakukan oleh Pargiyani et al. (2019) membahas mengenai pengembangan sistem informasi yang digunakan untuk membantu pemangku kepentingan dalam mengatasi permasalahan pengelolaan penilaian hasil belajar siswa dan pembayaran SPP. Permasalahan yang dihadapi oleh pemangku kepentingan yaitu pengelolaan data yang belum dilakukan secara efisien sehingga berdampak pada pemberian rapor yang terlambat, pengarsipan nilai siswa yang sulit dilakukan karena belum adanya sentralisasi data, data yang telah disimpan dalam bentuk Excel dapat terkena virus, terhapus atau tergantikan datanya, serta kesalahan saat pencatatan atau kesalahan perhitungan saat melakukan pembayaran SPP. Dengan adanya permasalahan tersebut dikembangkan Sistem Informasi Akademik yang dapat membantu mengelola hasil belajar siswa, pembaharuan informasi pembayaran dan memudahkan komunikasi orang tua siswa dengan wali kelas. Pengembangan sistem dilakukan dengan menggunakan model *waterfall,* menerapkan pendekatan berorientasi objek pada perancangan dengan melakukan pemodelan Unified Modelling Language (UML) dan kerangka kerja yang digunakan adalah Codeigniter. Pengujian yang dilakukan dalam penelitian ini yaitu pengujian unit dengan teknik *basis path*, pengujian validasi yang menghasilkan hasil valid dan pengujian kompatibilitas dengan hasil aplikasi dapat berjalan pada peramban Firefox dan Chrome. Keterkaitan dengan penelitian yang penulis lakukan yaitu permasalahan yang sama terkait dengan pengelolaan manajemen sekolah. Penelitian yang penulis lakukan terkait kehadiran siswa sedangkan penelitian Pargiyani et al. (2019) terkait dengan pengelolaan hasil belajar siswa dan pembayaran SPP.

Penelitian yang kedua merupakan penelitian yang berjudul "Pengembangan Sistem Informasi Pencatatan dan Pemantauan Distribusi Surat Masuk dan Surat Keluar dengan Pendekatan Berorientasi Objek (Studi Kasus: Datasemen A Pelopor Satbrimob Polda Jatim)." Penelitian yang dilakukan oleh Aryasworo et al. (2018) membahas mengenai pengembangan sistem untuk membantu dalam pengelolaan surat yang masih dilakukan secara manual sehingga menimbulkan masalah, salah satunya adalah kesulitan dalam melakukan pemantauan surat yang telah didistribusikan ke bagian. Oleh karena itu dikembangkan sistem dengan menggunakan model *waterfall*, menerapkan pendekatan berorientasi objek pada perancangan sistem dengan melakukan pemodelan *Unified Modelling Language* (UML), pengimplementasian yang merupakan hasil terjemahan dari perancangan dengan menggunakan kerangka kerja Codeigniter dan bahasa pemrograman PHP serta untuk penyimpanan datanya menggunkan MySQL. Untuk pengujian mengunakan pengujian validasi

dan kompatibilitas. Dilakukan pengujian validasi untuk menguji sistem yang telah dikembangkan berdasarkan persyaratan fungsional (Pressman, 2010). Pengujian kompatibilitas dilakukan untuk memastikan sistem yang dibangun dapat berjalan dengan baik pada lingkungan yang heterogen (Yoon, et al., 2008). Penelitian yang dilakukan oleh Aryasworo et al. (2018) menjadi referensi penulis dalam pengujian sistem menggunakan pengujian validasi dan kompatibilitas.

Penelitian yang ketiga merupakan penelitian yang berjudul "*A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model*." Penelitian yang dilakukan oleh Alshamrani & Bahattab (2015) ini membandingkan 3 model SDLC yang digunakan untuk mengembangkan sistem yaitu model *spiral*, *waterfall*, dan *incremental/iterative*. Perbandingan dilakukan dengan melihat kelebihan, kekurangan, dan waktu penggunaan. Dari penelitian ini dapat disimpulkan bahwa dalam melakukan pengembangan sistem terdapat banyak model yang dapat digunakan. Model *spiral*, *waterfall*, dan *incremental/iterative* memiliki kelebihan dan kekurangannya masing-masing, sehingga pemilihan model yang digunakan untuk pengembangan tergantung dengan kebutuhan dari klien tersebut. Pada model *waterfall* tahapan-tahapan yang dilakukan secara berurutan tanpa adanya tahapan yang tumpang tindih dimulai dari tahap analisis persyaratan, perancangan, pengkodean, pengujian dan pemeliharaan. Kontrol kualitas pada model *waterfall* menjadi hal yang diperhatikan karena adanya perencanaan dan dokumentasi yang intensif. Kelebihan dari model *waterfall* adalah mudah untuk diimplementasikan dan digunakan, setiap tahapan dilakukan secara bertahap, model ini telah dikenal dan banyak digunakan, dan meminimalkan adanya *overhead* perencanaan. Sedangkan untuk kekurangannya, model *waterfall* ini tidak fleksibel terhadap perubahan-perubahan. Penggunaan model *waterfall* cocok digunakan jika persyaratan telah diidentifikasi dengan jelas dan ingin membuat versi terbaru dari produk yang telah ada. Dari penelitian yang dilakukan oleh Alshamrani & Bahattab (2015) menjadi referensi untuk mengembangkan sistem dengan menggunakan model *waterfall*. Karena semua kebutuhan ditetapkan dengan jelas diawal sebelum dapat dilakukan perancangan dan pengembangan sistem.

Penelitian yang keempat merupakan penelitian yang berjudul "Pengembangan Sistem Informasi Penggajian Pegawai dan Dokumentasi Pekerjaan Berbasis Web (Studi Kasus: PT. Yuga Elektro Tuban)." Penelitian ini dilakukan untuk membantu perusahaan yuga elektro tuban dalam meningkatkan efisiensi pembuatan laporan keuangan dan pembuatan dokumentasi pekerjaan dengan menggunakan alat bantu berupa sistem informasi. Dalam penelitian ini sistem informasi harus dapat diakses dari banyak tempat, data terkumpul dalam satu tempat, dapat digunakan tanpa tergantung dengan komputer lokal dan dapat mendukung proses bisnis dari perusahaan. Oleh karena itu, sistem informasi yang dikembangkan dirasa cocok bila berbasis web. Pendekatan berorientasi objek digunakan dalam pengembangan sistem untuk memudahkan dalam menghadapi perubahan pada sistem daripada dengan pendekatan fungsional, serta jika terjadi perubahan pada suatu objek maka tidak akan berdampak pada objek yang lain. Dengan menggunakan pendekatan berorientasi objek dalam pengembangan

sistem dilakukan proses pemodelan sistem dengan alat bantu berupa notasi grafis yang didasarkan pada penggunaan *unified modeling language* (UML). Notasi grafis digunakan untuk menggambarkan sistem yang dikembangkan (Sommerville, 2011). Untuk implementasi sistem menggunakan PHP dengan kerangka kerja Laravel dan untuk penyimpanan datanya menggunakan MySQL. Kemudian sistem diuji dengan menggunakan *basis path* dengan berdasarkan pada skenario dari hasil perhitungan *cyclomatic complexity* terhadap *flow graph*, pengujian validasi berdasarkan *use case* yang telah dibuat dan diuji oleh aktor, serta pengujian kompatibilitas dilakukan pada beberapa peramban web yang berbeda. Penelitian yang dilakukan oleh Istanto et al. (2018) menjadi referensi penulis dalam pengembangan sistem dengan menggunakan pendekatan berorientasi objek dan berbasis web.

Penelitian yang kelima merupakan penelitian yang berjudul "Analisis Framework Codeigniter dan Laravel Menggunakan Web Server Apache." Penelitian yang dilakukan oleh Erinton et al. (2017) ini membahas mengenai pengkajian kerangka kerja Codeignter dan Laravel untuk digunakan dalam pengembangan website. Penggunaan kerangka kerja dalam pembuatan website dapat membantu pengembang sehingga tidak perlu membangun website dari awal dan dapat mempermudah pengembangan sistem dalam memperbarui fitur yang telah ada. Saat ini, terdapat banyak kerangka kerja yang dapat digunakan dalam membangun atau mengembangkan sistem sehingga diperlukan analisis untuk dapat membantu pengembang memilih kerangka kerja berdasarkan performanya. Pada penelitian ini dilakukan pengkajian kerangka kerja Codeigniter dan Laravel dengan melakukan analisis performansi pada *streaming service* dan FTP *service* (*download*) berdasarkan QoS (*Quality of Service*) yang di tetapkan ITU-T dengan menggunakan *web server* apache. Pengujian yang dilakukan pada penelitian yaitu *load test* yang digunakan untuk mengetahui waktu yang dibutuhkan untuk menampilkan halaman atau *client* yang mengirim pesan ke *server* sampai mendapat jawaban, besarnya halaman yang harus dimuat dalam satuan KB, dan kecepatan memuat halaman dalam satuan Kbit/s. Serta dilakukan juga pegujian s*tress test* utuk mensimulasikan variasi dari pengguna saat mengakses aplikasi secara bersamaan. Dari pengujian yang telah dilakukan kerangka kerja Codeigniter memiliki kelebihan dalam hal performansi dibandingan dengan kerangka kerja Laravel. Kerangka kerja Laravel memiliki banyak *library* sehingga ukuran halaman yang harus dimuat lebih besar dan hal ini juga akan berpengaruh pada waktu untuk menampilkan halaman yang lebih lama. Dari penelitian yang telah dilakukan oleh Erinton et al. (2017) menjadi referensi untuk melakukan implementasi hasil perancangan yang telah dibuat ke dalam bentuk kode program dengan menggunakan kerangka kerja Codeigniter. Kerangka kerja Codeigniter dalam hal performansi memiliki keunggulan.

2.2 Profil SMA Suluh Jakarta

Yayasan Kesejahteraan Keluarga Hortikultura pada tanggal 2 Februari 1975 mengeluarkan surat keputusan no. 10/SMA/1975. Keputusan yang dikeluarkan tersebut mengenai pembukaan Sekolah Menengah Atas bernama SMA Suluh

Jakarta pada tanggal 1 Januari 1976. Pembukaan sekolah dilakukan karena banyaknya putra-putri dari pegawai pertanian dan masyarakat yang kesulitan untuk melanjutkan sekolahnya ke jenjang SMA yang saat itu jumlahnya masih kurang memadai. Dana untuk pelaksanaan pendidikan dibebankan kepada peserta didik berupa pembayaran SPP dan uang sumbangan yang tidak mengikat.

Pada tahun 1976 SMA Suluh Jakarta berstatus terdaftar dengan kepala sekolah yang memimpin saat itu Bapak Drs.Soemardi. Pada tahun 1982 digantikan oleh Bapak R.Moerbono dan pada tahun 1983 digantikan kembali oleh Bapak Drs.Soemardi. Setelah itu, SMA Suluh Jakarta pernah dikepalai oleh Drs. L.Darsini, Ir. Susilo. M. Si, Dra. Empong Heryani, dan Sudarsono, S.Pd. Sedangkan saat ini SMA Suluh Jakarta dikepalai oleh Bapak Saifudin S.Ag. Untuk status sekolah dari SMA Suluh Jakarta pada tahun 1980 diakui, kemudian tahun 1986 disamakan, tahun 1999 terakreditasi A, tahun 2009-2010 RSKM (Rintisan Sekolah Kategori Mandiri), dan tahun 2011-2018 SMA Suluh Jakarta berstatus SKM (Sekolah Kategori Mandiri) atau SSN (Sekolah Standar Nasional).

2.2.1 Visi

Terwujudnya peserta didik yang berprestasi dan berakhlak mulia yang dilandasi IMTAK dan IPTEK.

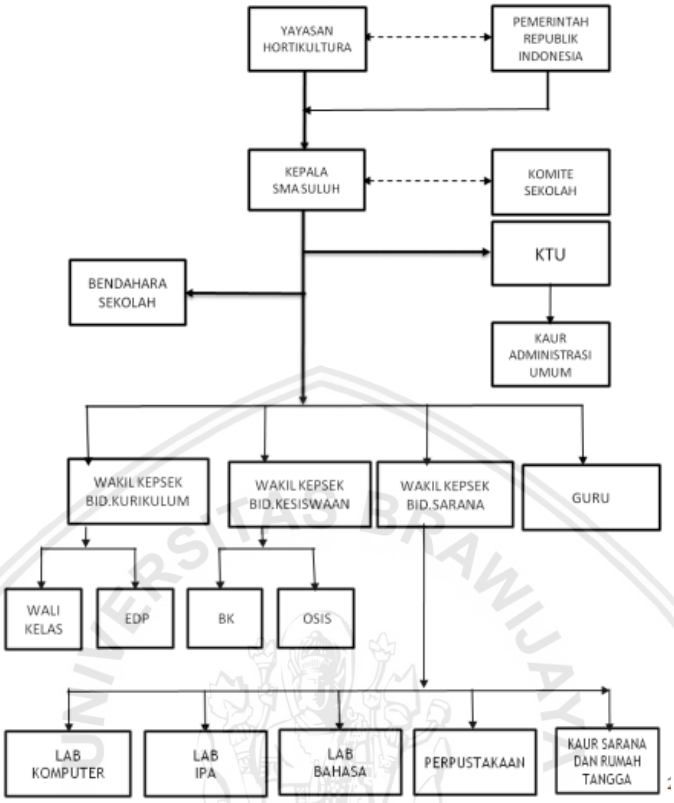
2.2.2 Misi

1. Meningkatkan Pembinaan akhlak dan berbudi pekerti luhur.
2. Mengembangkan minat, bakat dan kreativitas peserta didik agar tumbuh dan berkembang sesuai dengan potensi yang dimiliki.
3. Meningkatkan profesionalisme tenaga pendidik dan kependidikan dalam implementasi manajemen partisipasi untuk peningkatan mutu berbasis sekolah.
4. Meningkatkan pelayanan kegiatan belajar mengajar yang dapat memotivasi peserta didik agar kreatif, inisiatif, dan inovatif, baik dalam kegiatan intra maupun ekstrakurikuler.
5. Menjadikan SMA SULUH sebagai Sekolah Katagori Mandiri.
6. Menghasilkan kualitas lulusan yang mampu berkompetisi pada masa yang akan datang.

2.2.3 Struktur Organisasi

Berdasarkan struktur organisasi dari SMA Suluh Jakarta yang ditunjukkan oleh Gambar 2.1, SMA Suluh Jakarta merupakan sekolah yang berada dibawah naungan sebuah yayasan yaitu Yayasan Hortikultura dan dikepalai oleh seorang kepala sekolah dengan 4 bagian yang berada dibawahnya serta dibantu dengan adanya KTU dan bendahara sekolah. Terdapat 2 subbagian wakil kepala sekolah bidang kurikulum yaitu wali kelas dan EDP, 2 subbagian wakil kepala sekolah bidang kesiswaan yaitu BK dan OSIS, dan 5 subbagian wakil kepala sekolah bidang

sarana yaitu lab komputer, lab IPA, lab bahasa, perpustakaan serta KAUR sarana dan rumah tangga.



Gambar 2.1 Bagan Organisasi SMA Suluh Jakarta

2.3 Sistem Informasi

Sistem menurut O'Brien & Marakas (2011) merupakan komponen-komponen dengan batas yang jelas dan saling terkait bekerja sama untuk mencapai tujuan bersama dengan menerima *input* dan menghasilkan *output*. O'Brien & Marakas juga menjelaskan bahwa sistem informasi adalah mengolah sumber daya data yang diterima sebagai masukkan (*input*) menjadi keluaran (*output*) berupa produk informasi. Untuk melakukan *input*, pemrosesan, penyimpanan, *output*, dan kegiatan kontrol yang mengubah sumber daya data menjadi produk informasi, sistem informasi tergantung pada data, jaringan, perangkat lunak, perangkat keras, dan sumber daya manusia (pengguna akhir dan spesialis sistem informasi). Laudon & Laudon (2014) menjelaskan bahwa dalam membangun sistem informasi yang baru merupakan salah satu jenis perubahan organisasi yang direncanakan. Terdapat 4 jenis perubahan yang dapat memungkinkan teknologi yaitu otomatisasi, rasionalitas prosedur, perubahan paradigma dan mendesain ulang proses bisnis.

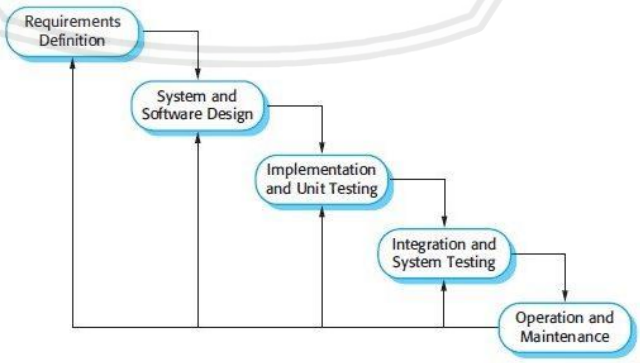
Pengenalan sistem informasi yang baru menurut Laudon & Laudon (2014) bukan hanya terkait dengan perangkat keras dan perangkat lunak saja, tetapi juga berkaitan dengan perubahan dalam pekerjaan, manajemen, keterampilan, dan

organisasi. Dalam membangun sistem yang baru, perlu diperhatikan bagaimana sistem itu dapat mempengaruhi proses bisnis yang telah ada serta organisasinya secara keseluruhan. Sistem informasi dibangun untuk memberikan solusi dari sebagian atau serangkaian masalah yang dirasakan atau dihadapi oleh organisasi. Sistem informasi diimplimentasikan dari desain yang diterjemahkan ke dalam bentuk kode program dengan menggunakan kelas-kelas yang tersedia pada *library* atau menambahkan yang baru dengan melibatkan *database* dalam implementasinya. Setelah sistem dapat dihasilkan maka dilakukan pengujian dan pengevaluasian terhadap sistem secara menyeluruh.

2.4 *Software Development Life Cycle* (SDLC)

Salah satu metode dalam mengembangkan solusi sistem informasi dengan menggunakan pendekatan sistem disebut sebagai SDLC (*Software Develompment Life Cycle*). Tahapan-tahapan pada SDLC terdiri dari investigasi, analisis, desain, implementasi dan pemeliharaan. Setiap tahapan memiliki keterikatan satu sama lain dan tahapan sebelumnya yang telah dilakukan dapat dilakukan kembali atau berulang (O'Brien & Marakas, 2011). SDLC menurut Alshamrani & Bahattab (2015) adalah metodologi yang digunakan dalam merancang, membangun dan memelihara sistem industri dan informasi. Setiap model pada SDLC memiliki fasenya masing-masing yang harus dikerjakan agar sistem yang dikembangkan dapat tercapai dan terselesaikan. SDLC memiliki banyak model dengan kelebihan dan kekurangannya masing-masing. Menurut Mishra & Dubey (2013) dengan adanya penggunaan SDLC, perangkat lunak yang dikembangkan dapat terselesaikan sesuai dengan standar dan waktu yang telah ditentukan. Setiap proses dalam pengembangan perangkat lunak terbagi menjadi tahapan-tahapan logis sehingga memungkinkan untuk mengatur waktu penyelesaian, anggaran yang dibutuhkan, dan pengerjaannya yang efisien. Model SDLC yang akan digunakan dalam penelitian ini adalah model *waterfall.*

2.4.1 Model *Waterfall*



Gambar 2.2 Tahapan Fase pada Model *Waterfall*

Sumber: Sommerville (2011)

Proses yang terjadi pada model *waterfall* menurut Sommerville (2011) berjalan karena adanya sebuah perencanaan. Oleh karena itu, pada dasarnya

semua tahapan yang ada pada model *waterfall* dijadwalkan terlebih dahulu sebelum dapat dimulai pengerjaannya. Pada model *waterfall* terdapat 5 tahapan fase yaitu *requirements definition*, *system and software design*, *implementation and unit testing*, *integration and system testing,* dan *operation and maintenance.* Pada model *waterfall* jika tahapan sebelumnya belum terselesaikan maka tahap selanjutnya tidak dapat dilakukan (Sommerville, 2011). Berikut ini merupakan penjelasan dari setiap tahapan fase pada model *waterfall* menurut Sommerville (2011) yang ditunjukkan oleh Gambar 2.2, yaitu:

1. *Requirements Definitions*

   Pada tahap *requirements definitions* dilakukan analisis terhadap kebutuhan dan batasan terhadap sistem atau layanan yang nantinya akan dibuat. Serta menetapkan tujuan dibuatnya sistem bersama dengan pengguna sistem. Setelah itu, dibuat rincian mengenai spesifikasi dari sistem yang akan dibuat.

2. *System and Software Design*

   Pada tahap ini dilakukan pengalokasian persyaratan kedalam bentuk perangkat lunak atau perangkat keras sistem dengan membangun sistem arsitektur secara keseluruhan. Perancangan dalam perangkat lunak terkait dengan penggambaran atau identifikasi dari perangkat lunak.

3. *Implementation and Unit Testing*

   Pada tahap *implementation and unit testing* perancangan dari perangkat lunak yang telah dilakukan direalisasikan sebagai unit program atau satu set program. Kemudian dilakukan verifikasi pada setiap unit untuk memastikan telah terpenuhi spesifikasi atau belum.

4. *Integration and System Testing*

   Pada tahap *integration and system testing*, sistem yang lengkap diintegrasikan dan diuji untuk memastikan perangkat lunak telah memenuhi persyaratan. Setelah dilakukan pengujian sistem dapat digunakan oleh pengguna.

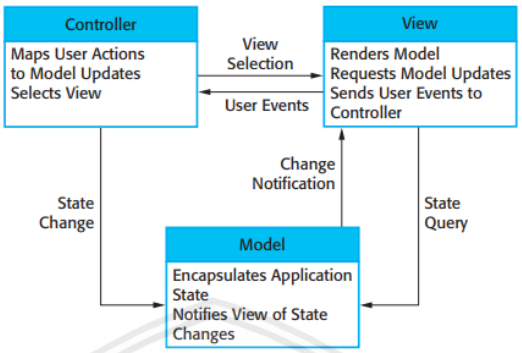
5. *Operation and Maintenance*

   Pada tahap *operation and maintenance* merupakan siklus hidup terlama. Sistem dipasang dan digunakan secara praktis oleh pengguna. Untuk perawatan merupakan proses pengoreksian atau pembetulan terhadap kesalahan yang mungkin tidak ditemukan selama siklus hidup.

2.4.2 Model-View-Controller (MVC)

Menurut Sommerville (2011) *Model-View-Controller* (MVC) memiliki 3 komponen yang memiliki perannya masing-masing dan saling berinteraksi dalam sistem. Komponen *view* berperan dalam mengelola tampilan dari data yang akan dilihat oleh pengguna, komponen *controller* berperan dalam mengelola interaksi pengguna saat menggunakan sistem seperti menekan tombol atau mengklik ikon,

selain itu *controller* juga menghubungkan interaksi antara *view* dan *model*. Komponen *model* berperan dalam mengelola data dan operasi dari data tersebut. Pada Gambar 2.3 menunjukkan pola pengorganisasian dari *Model-View-Controller* (MVC).



Gambar 2.3 Pola *Model-View-Controller* (MVC)

Sumber : Sommerville (2011)

2.4.2.1 Kerangka Kerja Codeigniter

Kerangka kerja merupakan pengembangan untuk membuat aplikasi atau subsistem yang lebih spesifik dari struktur generik (Sommerville, 2011). Kerangka kerja Codeigniter menurut Raharjo (2018) merupakan kerangka kerja web yang menggunakan bahasa pemrograman PHP dengan pola desain *Model View Controller* (MVC). *Model* dan *Controller* merupakan komponen berupa kelas yang merupakan implementasi dari pemrograman berorientasi objek. Dibandingkan dengan kerangka kerja web lainnya, Codeigniter memiliki desain lebih sederhana dan fleksibel. Pengembang yang menggunakan Codeigniter dapat menggunakan kerangka kerja secara parsial atau keseluruhan. Codeigniter memiliki beberapa kelebihan yaitu :

1. Bersifat *free* dan *open souce.*
2. Codeigniter memiliki ukuran file yang lebih kecil, setelah dilakukan instalasi ukuran file yang dimiliki sebesar 2 MB.
3. Dengan penggunaan Codeigniter aplikasi dapat berjalan lebih cepat.
4. Pola desain dari kerangka kerja Codeigniter adalah MVC sehingga didalam satu file tidak terisi dengan banyak kode program. Serta dengan pola desain MVC ini dapat membantu dalam proses membaca, memahami, dan pemeliharaannya dikemudian hari.
5. Dapat dilakukan pengembangan sesuai dengan kebutuhan.
6. Dokumentasi dari Codeigniter sangat baik sehingga dapat dengan mudah untuk menemukan pustaka kelas dan fungsi Codeigniter dalam dokumentasi yang ada dalam paket distribusinya.

2.5 Proses Bisnis

Menurut Monk & Wagner (2013) proses bisnis merupakan serangkaian kegiatan yang dilakukan untuk menghasilkan keluaran yang bernilai untuk pelanggan yang berasal dari satu atau lebih jenis masukan. Pelanggan yang terdapat pada proses bisnis dapat berupa pelanggan eksternal atau pelanggan internal. Pelanggan eksternal meliputi orang yang membeli produk berupa barang jadi, sedangkan pelanggan internal meliputi rekan atau kolega dari departemen lain. Proses bisnis menurut Weske (2007) merupakan koordinasi dari serangkaian kegiatan yang dilakukan pada lingkungan organisasi dan teknis untuk dapat mencapai tujuan bisnis. Proses bisnis yang ada pada sebuah organisasi dapat berinteraksi dengan proses bisnis dari organisasi yang lain.

2.6 *Business Process Model and Notation* (BPMN)

Menurut Object Management Group (2013) dengan adanya *Business Process Model and Notation* (BPMN) dapat memberikan gambaran notasi-notasi yang dapat dipahami oleh semua pengguna bisnis dalam membuat konsep awal proses yang dilakukan oleh analis bisnis, mengimplementasikan teknologi yang nantinya melakukan proses yang dilakukan oleh pengembang teknis, serta pihak-pihak yang ada didalam bisnis yang nantinya mengelola dan memantau proses-proses tersebut.

Object Management Group (2013) juga menjelaskan, BPMN dikembangkan untuk menciptakan mekanisme pemodelan proses bisnis yang mudah untuk dilakukan, tetapi BPMN juga dapat menangani proses bisnis yang kompleks. Untuk menangani dua hal yang berbeda ini, dilakukan pendekatan untuk mengelompokkan aspek grafis notasi ke dalam kategori tertentu. Kategori tersebut dibagi menjadi lima, yaitu *flow objects, data, connecting objects, swimlanes,* dan *artifacts*.

2.6.1 Flow Objects

Menurut Object Management Group (2013) *flow objects* merupakan bagian dari elemen grafis utama yang menggambarkan perilaku dari proses bisnis dan terbagi menjadi tiga macam, yaitu:

1. *Events.*

   Menurut Object Management Group (2013) pada proses yang sedang berlangsung, *event* menjadi sesuatu yang terjadi didalamnya dan mempengaruhi aliran pada model dengan adanya sebab (pemicu) atau dampak (hasil). *Events* terbagi menjadi tiga macam yaitu *start, intermediate,* dan *end*. Penjelasan terkait tiga macam *events* ditunjukkan oleh Tabel 2.1.

Tabel 2.1 Notasi-Notasi dari *Events*

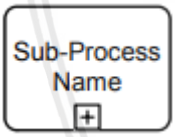
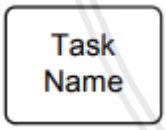
Notasi	Nama	Deskripsi
	Start	Notasi yang digunakan untuk memulai proses tertentu.
	Intermediate	Notasi ini mempengaruhi aliran proses namun tidak dapat memulai atau menghentikan proses. Notasi ini dapat digunakan diantara *start* dan *end event*.
	End	Notasi yang digunakan untuk mengakhiri proses.

Sumber: Object Management Group (2013)

2. *Activities.*

Activities menurut Object Management Group (2013) merupakan aktivitas atau kegiatan yang dilakukan dalam suatu proses. *Activities* terbagi menjadi dua macam yaitu *sub-process* dan *task*. Penjelasan terkait dua macam *activities* ditunjukkan oleh Tabel 2.2.

Tabel 2.2 Notasi-Notasi dari *Activities*

Notasi	Nama	Deskripsi
	*Sub-Process*	Notasi ini menggambarkan adanya proses didalamnya dengan tingkatan yang lebih rendah.
	*Task*	Notasi ini digunakan ketika proses tidak dipecah lagi kedalam proses yang lebih detail sehingga bersifat *atomic*.

Sumber: Object Management Group (2013)

3. *Gateways.*

Gateways menurut Object Management Group (2013) digunakan untuk mengkontrol alur dari proses bisnis sesuai dengan tipe percabangan. Penjelasan terkait tipe dari percabangan ditunjukkan oleh Tabel 2.3.

Tabel 2.3 Notasi-Notasi dari *Gateways*

Notasi	Nama	Deskripsi
	Parallel	Notasi digunakan untuk membuat jalur paralel dengan melakukan sinkronisasi jalur-jalur (menggabungkan) untuk menghasilkan jalur bersama pada jalur paralel.

Tabel 2.3 Notasi-Notasi dari *Gateways* (lanjutan)

Notasi	Nama	Deskripsi
atau	*Exclusive*	Notasi ini digunakan didalam proses untuk membuat jalur alternatif dengan satu jalur alternatif saja yang dapat dipilih. Oleh karena itu, hanya ada satu jalur masuk dan satu jalur keluar.
	Inclusive	Notasi tidak hanya digunakan untuk membuat jalur alternatif saja tetapi juga jalur paralel dengan setiap jalur atau gabungan dari jalur dapat dipilih.
atau	*Event-Based*	Notasi digunakan ketika jalur alternatif di dalam proses didasarkan pada suatu *event*.
	Parallel Event-Based	Notasi ini mirip dengan *parallel gateway* akan tetapi terdapat *event* yang mendasarinya. Jika suatu *event* di *trigger* maka *event* yang lainnya akan menunggu sampai sebelum proses *normally* selesai.
	Complex	Notasi digunakan untuk menggambarkan sinkronisasi yang kompleks. Notasi ini mirip dengan *inclusive gateway.*

Sumber: Object Management Group (2013)

2.6.2 Data

Persyaratan tradisional dari pemodelan proses adalah untuk memodelkan item yang dibuat, dimanipulasi dan digunakan di dalam proses. Hal terpentimg dari aspek ini adalah kemampuan untuk menangkap, meminta, atau memanipulasi struktur data itu. Pada BPMN terdapat beberapa elemen yang dapat menyimpan dan menyampaikan item selama proses yang disebut sebagai *item-aware* (Object Management Group, 2013). Penjelasan terkait elemen-elemen *item-aware* tersebut ditunjukkan oleh Tabel 2.4.

Tabel 2.4 Notasi-Notasi dari *Data*

Notasi	Nama	Deskripsi
	Data Objects	Notasi ini digunakan di dalam aliran proses untuk memodelkan data.
	Data Inputs	Notasi ini digunakan untuk menunjukkan masukkan ke proses tingkat atas.
	Data Outputs	Notasi ini digambarkan untuk menunjukkan keluaran dari proses.
	Data Stores	Notasi ini digunakan ketika aktivitas mengambil atau memperbarui informasi yang tersimpan.

Sumber: Object Management Group (2013)

2.6.3 Connecting Objects

Connecting Objects pada BPMN digunakan untuk menghubungkan *object-object* dengan menggunakan *flow* (Object Management Group, 2013). Penjelasan terkait tipe dari *connecting objects* ditunjukkan oleh Tabel 2.5.

Tabel 2.5 Notasi-Notasi dari *Connecting Objects*

Notasi	Nama	Deskripsi
	Sequence Flows	Notasi ini digunakan untuk menunjukkan urutan alur aktivitas dalam suatu proses.
	Messages Flows	Notasi ini digunakan untuk menunjukkan aliran pesan antara dua partisipan yang siap mengirim dan menerima.
	Associations	Notasi digunakan untuk menghubungkan *flow objects* dengan artefak dan informasi.
	Data Associations	Notasi ini digunakan untuk memindahkan data antara objek data, properti, masukkan dan keluaran dari *event*, proses dan *GlobalTasks*.

Sumber: Object Management Group (2013)

2.6.4 Swimlanes

Menurut Object Management Group (2013), *swimlanes* merupakan wadah yang berisi sekumpulan aktivitas-aktivitas. Notasi yang ada pada s*wimlanes* yaitu *pool* dan *lane*. Penjelasan terkait *pool* dan *lane* ditunjukkan oleh Tabel 2.6.

Tabel 2.6 Notasi-Notasi dari *Swimlanes*

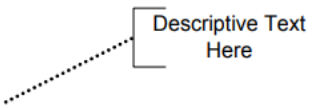
Notasi	Nama	Deskripsi
	Pool	Notasi ini digunakan untuk menggambarkan partisipan yang ada pada kolaborasi. Notasi ini bisa menjadi wadah untuk sekumpulan aktivitas dari *pool* lain (B2B).
	Lane	Notasi ini merupakan sub-partisi pada proses atau *pool* yang digunakan untuk mengelompokkan aktivitas.

Sumber: Object Management Group (2013)

2.6.5 Artifacts

Artifacts pada BPMN digunakan untuk memberikan keterangan tambahan untuk suatu proses (Object Management Group, 2013). Penjelasan terkait notasi pada *artifacts* ditunjukkan oleh Tabel 2.7.

Tabel 2.7 Notasi-Notasi dari *Artifacts*

Notasi	Nama	Deskripsi
	Group	Notasi ini digunakan untuk menggelompokkan grafis sesuai dengan kategorinya.
	*Text Annotation*	Notasi ini digunakan pada diagram BPMN untuk memberikan informasi tambahan kepada pembaca berupa teks.

Sumber: Object Management Group (2013)

2.7 Pemodelan *Use Case*

Use case digunakan untuk menggambarkan bagaimana sistem akan bekerja dan digunakan oleh pemangku kepentingan. Dengan *use case* dapat membantu dalam melakukan diskusi dan membangun pemahamanan yang sama antara semua pemangku kepentingan dari suatu sistem serta bagian pengembang dapat memahami lebih mendalam mengenai perilaku sistem, desain sistem dan

artefak-artefak lain yang digunakan dalam pembangunan sistem. Perlu adanya untuk mengidentifikasi siapa yang yang akan menggunakan sistem dan apa yang mereka harapkan (Bittner & Spence, 2002).

2.7.1 Identifikasi Tipe Pemangku Kepentingan

Pemangku kepentingan menurut Bittner & Spence (2002) merupakan seseorang yang akan dipengaruhi secara material oleh hasil dari sistem atau proyek yang menghasilkan sistem. Tipe pemangku kepentingan merupakan pemangku kepentingan yang memiliki karakteristik serta hubungan yang sama dengan sistem atau proyek yang menghasilkan sistem. Saat melakukan pengidentifikasian tipe pemangku kepentingan, fokuskan pada bagaimana sistem atau proyek yang dihasilkan mempengaruhi pemangku kepentingan. Bittner & Spence mengkategorikan pemangku kepentingan kedalam lima kategori yaitu:

1. *Users* (pengguna), merupakan pemangku kepentingan yang berinteraksi langsung dengan sistem dan menjadi aktor didalam pemodelan *use case*.
2. *Sponsors* (sponsor), merupakan pemangku kepentingan yang berinvestasi pada pembangunan sistem akan tetapi tidak terlibat secara langsung sebagai pengguna sistem.
3. *Developers* (pengembang), merupakan pemangku kepentingan yang terlibat dalam memberikan dukungan dan melakukan produksi.
4. *Authorities* (pihak yang berwenang), merupakan pemangku kepentingan yang ahli dalam aspek tertentu dalam domain masalah atau solusi seperti otoritas legislatif, pakar domain, badan pengawas eksternal, pakar teknologi, standar organisasi, dan departemen tata kelola organisasi.
5. *Customers* (pelanggan), merupakan pemangku kepentingan yang akan membeli sistem yang telah *final.*

2.7.2 Analisis Masalah

Analisis masalah yang dilakukan nantinya dapat memunculkan solusi-solusi, serta untuk memastikan fungsionalitas sistem yang disediakan dapat mengurangi permasalahan yang dihadapi oleh pemangku kepentingan. Memahami masalah yang coba diselesaikan oleh pengguna merupakan cara yang dilakukan untuk memenuhi kebutuhan mereka. Untuk menangkap permasalah yang ada, dilakukan pembuatan pernyataan masalah yang berisi permasalah dan solusi dari hasil berdiskusi dengan pemangku kepentingan (Bittner & Spence, 2002). Kerangka penulisan untuk pernyataan masalah ditunjukkan oleh Tabel 2.8.

Tabel 2.8 Kerangka Penulisan untuk Analisis Masalah

The problem of	Pada bagian ini dijelaskan mengenai masalah yang dihadapi.

Tabel 2.8 Kerangka Penulisan untuk Analisis Masalah (lanjutan)

Affects	Pada bagian ini dijelaskan mengenai pemangku kepentingan yang terkena dampak dari permasalahan.
The impact of which is	Pada bagian ini dijelaskan mengenai dampak yang diterima oleh pemangku kepentingan dari permasalahan yang dihadapi.
A successful solution would	Pada bagian ini dijelaskan mengenai solusi yang diberikan berupa manfaat utama apa yang akan diperoleh.

Sumber: Bittner & Spence (2002)

2.7.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Menurut Bittner & Spence (2002) pemenuhan kebutuhan dari pemangku kepentingan merupakan cara yang dilakukan untuk menyelesaikan permasalahan. Pada pendeskripsian kebutuhan pemangku kepentingan harus dimasukkan alasan mengapa kebutuhan tersebut menjadi penting dan berdampak pada pemangku kepentingan. Perlunya adanya pemahaman mengenai:

1. Pemangku kepentingan dan pengguna menempatkan pemenuhan setiap kebutuhan sebagai kepentingan relatif.
2. Dari kebutuhan yang ada, pemangku kepentingan mana yang memerlukannya.
3. Permasalahan yang ada dijelaskan dengan proses bisnis yang saat ini berjalan.
4. Solusi yang diharapkan oleh pemangku kepentingan dijelaskan dalam bentuk situasi bisnis yang diinginkan.

2.7.4 Identifikasi Fitur

Fitur merupakan layanan atau kemampuan yang dimiliki oleh sistem untuk dapat memenuhi kebutuhan dan memberikan manfaat untuk pemangku kepentingan. Tingkatan detail fitur harus cukup umum agar dapat dipahami oleh pemangku kepentingan, karena fitur berisi kemampuan dan kualitas dari produk yang akan dibangun dan fitur harus dapat diakses oleh semua pemangku kepentingan atau semua angota tim proyek (Bittner & Spence, 2002). Kerangka penulisan untuk identifikasi fitur ditunjukkan oleh Tabel 2.9.

Tabel 2.9 Kerangka Penulisan untuk Identifikasi Fitur

Kode Fitur	Deskripsi
Pada bagian ini berisi kode dari fitur.	Pada bagian ini dijelaskan mengenai kemampuan yang dapat dilakukan oleh sistem.

Sumber: Diadaptasi dari Bittner & Spence (2002)

2.7.5 Identifikasi Persyaratan Fungsional dan NonFungsional

Menurut Bittner & Spence (2002) persyaratan terbagi menjadi 2 kategori yaitu persyaratan fungsional dan nonfungsional. Berikut ini adalah penjelasannya:

1. Persyaratan Fungsional, merupakan perilaku-perilaku yang harus sistem dapat lakukan. Pada persyaratan fungsional masukkan dan keluaran dari sistem ditentukan.
2. Persyaratan NonFungsional, merupakan kualitas lain yang perlu dimiliki oleh sistem yang terkait dengan kinerja, kebergunaan, kendala, dan sistem pendukung.

2.7.6 Pemodelan *Use Case*

Pemodelan *use case* menggambarkan penggunaan sistem oleh pengguna dengan penggambaran yang dilakukan berupa kumpulan aktor, *use case*, dan asosiasi aktor dengan *use case*. Dengan adanya pemodelan, sistem yang sedang dibangun dapat lebih mudah dipahami dan pemodelan yang dilakukan berdasarkan pada persyaratan fungsional sistem. Pada pemodelan *use case* di dalamnya terdapat *actors* dan *use case*. *Actors* merupakan orang atau objek lain yang berinteraksi dengan sistem. Penamaan untuk *actor* menggunakan nama atau deskripsi singkat yang berkaitan dengan interksi penggunaannya dengan sistem. *Use case* merupakan hal bernilai yang diberikan sistem kepada aktor (Bittner & Spence, 2002).

Untuk dokumentasi *use case*, setiap *use case* dijelaskan mengenai tujuan dari *use case* dan nilai yang dihasilkan oleh *use case* untuk pemangku kepentingan. Pendeskripsian dilakukan untuk setiap use case yang telah teridentifikasi. Di dalam setiap deskripsi *use case* harus berisi mengenai pemangku kepentingan yang menerima nilai spesifik dari sistem, alur yang dilakukan sistem untuk mencapai nilai, serta pendeskripsikan yang dilakukan jelas, tidak ambigu, dan singkat. Untuk kerangka penulisannya ditunjukkan oleh Tabel 2.10.

Tabel 2.10 Kerangka Penulisan untuk Spesifikasi *Use Case*

Brief Description	Pada bagian ini dijelaskan secara singkat mengenai *use case*.
Actor	Pada bagian ini dijelaskan mengenai aktor yang melakukan interaksi dengan *use case.*
Pre-condition	Pada bagian ini dijelaskan mengenai kondisi yang perlu dipenuhi agar alur dasar dapat dijalankan.
Post-condition	Pada bagian ini dijelaskan mengenai kondisi yang terjadi jika alur dasar berhasil dijalankan.
Extension	Pada bagian ini berisi *use case* yang merupakan pengembangan dari *use case* yang dilakukan.

Tabel 2.10 Kerangka Penulisan untuk Spesifikasi *Use Case* (lanjutan)

Basic Flow	Pada bagian ini dijelaskan mengenai alur dasar yang dijalankan oleh sistem.
Alternative Flow	Pada bagian ini dijelaskan mengenai alur yang dijalankan ketika alur dasar tidak berjalan dengan semestinya.
Subflow	Pada bagian ini dijelaskan mengenai alur-alur yang lebih sederhana dari alur-alur *use case* yang terlalu detail.

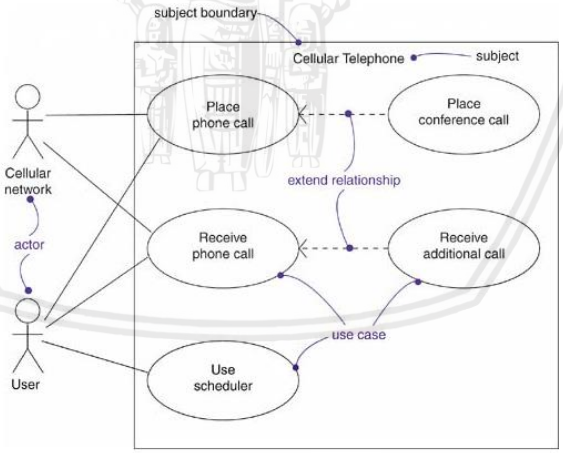
Sumber: Diadapatasi dari Bittner & Spence (2002)

2.8 *Unified Modeling Language* (UML)

Bahasa yang digunakan untuk memvisualisasikan, menspesifikasi, membangun dan mendokumentasi artefak dari sistem perangkat lunak yang intensif disebut sebagai UML (Booch, et al., 2005).

2.8.1 Use Case Diagram

Use case diagram menurut Booch et al. (2005) merupakan diagram UML yang digunakan untuk memodelkan perilaku dari sistem, subsistem atau kelas. Use case diagram diterapkan untuk menentukan, memvisualisasikan, dan mendokumentasikan perilaku dari sistem, subsistem, atau kelas sehingga pengguna dapat menggunakan elemen dan pengembang dapat menerapkan elemen tersebut. Gambar 2.4 menunjukkan contoh dari *use case diagram*.



Gambar 2.4 Contoh dari *Use Case Diagram*

Sumber: Booch et al. (2005)

Notasi-notasi yang terdapat pada *use case diagram* ditunjukkan oleh Tabel 2.11 yang terdiri dari *actor*, *use case*, *association*, *include*, *extend*, dan *generalization*.

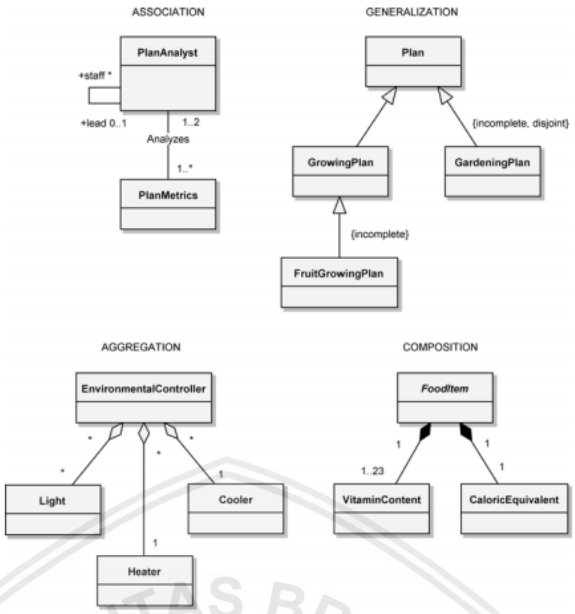
Tabel 2.11 Notasi-Notasi dari *Use Case Diagram*

Notasi	Nama	Deskripsi
	Actor	*Actor* merupakan entitas yang berinteraksi langsung dengan sistem, dapat berupa orang atau sistem lain.
	Use case	*Use case* merupakan kegiatan atau tindakan yang dilakukan *actor* terhadap sistem.
	Associations	*Associations* digunakan untuk menghubungkan antara *actor* dan *use case*.
	Include	*Include* merupakan pemanggilan *use case* oleh *use case* lain, sehingga untuk *use case* selanjutnya diperlukan *use case* disebelumnya.
	Extend	*Extend* digunakan untuk mengembangkan *use case* yang telah ada.
	Generalization	*Generalization* digunakan untuk menghubungkan use case. *Use case* dapat memiliki perilaku umum yang di dalamnya terdapat *use case* lain (*child use case*) yang dapat memodifikasi dengan menambahkan langkah.

Sumber: Booch et al. (2007)

2.8.2 Class Diagram

Class diagram menurut Booch et al. (2007) merupakan gambaran dari kelas-kelas dan hubungan antar kelas dalam suatu sistem. *Class diagram* tunggal merepresentasikan tampilan struktur kelas yang membentuk arsitektur sistem. Pada Gambar 2.5 merupakan contoh relasi-relasi antar *class* yang ada pada *class diagram*.

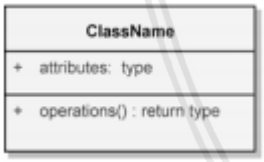
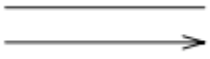


Gambar 2.5 Contoh Relasi antar *Class*

Sumber: Booch et al. (2007)

Notasi-notasi yang digunakan untuk menggambarkan hubungan antar kelas pada *class diagram* ditunjukkan oleh Tabel 2.12.

Tabel 2.12 Notasi-Notasi dari *Class Diagram*

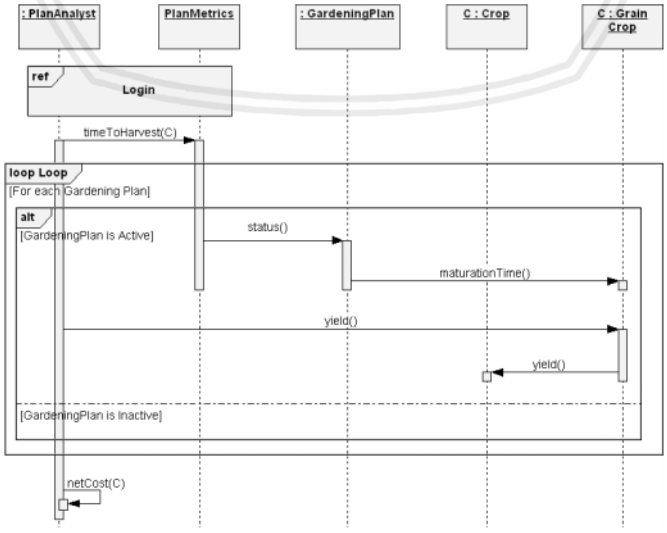
Notasi	Nama	Deskripsi
	*Class*	Notasi ini digunakan untuk menggambarkan kumpulan objek yang berbagi struktur, perilaku, dan semantik umum. Komponen yang terdapat pada kelas yaitu nama kelas, atribut, dan operasi.
	Aggregation	Notasi ini digunakan untuk menggambarkan kelas yang menjadi bagian dari objek agregat. Notasi ini menunjukkan hirarki bagian atau keseluruhan dan hubungan dari agregat ke bagian-bagiannya.
	*Association*	Notasi ini digunakan untuk menggambarkan hubungan semantik dan menghubungkan kelas yang satu dengan kelas yang lainnya.

Tabel 2.12 Notasi-Notasi dari *Class Diagram* (lanjutan)

Notasi	Nama	Deskripsi
	Composition	Notasi ini digambarkan dengan *diamond* yang terisi dan menunjukkan hubungan kontainmen. Bebeda dari agregasi, dengan notasi ini untuk pembuatan dan penghapusan kelas saling terkait.
	Generalization	Notasi ini digunakan untuk menggambarkan hubungan antara *superclass* dan *subclass*, dengan *subclass* yang mewarisi perilaku dan struktur dari *superclass* nya.
'+'	*Public* (visibility of a package elemen)	Notasi ini digunakan agar kelas dapat diakses oleh elemen manapun.
'-'	*Private* (visibility of a package elemen)	Notasi ini digunakan agar hanya elemen di dalam kelas itu sendiri yang bisa mengaksesnya.
'#'	*Protected* (visibility of a package elemen)	Notasi ini digunakan agar elemen pada subkelas dan kelas itu sendiri yang dapat mengaksesnya.

Sumber : Object Management Group (2005)

2.8.3 Sequence Diagram



Gambar 2.6 Contoh dari *Sequence Diagram*

Sumber: Booch et al. (2007)

Sequence diagram merupakan diagram UML yang menggambarkan kolaborasi antar objek. *Sequence diagram* digunakan untuk mendefinisikan urutan pesan antar objek berdasarkan waktu (Booch, et al., 2007). Contoh dari *sequence diagram* ditunjukkan oleh Gambar 2.6. *Sequence diagram* memiliki beberapa notasi yang ditunjukkan oleh Tabel 2.13.

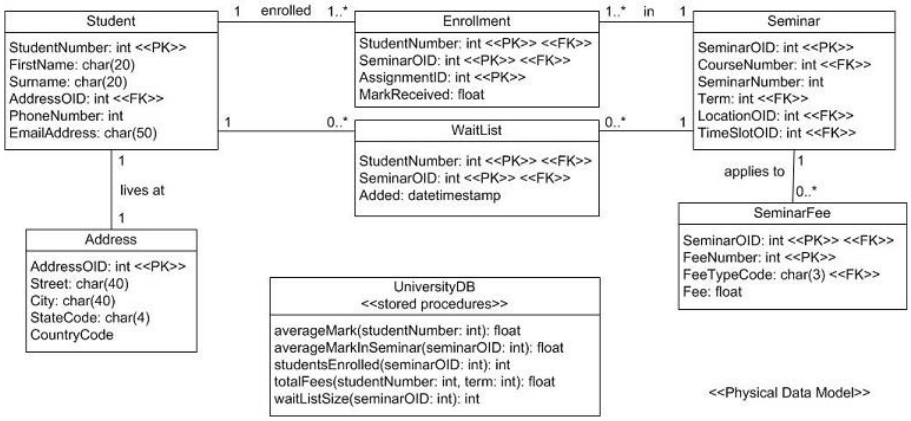
Tabel 2.13 Notasi-Notasi dari *Sequence Diagram*

Notasi	Nama	Deskripsi
	Frame	Notasi yang didalamnya terdapat interaksi-interaksi yang terjadi pada *sequence diagram*.
	Lifeline	Notasi ini menjelaskan mengenai garis waktu dari suatu proses.
	CombinedFragment	Notasi ini digunakan untuk menjelaskan ekspresi fragmen interaksi.
	Messages (Synchronous)	Notasi ini digunakan untuk interaksi antar *lifeline.*
	Reply messages	Notasi ini digunakan untuk pengembalian nilai dengan menggunakan pesan.

Sumber : Object Management Group (2005)

2.9 *Physical Data Model*

Menurut Inc (2013), *physical data model* merupakan pemodelan yang dilakukan untuk merancang skema dari *database* yang terdiri dari kolom data, tabel data dan hubungan antar tabel. Fitur pada *physical data model* meliputi spesifikasi dari semua tabel dan kolom, *foreign key* yang digunakan untuk menggambarkan hubungan antar tabel, denormalisasi dapat dilakukan sesuai dengan kebutuhan pengguna, *physical data model* untuk setiap RDBMS (*Relation Database Management System*) akan berbeda-beda seperti tipe data pada kolom SQL Server mungkin akan berbeda dengan MySQL, dan pertimbangan *physical* dapat membuat *physical data model* menjadi berbeda dengan *logical data model* (1keydata, 2009). Contoh dari penggambaran *physical data model* ditunjukkan oleh Gambar 2.7.



Gambar 2.7 Contoh dari *Physical Data Model*

Sumber: Inc (2013)

2.10 Pengujian Perangkat Lunak

Pengujian perangkat lunak menurut Sommerville (2011) dilakukan untuk menemukan kekurangan atau cacat pada program sebelum nantinya program tersebut digunakan oleh pengguna. Pengujian perangkat lunak juga dilakukan untuk memastikan program dapat berjalan atau berfungsi sesuai dengan persyaratan fungsional dan nonfungsional. Pengujian pada perangkat lunak dilakukan dengan menggunakan data buatan dan hasil dari uji coba yang dilakukan diperiksa untuk dapat menemukan kesalahan-kesalahan, keanehan atau kelainan pada program. Pada pengujian perangkat lunak terdapat dua tujuan yang berbeda yaitu :

1. Pengujian dilakukan untuk ditampilkan kepada pengembang dan pelanggan bahwa perangkat lunak yang dikembangkan telah memenuhi persyaratan.
2. Pengujian dilakukan untuk menemukan kesalahan-kesalahan pada perangkat lunak yang tidak sesuai dengan spesifikasinya.

2.10.2 Pengujian Validasi

Pengujian validasi dilakukan untuk menunjukkan kesesuaian antara perangkat lunak yang dikembangkan dengan persyaratan fungsional dengan menggunakan serangkaian tes. Serangkaian tes yang akan dilakukan dalam pengujian dan prosedur uji yang mendefinisikan kasus uji diuraikan dalam rencana pengujian. Validasi dapat dikatakan berhasil jika perangkat lunak yang dikembangkan dapat berfungsi sesuai dengan yang diharapkan oleh pengguna atau persyaratan fungsional dapat tercapai (Pressman, 2010).

Menurut Pressman (2010) terdapat dua kondisi yang mungkin terjadi setelah dilakukannya uji kasus validasi, yaitu :

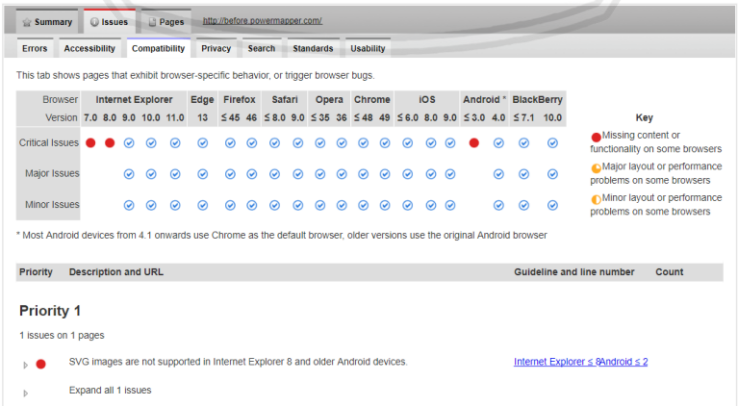
1. Karakteristik kinerja atau fungsi dari perangkat lunak telah sesuai dengan spesifikasi dan dapat diterima.

2. Terdapat kesalahan-kesalahan berupa ditemukannya penyimpangan dari spesifikasi dan dilakukan pencatatan kekurangan.

2.10.3 Pengujian Kompatibilitas

Pengujian kompatibilitas dilakukan untuk menemukan kesalahan-kesalahan terkait dengan lingkungan tertentu misalnya adalah *browser*. Aplikasi berbasis web dijalankan pada berbagai lingkungan sisi klien sehingga terdapat perbedaan pada komputer, perangkat layar, sistem operasi, *browser*, dan kecepatan koneksi jaringan yang tersedia. Hal ini akan berpengaruh pada pemrosesan yang terjadi pada aplikasi berbasis web. Seperti penggunaan *browser* yang berbeda memungkinkan terdapatnya beberapa konfigurasi yang tidak tersedia. Pengujian kompatibilitas dilakukan sebelum aplikasi berbasis web dapat diakses secara *online*. Dalam pengujian kompatibilitas langkah awal yang perlu dilakukan adalah menentukan variasi dari konfigurasi komputasi sisi klien. Kemudian dilakukan beberapa tes berupa tes antarmuka, tes navigasi, tes kinerja dan tes keamanan. Dengan adanya serangkaian tes memungkinkan ditemukannya kesalahan atau masalah eksekusi pada konfigurasi yang berbeda (Pressman, 2010).

SortSite merupakan aplikasi yang digunakan untuk memeriksa situs web. Pada aplikasi SortSite terdapat beberapa poin yang dicek yaitu aksesibilitas, kompatibilitas, tautan rusak, optimasi mesin pencari, standar web, privasi, dan *usability*. Tes yang digunakan untuk kompatibilitas *browser* dilakukan pada *browser* Internet Explorer, Chrome, Safari, Opera, Firefox, Edge, iphone/iPad, Android, dan Blckberry. Pemeriksaan situs web dari kompatibilitas *website* untuk memastikan semua fungsi yang ada pada web dapat berfungsi pada beberapa *browser* yang berbeda. Masalah yang biasanya ada pada saat dilakukan tes kompatibilitas yaitu fitur css, tag HTML, format gambar, teknologi yang tidak didukung oleh semua *browser* (PowerMapper, 2019). Tampilan dari hasil pengujian kompatibilitas dengan menggunakan aplikasi Sortsite ditunjukkan Gambar 2.8.

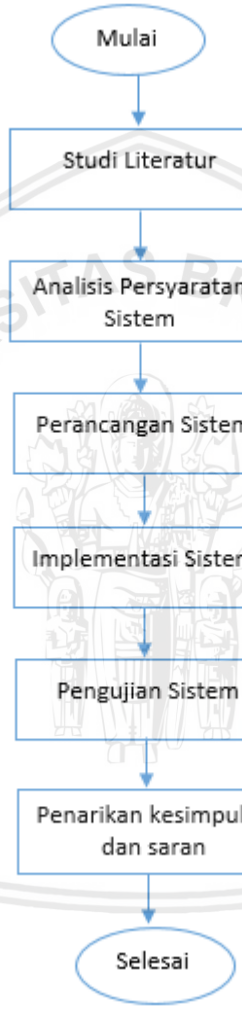


Gambar 2.8 Contoh dari Hasil Pengujian Kompatibilitas dengan Aplikasi SortSite

Sumber: PowerMapper (2019)

BAB 3 METODOLOGI

Bab ini menjelaskan mengenai tahapan-tahapan dalam melakukan penelitian. Metode yang digunakan dalam pengembangan Sistem Informasi Kehadiran Siswa menggunakan model *waterfall.* Penelitian ini diawali dengan melakukan studi literatur sampai dilakukannya penarikan kesimpulan dan pemberian saran. Diagram alir yang menggambarkan urutan pengerjaan dalam penelitian ini ditampilkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Pengerjaan Penelitian

3.1 Studi Literatur

Tahap studi literatur merupakan tahap yang dilakukan untuk mencari sumber rujukan yang terkait dengan penelitian yang dilakukan yaitu pengembangan Sistem Informasi Kehadiran Siswa. Sumber rujukan digunakan sebagai dasar teori dalam penulisan penelitian ini. Sumber rujukan yang digunakan diperoleh dari buku, *e-book*, jurnal, penelitian terdahulu dan artikel valid yang bersumber dari internet. Literatur yang digunakan dalam penulisan

penelitian ini berkaitan dengan sistem informasi, *Software Development Life Cycle* (SDLC), proses bisnis, *Business Process Model and Notation* (BPMN), pemodelan *use case*, *physical data model, Unified Modeling Language* (UML), pengujian perangkat lunak, dan penelitian terdahulu.

3.2 Analisis Persyaratan Sistem

Tahap analisis persyaratan dilakukan dengan melakukan wawancara dan studi dokumentasi untuk mengetahui pengelolaan data kehadiran siswa yang ada pada SMA Suluh Jakarta. Dengan adanya wawancara yang dilakukan dapat diketahui kendala dan kebutuhan dari pemangku kepentingan. Wawancara dilakukan dengan Guru BK selaku pihak yang berkaitan secara langsung dengan presensi siswa dan guru piket selaku pihak yang berkaitan dengan pendataan kehadiran siswa yang terlambat atau tidak hadir. Kemudian dilakukan studi dokumentasi dengan melihat dokumen-dokumen hasil perekapan data kehadiran siswa yang saat ini ada.

Dari wawancara yang dilakukan dapat diketahui proses bisnis yang saat ini sedang berjalan pada SMA Suluh Jakarta. Kemudian dilakukan analisis terhadap proses bisnis yang ada dan digambarkan dalam bentuk BPMN. Proses bisnis terkait pendataan kehadiran siswa menjadi hal yang penting bagi pihak sekolah karena dengan adanya pendataan kemudian dilakukan perekapan, persentase kehadiran siswa dapat dipantau. SMA Suluh juga memiliki aturan jika siswa tidak hadir tanpa keterangan melebihi 15% dari jumlah hari efektif sekolah maka siswa dinyatakan tidak naik kelas. Dengan adanya pendataan kehadiran juga dapat mendisiplinkan siswa sesuai dengan tujuan SMA Suluh Jakarta. Dari analisis permasalahan yang diperoleh dari wawancara, dibuat tabel terkait permasalahan yang dihadapi dan solusi yang diberikan terkait permasalahan yang ada. Kemudian dilakukan penggambaran proses bisnis *to-be* yang menjadi proses bisnis usulan peneliti terhadap proses bisnis yang saat ini ada. Untuk mengetahui pihak-pihak yang terlibat, dilakukan identifikasi tipe dari pemangku kepentingan.

Kegiatan yang dilakukan selanjutnya adalah identifikasi kebutuhan pengguna. Identifikasi kebutuhan pengguna dilakukan untuk dapat mengetahui situasi yang saat ini ada, kebutuhan yang diperlukan oleh pengguna serta solusi yang ditawarkan terkait dengan kebutuhan pengguna.

Kemudian dilakukan identifikasi pengguna, untuk mengelompokkan tipe calon pengguna yang akan menggunakan sistem berdasarkan pada tipe pemangku kepentingan. Dengan dilakukannya identifikasi pengguna dapat diketahui individu yang akan berinteraksi secara langsung dengan sistem. Setelah itu dilakukan identifikasi fitur untuk megetahui fungsi-fungsi yang akan ada pada sistem. Fungsi-fungsi ini merupakan realisasi dari solusi yang ditawarkan. Kemudian hasil dari identifikasi fitur dihubungkan dengan kebutuhan pengguna. Hal ini dilakukan untuk mengelompokkan fitur sebagai solusi dari kebutuhan pengguna yang telah teridentifikasi. Setelah itu untuk dapat mengetahui kondisi yang harus dimiliki oleh sistem agar fitur yang ada dapat terealisasikan, dibuat persyaratan fungsional sistem dan dibuat persyaratan nonfungsional untuk mendukung sistem. Setelah

itu dilakukan pemodelan *use case* untuk dapat menggambarkan interaksi yang dapat dilakukan oleh aktor terhadap sistem. Kemudian penjelasan terkait *use case* dijelaskan dalam spesifikasi *use case*.

3.3 Perancangan Sistem

Setelah tahap analisis persyaratan dilakukan, tahap yang dilakukan selanjutnya adalah perancangan sistem. Perancangan sistem merupakan pemodelan sistem ke dalam bentuk *Unified Modeling Language* (UML). Perancangan pada Sistem Informasi Kehadiran Siswa yang dilakukan berupa perancangan arsitektur, diagram kelas, *sequence diagram,* basis data, dan antarmuka pengguna.

3.4 Implementasi Sistem

Tahap yang dilakukan setelah perancangan sistem adalah implementasi sistem. Tahap implementasi merupakan tahapan pengimplementasian rancangan yang telah dibuat sebelumnya kedalam betuk kode program. Implementasi sistem dilakukan dengan menggunakan kerangka kerja Codeigniter untuk membuat antarmuka dan fungsi dari sistem. Sedangkan untuk penyimpanan datanya menggunakan MySQL yang dapat diakses melalui PhpMyadmin. Hasil dari implementasi yang dilakukan berupa Sistem Informasi Kehadiran Siswa yang menyediakan fitur-fitur yang sesuai dengan hasil analisis persyaratan.

3.5 Pengujian Sistem

Setelah tahap implementasi dilakukan, tahap yang dilakukan selanjutnya yaitu pengujian sistem. Pengujian yang dilakukan adalah dengan menggunakan pengujian validasi dan kompatibilitas. Pengujian validasi dilakukan untuk memastikan sistem yang dikembangkan telah sesuai dengan spesifikasi persyaratan dan dapat diterima oleh pemangku kepentingan. Sedangkan pengujian kompatibilitas dilakukan untuk memastikan pada beberapa peramban yang berbeda sistem yang dikembangkan dapat berjalan dan berfungsi dengan baik tanpa adanya konten yang hilang.

3.6 Penarikan Kesimpulan dan Saran

Tahap terakhir yang dilakukan adalah menarik kesimpulan berdasarkan tahapan-tahapan yang telah dilakukan sebelumnya. Selain penarikan kesimpulan, diberikan juga saran untuk penelitian selanjutnya, agar menjadi pertimbangan dalam pengembangan lebih lanjut terhadap sistem yang telah dibuat serta memperbaiki kesalahan-kesalahan yang masih terdapat pada sistem yang telah dikembangkan.

BAB 4 ANALISIS PERSYARATAN

Bab ini menjelaskan mengenai analisis persyaratan yang diperoleh dengan melakukan wawancara dan studi dokumentasi. Hasil dari analisis persyaratan berupa pemodelan proses bisnis berupa proses bisnis *as-is* dan *to-be*, identifikasi tipe pemangku kepentingan, analisis masalah, identifikasi kebutuhan pemangku kepentingan dan pengguna, identifikasi pengguna, identifikasi fitur, persyaratan fungsional dan nonfungsional, serta pemodelan *use case*.

4.1 Pemodelan Proses Bisnis

Pemodelan proses bisnis digunakan untuk melakukan pemodelan terhadap aktivitas-aktivitas bisnis yang dilakukan oleh SMA Suluh Jakarta untuk memenuhi kebutuhan dalam pengelolaan data kehadiran siswa. Informasi terkait proses bisnis yang ada pada SMA Suluh Jakarta, didapatkan dengan melakukan wawancara dengan guru BK selaku pihak yang menangani presensi siswa, guru piket selaku guru yang berkaitan dengan pendataan kehadiran siswa dan dilakukan studi dokumentasi dengan melihat dokumen-dokumen perekapan data kehadiran yang saat ini ada. Dalam pemodelan proses bisnis terdapat dua pemodelan yaitu proses bisnis yang sedang berjalan (*as-is)* dan proses bisnis yang diusulkan (*to-be).*

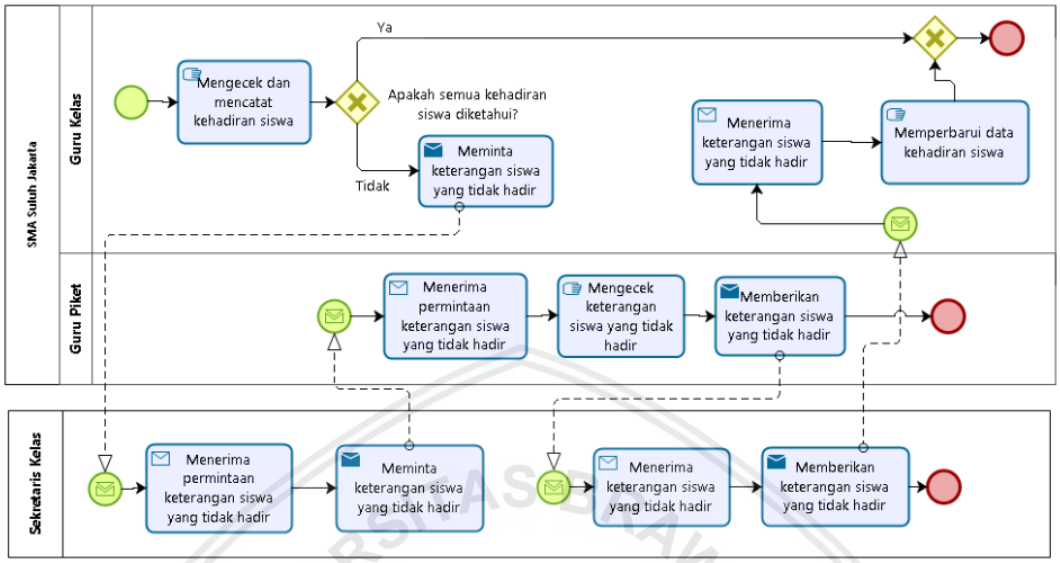
4.1.1 Proses Bisnis *As-Is*

Proses bisnis *as-is* merupakan proses bisnis yang saat ini sedang berjalan, sehingga proses bisnis as-is dalam penelitian ini diidentifikasi berdasarkan kondisi saat ini ada pada SMA Suluh Jakarta dan sebelum diberikannya solusi terhadap proses bisnis tersebut. Dengan adanya analisis dan pemodelan terhadap proses bisnis yang saat ini sedang berjalan (*as-is)* dapat memudahkan peneliti dalam mengidentifikasi solusi yang memungkinkan bagi pihak SMA Suluh Jakarta, sehingga proses bisnis yang saat ini sedang berjalan (*as-is)* dapat diperbaiki atau ditingkatkan lagi. Hasil dari analisis dan pemodelan proses bisnis yang dilakukan digambarkan dengan menggunakan BPMN, agar pemangku kepentingan dapat lebih mudah memahami proses bisnis yang saat ini sedang berjalan (*as-is).*

4.1.1.1 Proses Bisnis As-Is Mendata Kehadiran Siswa

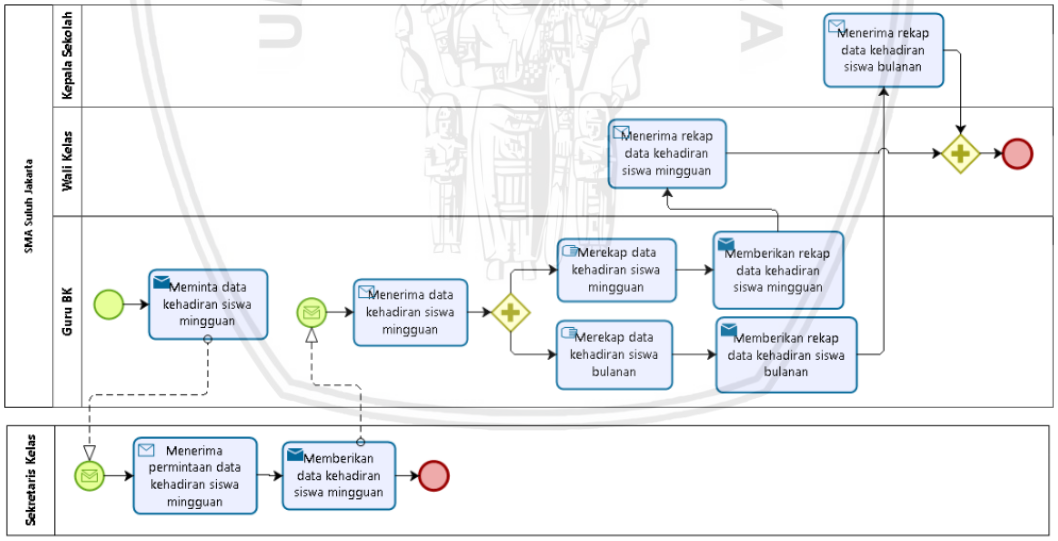
Proses bisnis untuk mendata kehadiran siswa yang saat ini sedang berjalan pada SMA Suluh Jakarta dapat dilihat pada Gambar 4.1, dengan guru kelas, sekretaris kelas dan guru piket sebagai unit bisnis yang melakukan pendataan kehadiran siswa. Proses bisnis ini dimulai ketika guru kelas mengecek dan mencatat kehadiran siswa, jika data kehadiran semua siswa diketahui maka pendataan kehadiran siswa selesai dilakukan. Tetapi jika terdapat kehadiran siswa yang tidak diketahui maka terdapat proses meminta keterangan siswa yang tidak hadir kepada guru piket. Karena jika terdapat siswa yang terlambat atau orang tua siswa yang datang ke sekolah untuk memberikan surat keterangan sakit atau izin siswa yang tidak dapat hadir di sekolah akan melalui guru piket. Kemudian guru

piket akan melakukan pengecekkan keterangan kehadiran siswa yang diminta oleh sekretaris kelas dan memberikan data tersebut kepada guru kelas melalui sekretaris kelas, sehingga data kehadiran siswa dapat diperbarui.



Gambar 4.1 Proses Binis *As-Is* dari Mendata Kehadiran Siswa

4.1.1.2 Proses Bisnis As-Is Merekap Data Kehadiran



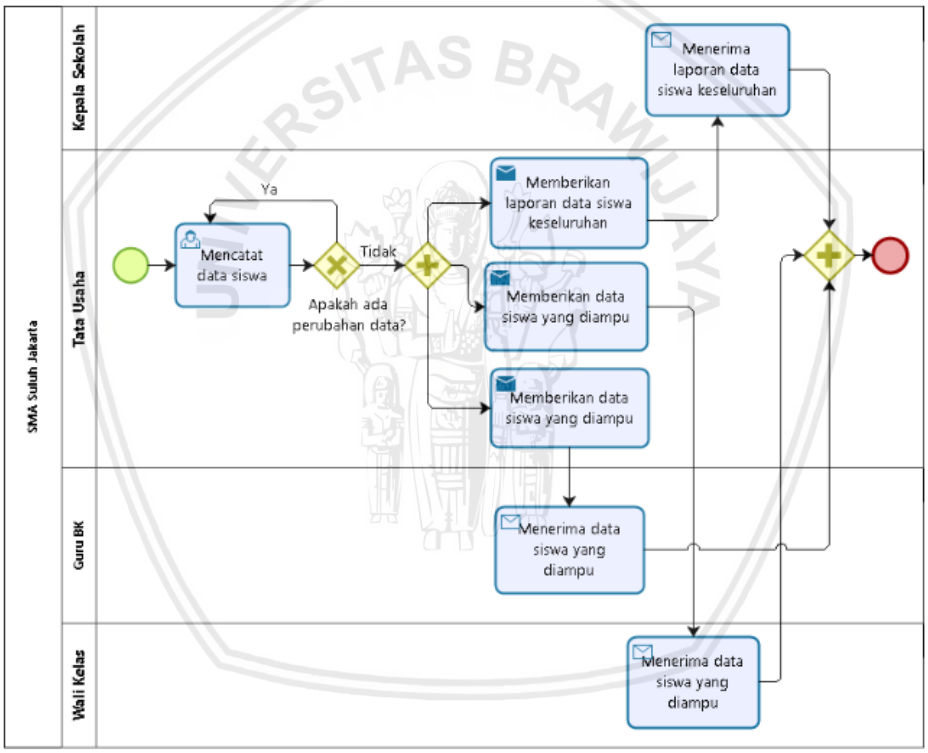
Gambar 4.2 Proses Binis *As-Is* dari Merekap Data Kehadiran

Proses bisnis untuk merekap data kehadiran yang saat ini sedang berjalan pada SMA Suluh Jakarta dapat dilihat pada Gambar 4.2, dengan unit bisnis yang melakukannya adalah sekretaris kelas, guru BK, wali kelas, dan kepala sekolah. Proses bisnis ini dimulai ketika guru BK meminta data kehadiran siswa mingguan kepada sekretaris tiap kelas. Kemudian dari data kehadiran siswa yang diterima dari tiap kelas, guru BK melakukan perekapan data kehadiran siswa mingguan dengan memasukkan data tersebut ke dalam Microsoft Excel. Setelah itu dalam bentuk laporan, data kehadiran siswa diberikan kepada wali kelas. Guru BK juga

memiliki jadwal perekapan data kehadiran siswa bulanan dan data yang telah direkap diberikan kepada kepala sekolah.

4.1.1.3 Proses Bisnis As-Is Mengelola Data Siswa

Proses bisnis untuk mengelola data siswa yang saat ini sedang berjalan pada SMA Suluh Jakarta dapat dilihat pada Gambar 4.3 dengan unit bisnis yang melakukannya adalah kepala sekolah, wali kelas, guru BK, dan tata usaha. Proses bisnis ini dimulai ketika tata usaha mencatat data siswa. Jika ada perubahan data siswa, maka data akan dicatat kembali untuk diperbarui. Namun jika sudah tidak ada perubahan data, maka data tersebut dapat diberikan kepada kepala sekolah berupa data siswa secara keseluruhan. Data tersebut juga diberikan kepada wali kelas dan guru BK berupa data siswa yang diampu. Nantinya data siswa yang diterima oleh wali kelas digunakan sebagai acuan dalam melakukan pengecekkan data kehadiran siswa.



Gambar 4.3 Proses Bisnis *As-Is* dari Mengelola Data Siswa

4.1.2 Analisis Masalah

Analisis permasalahan dilakukan untuk mengetahui permasalahan yang terjadi pada SMA Suluh Jakarta dan harus diselesaikan oleh pemangku kepentingan yang ada. Dalam penelitian ini, analisis masalah didapatkan dengan melakukan wawancara dengan ibu Dina Apriliyati selaku guru BK yang menangani perekapan data kehadiran setiap minggunya. Pada Tabel 4.1 dijelaskan analisis permasalahan yang telah dilakukan dengan melakukan wawancara bersama ibu Dina Apriliyati.

Tabel 4.1 Analisis Permasalahan 1

The problem of	• Perekapan data kehadiran hanya dapat dilakukan seminggu sekali dengan melakukan pencatatan data kehadiran siswa sebanyak dua kali yaitu saat pendataan kehadiran dikelas dan saat jadwal perekapan satu minggu sekali. • Hasil dari data kehadiran siswa dibutuhkan tidak selalu tersedia atau memerlukan proses mencari data terlebih dahulu.
Affects	Pengguna dan pelanggan.
The impact of which is	• Data yang dimasukkan rawan terhadap kesalahan-kesalahan. • Data tidak bisa dilaporkan kapanpun jika diperlukan.
A successful solution would	Sistem yang menyediakan layanan untuk pendataan kehadiran dengan data yang telah diperoleh tersimpan kedalam sistem penyimpanan data yang terpusat, serta data yang telah dimasukkan dapat ditampilkan kapanpun jika diperlukan.

Selain itu, peneliti juga melakukan wawancara dengan dengan ibu Dessy Akhriani selaku guru BK dan juga dapat bertugas sebagai guru piket. Dari hasil wawancara yang dilakukan didapatkan analisis permasalahan yang ada. Pada Tabel 4.2 dijelaskan analisis permasalahan yang telah dilakukan dengan melakukan wawancara bersama ibu Dessy Akhriani.

Tabel 4.2 Analisis Permasalahan 2

The problem of	• Perlu adanya konfirmasi ke guru piket ketika terdapat kehadiran siswa yang tidak diketahui. • Pencatatan ketidakhadiran siswa yang menitipkan surat melalui guru piket akan dicatat dua kali yaitu dicatat pada buku piket dan buku presensi di kelas.
Affects	Pengguna dan pelanggan.
The impact of which is	Data kehadiran siswa tidak dapat diketahui secara jelas oleh guru di kelas.
A successful solution would	Sistem yang menyediakan layanan pendataan kehadiran yang dapat diakses oleh guru piket

Tabel 4.2 Analisis Permasalahan 2 (lanjutan)

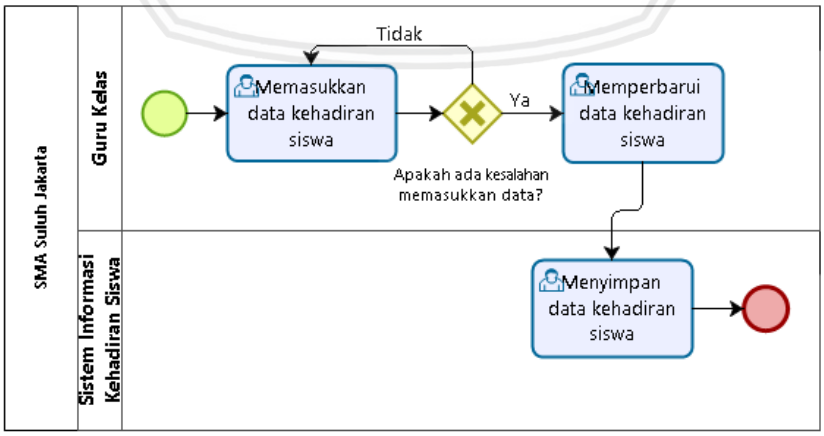
	untuk memperbarui data, sehingga data yang ada di kelas pun menjadi lebih jelas dan terbarui.

4.1.3 Proses Bisnis *To-Be*

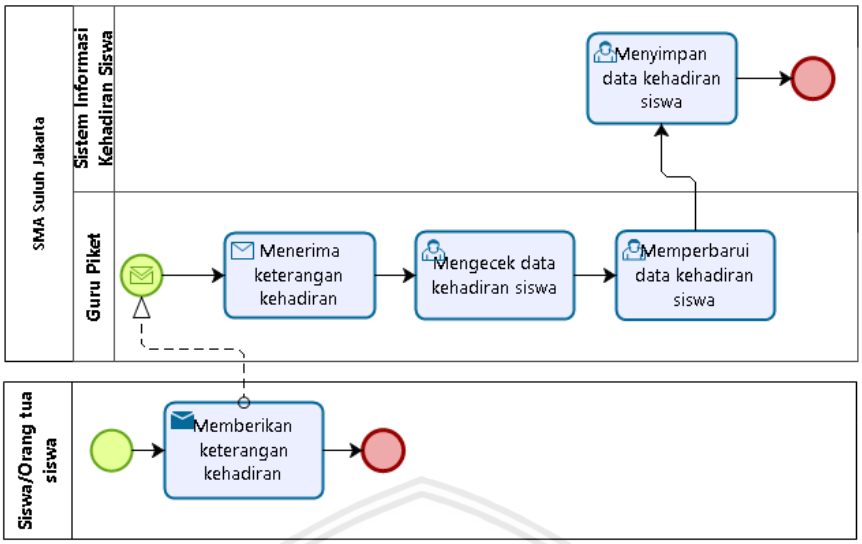
Proses bisnis *to-be* dalam penelitian ini merupakan proses bisnis yang diusulkan oleh peneliti kepada pemangku kepentingan. Proses bisnis *to-be* didapatkan dari hasil identifikasi yang telah dilakukan pada proses bisnis *as-is,* kemudian dari proses bisnis *as-is* tersebut terdapat aktivitas-aktivitas yang diubah atau dieliminasi sesuai dengan kebutuhan pemangku kepentingan. Hasil dari pemodelan proses bisnis *to-be* ini nantinya digunakan untuk analisis pesyaratan.

4.1.3.1 Proses Bisnis To-Be Mendata Kehadiran Siswa dan Merevisi Data Kehadiran Siswa

Proses bisnis *to-be* yang diusulkan kepada pemangku kepentingan terkait dengan proses bisnis *as-is* mendata kehadiran siswa dimodelkan kedalam 2 proses bisnis *to-be* yaitu proses bisnis *to-be* mendata kehadiran siswa dan merevisi data kehadiran siswa. Pada proses bisnis *to-be* mendata kehadiran kehadiran siswa yang ditunjukkan oleh Gambar 4.4, guru kelas dapat memasukkan data kehadiran siswa secara langsung ke dalam *database* melalui Sistem Informasi Kehadiran Siswa. Data yang telah dimasukkan ke dalam sistem oleh guru kelas dapat diperbarui jika terdapat kesalahan dalam memasukkan data dan data dapat diakses oleh unit bisnis lain yang memiliki akses. Sedangkan pada proses bisnis *to-be* merevisi data kehadiran siswa yang ditunjukkan oleh Gambar 4.5, guru piket dapat memperbarui data kehadiran siswa yang sebelumnya telah dimasukkan oleh guru kelas. Dengan adanya Sistem Informasi Kehadiran Siswa, jika terdapat siswa yang terlambat atau tidak hadir guru piket dapat memperbarui secara langsung data yang telah dimasukkan oleh guru kelas, sehingga guru kelas tidak perlu lagi mengonfirmasi kehadiran siswa tersebut kepada guru piket.



Gambar 4.4 Proses Bisnis *To-Be* dari Mendata Kehadiran Siswa



Gambar 4.5 Proses Bisnis *To-Be* dari Merevisi Data Kehadiran Siswa

Pada aktivitas proses bisnis *to-be* mendata kehadiran siswa dan merevisi data kehadiran siswa, terdapat perubahan dan eliminasi aktivitas yang sebelumnya ada pada proses bisnis *as-is* mendata kehadiran siswa. Perubahan aktivitas dari proses bisnis *as-is* mendata kehadiran siswa menjadi aktivitas pada proses bisnis *to-be* mendata kehadiran siswa dan merevisi data kehadiran siswa ditunjukkan oleh Tabel 4.3.

Tabel 4.3 Perubahan Aktivitas Proses Bisnis *As-Is* Mendata Kehadiran Siswa Ke Proses Bisnis *To-Be* Mendata Kehadiran Siswa dan Merevisi Data Kehadiran Siswa

Unit Bisnis	Aktivitas pada Proses Bisnis *As-Is* Mendata Kehadiran	Aktivitas pada Proses Bisnis *To-Be*		Keterangan
		Mendata Kehadiran Siswa	Merevisi Data Kehadiran Siswa	
Guru kelas	Mengecek dan mencatat kehadiran siswa	Memasukkan data kehadiran siswa	-	Diubah
	Memperbarui data kehadiran siswa	Memperbarui data kehadiran siswa	-	-
	Meminta keterangan siswa yang tidak hadir	-	-	Dieliminasi

Tabel 4.3 Perubahan Aktivitas Proses Bisnis *As-Is* Mendata Kehadiran Siswa Ke Proses Bisnis *To-Be* Mendata Kehadiran Siswa dan Merevisi Data Kehadiran Siswa (lanjutan)

Unit Bisnis	Aktivitas pada Proses Bisnis *As-Is* Mendata Kehadiran Siswa	Aktivitas pada Proses Bisnis *To-Be*		Keterangan
		Mendata Kehadiran Siswa	Merevisi Data Kehadiran Siswa	
	Menerima keterangan siswa yang tidak hadir	-	-	Dieliminasi
Sekretaris kelas	Menerima permintaan keterangan siswa yang tidak hadir	-	-	Dieliminasi
	Meminta keterangan siswa yang tidak hadir	-	-	Dieliminasi
	Menerima keterangan siswa yang tidak hadir	-	-	Dieliminasi
	Memberikan keterangan siswa yang tidak hadir	-	-	Dieliminasi
Guru piket	Menerima permintaan keterangan siswa yang tidak hadir	-	Menerima keterangan kehadiran	Diubah
	Mengecek keterangan siswa yang tidak hadir	-	Mengecek data kehadiran siswa	Diubah

Tabel 4.3 Perubahan Aktivitas Proses Bisnis *As-Is* Mendata Kehadiran Siswa Ke Proses Bisnis *To-Be* Mendata Kehadiran Siswa dan Merevisi Data Kehadiran Siswa (lanjutan)

Unit Bisnis	Aktivitas pada Proses Bisnis *As-Is* Mendata Kehadiran Siswa	Aktivitas pada Proses Bisnis *To-Be*		Keterangan
		Mendata Kehadiran Siswa	Merevisi Data Kehadiran Siswa	
	Memberikan keterangan siswa yang tidak hadir	-	Memperbarui data kehadiran siswa	Diubah

Pada aktivitas meminta keterangan siswa yang tidak hadir dan menerima keterangan siswa yang tidak hadir yang dilakukan oleh guru kelas dieliminasi. Serta pada aktivitas menerima permintaan keterangan siswa yang tidak hadir, meminta keterangan siswa yang tidak hadir, menerima keterangan siswa yang tidak hadir, dan memberikan keterangan siswa yang tidak hadir yang dilakukan oleh sekretaris kelas juga dieliminasi. Aktivitas-aktivitas tersebut dieliminasi karena proses mengonfirmasi kehadiran siswa kepada guru piket melalui sekretaris kelas tidak perlu lagi dilakukan, karena guru kelas dapat memperbarui secara langsung data kehadiran siswa yang telah dimasukkan oleh guru kelas ke dalam *database* dengan mengakses Sistem Informasi Kehadiran Siswa.

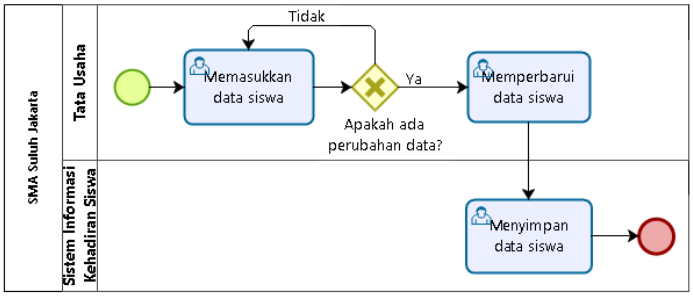
4.1.3.2 Proses Bisnis To-Be Merekap Data Kehadiran

Pada proses bisnis *to-be* merekap data kehadiran terdapat perubahan dari proses bisnis yang sedang berjalan (*as-is*) pada SMA Suluh Jakarta. Dengan adanya penggunaan Sistem Informasi Kehadiran Siswa, perekapan kehadiran siswa tidak perlu dilakukan kembali. Hasil dari data kehadiran siswa yang telah dimasukkan oleh guru kelas dan guru piket akan langsung terekap pada sistem, sehingga ketika kepala sekolah, guru BK, dan wali kelas ingin melihat datanya dapat langsung mengakses Sistem Informasi Kehadiran Siswa.

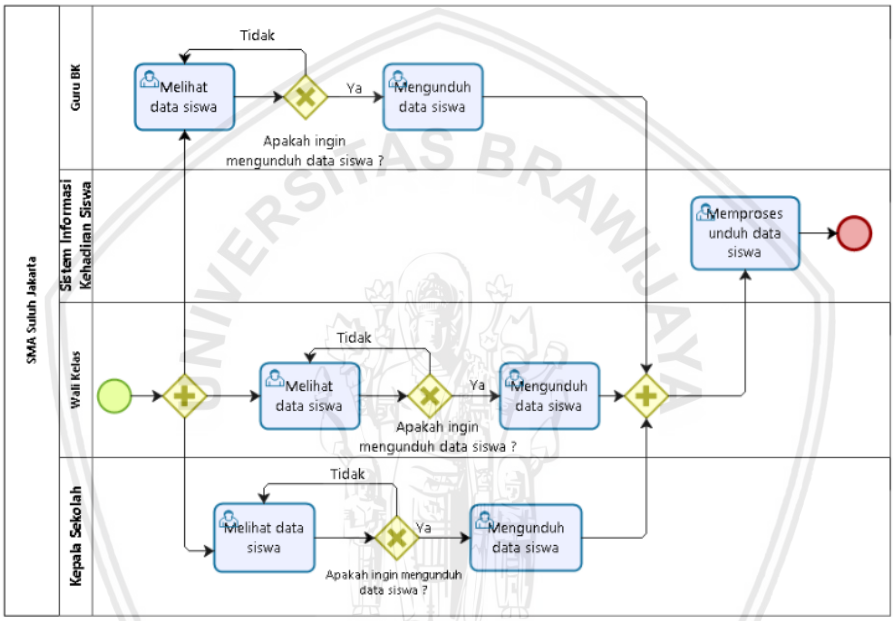
4.1.3.3 Proses Bisnis To-Be Mendata Siswa dan Memonitor Data Siswa

Proses bisnis *to-be* yang diusulkan kepada pemangku kepentingan terkait dengan proses bisnis *as-is* mengelola data siswa dimodelkan kedalam 2 proses bisnis *to-be* yaitu proses bisnis *to-be* mendata siswa dan memonitor data siswa. Pada proses bisnis *to-be* mendata siswa yang ditunjukkan oleh Gambar 4.6, tata usaha dapat memasukkan data siswa ke dalam sistem dan dapat dilakukan perubahan data jika diperlukan, serta data yang ada pada sistem dapat diakses oleh unit bisnis lain yang memiliki akses terhadap data tersebut. Pada proses bisnis *to-be* memonitor data siswa yang ditunjukkan oleh Gambar 4.7, kepala sekolah, wali kelas, dan guru BK dapat mengakses data siswa yang sebelumnya telah

dimasukkan oleh tata usaha kapan pun mereka membutuhkan data tersebut dengan mengakses Sistem Informasi Kehadiran Siswa.



Gambar 4.6 Proses Bisnis *To-Be* dari Mendata Siswa



Gambar 4.7 Proses Bisnis *To-Be* dari Memonitor Data Siswa

Pada aktivitas proses bisnis *to-be* mendata siswa dan memonitor data siswa, terdapat perubahan dan eliminasi aktivitas yang sebelumnya ada pada proses bisnis *as-is* mengelola data siswa. Perubahan aktivitas dari proses bisnis *as-is* mengelola data siswa menjadi aktivitas pada proses bisnis *to-be* mendata siswa dan memonitor data siswa ditunjukkan oleh Tabel 4.4.

Tabel 4.4 Perubahan Aktivitas Proses Bisnis *As-Is* Mengelola Data Siswa Ke Proses Bisnis *To-Be* Mendata Siswa dan Memonitor Data Siswa

Unit Bisnis	Aktivitas pada Proses Bisnis *As-Is* Mendata Siswa	Aktivitas pada Proses Bisnis *To-Be*		Keterangan
		Mendata Siswa	Memonitor Data Siswa	
Tata	Memberikan laporan data	-	-	Dieliminasi

Tabel 4.4 Perubahan Aktivitas Proses Bisnis *As-Is* Mengelola Data Siswa Ke Proses Bisnis *To-Be* Mendata Siswa dan Memonitor Data Siswa (lanjutan)

Unit Bisnis	Aktivitas pada Proses Bisnis *As-Is* Mendata Siswa	Aktivitas pada Proses Bisnis *To-Be*		Keterangan
		Mendata Siswa	Memonitor Data Siswa	
usaha	siswa keseluruhan			
	Mencatat data siswa	Memasukkan data siswa	-	Diubah
		Memperbarui data siswa	-	Diubah
	Memberikan laporan data siswa yang diampu	-	-	Dieliminasi
Kepala sekolah	Menerima laporan data siswa keseluruhan	-	Melihat data siswa	Diubah
		-	Mengunduh data siswa	
Wali kelas	Menerima laporan data siswa yang diampu	-	Melihat data siswa	Diubah
		-	Mengunduh data siswa	
Guru BK	Menerima laporan data siswa yang diampu	-	Melihat data siswa	Diubah
		-	Mengunduh data siswa	

Pada aktivitas memberikan laporan data siswa keseluruhan dan memberikan laporan data siswa yang diampu yang dilakukan oleh tata usaha dieliminasi. Aktivitas dieliminasi karena tata usaha tidak perlu lagi memberikan laporan berupa data siswa kepada kepala sekolah, wali kelas, dan guru BK. Data siswa yang telah dimasukkan oleh tata usaha ke dalam sistem dapat diakses secara langsung oleh kepala sekolah, wali kelas dan guru BK kapanpun mereka membutuhkan data siswa.

4.2 Analisis Persyaratan

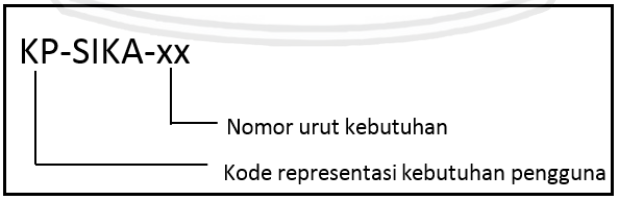
4.2.1 Identifikasi Tipe Pemangku Kepentingan

Identifikasi tipe pemangku kepentingan dilakukan untuk mengelompokkan pemangku kepentingan yang berhubungan dengan sistem berdasarkan tipe pemangu kepentingan. Hasil dari identifikasi tipe pemangku kepentingan, digunakan dalam analisis permasalahan. Sehingga peneliti dapat menganalisis permasalahan yang dihadapi dari setiap pemangku kepentingan yang ada pada SMA Suluh Jakarta. Identifikasi tipe pemangku kepentingan dapat dilihat pada Tabel 4.5.

Tabel 4.5 Identifikasi Tipe Pemangku Kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
Pengguna	Individu yang berperan sebagai aktor pada *use case* dan memiliki interaksi secara langsung dengan Sistem Informasi Kehadiran Siswa.	Kepala sekolah, wali kelas, guru kelas, guru BK, guru Piket, tata usaha
Pengembang	Individu atau organisasi yang melakukan pengembangan terhadap Sistem Informasi Kehadiran Siswa.	Peneliti
Pelanggan	Individu atau organisasi yang mendapat manfaat dengan dikembangkannya Sistem Informasi Kehadiran Siswa.	SMA Suluh Jakarta, siswa

4.2.2 Identifikasi Kebutuhan Pemangku Kepentingan dan pengguna



Gambar 4.8 Kode Identifikasi Kebutuhan Pengguna

Identifikasi kebutuhan pengguna diperoleh dari analisis permasalahan yang telah dilakukan sebelumnya yaitu dengan melakukan wawancara bersama guru BK dan guru piket SMA Suluh Jakarta. Kemudian hasil dari identifikasi kebutuhan pengguna, digunakan untuk menganalisis fitur yang akan ada pada Sistem Informasi Kehadiran Siswa yang merupakan solusi yang ditawarkan oleh peneliti untuk dapat memenuhi kebutuhan dari pengguna. Pada Gambar 4.8

merupakan penjelasan terkait pemberian kode pada identifikasi kebutuhan pengguna.

Pada Tabel 4.6 dijelaskan hasil dari identifikasi kebutuhan pengguna dengan pemberian kode pada setiap kebutuhan, penjelasan kebutuhan pengguna, pemangku kepentingan yang memiliki kaitan dengan kebutuhan pengguna yang telah teridentifikasi, situasi yang saat ini ada dalam pemenuhan kebutuhan pengguna, dan solusi yang ditawarkan untuk pemenuhan kebutuhan pengguna yang telah teridentifikasi.

Tabel 4.6 Hasil Identifikasi Kebutuhan Pengguna

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
KP-SIKA-1	Sistem harus menyediakan layanan untuk mendata kehadiran siswa.	Guru kelas	Pendataan kehadiran siswa dilakukan setiap pagi, jika terdapat kehadiran siswa tidak diketahui maka akan dikosongkan terlebih dahulu, kemudian meminta konfirmasi dari guru piket terkait dengan kehadiran siswa tersebut. Kemudian guru kelas yang akan memperbarui data.	Sistem informasi yang menyediakan layanan untuk mendata kehadiran siswa, dengan data yang dimasukkan tersimpan ke dalam sistem penyimpanan data (*database*) terpusat dan data dapat diperbarui secara langsung oleh pihak yang memiliki akses melalui Sistem Informasi Kehadiran Siswa.
KP-SIKA-2	Sistem harus menyediakan layanan untuk memonitor rekap data kehadiran.	Kepala sekolah, wali kelas, guru BK	Perekapan data dilakukan seminggu sekali dengan meminta data kehadiran setiap kelas kemudian data tersebut	Sistem informasi yang dapat menampilkan hasil rekap data terkait kehadiran siswa dan dokumen dapat disimpan dalam bentuk

Tabel 4.6 Hasil Identifikasi Kebutuhan Pengguna (lanjutan)

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
			dimasukkan kedalam Microsoft Excel.	Pdf.
KP-SIKA-3	Sistem harus menyediakan layanan untuk pengelolaan data siswa.	Tata usaha	Data terkait siswa yang terdaftar sebagai siswa di SMA Suluh Jakarta. Data siswa dapat diperbarui oleh tata usaha dengan mendatanya dengan kertas kemudian merekapnya dengan Microsoft Excel.	Sistem informasi yang menyediakan informasi terkait data siswa dan data dapat diunduh ke dalam bentuk Excel.
KP-SIKA-4	Sistem harus menyediakan layanan untuk memonitor kehadiran siswa secara menyeluruh.	Kepala sekolah, wali kelas, guru BK	Tidak ada	Sistem informasi yang menyediakan informasi terkait kehadiran siswa secara menyeluruh dari tiap tingkatan kelas yaitu kehadiran kelas X, kehadiran kelas XI, kehadiran kelas XII dan kehadiran secara menyeluruh dari siswa SMA

Tabel 4.6 Hasil Identifikasi Kebutuhan Pengguna (lanjutan)

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
				Suluh Jakarta dalam bentuk *pie chart*.
KP-SIKA-5	Informasi yang ada pada sistem hanya dapat diakses sesuai dengan identitas pengguna sistem.	Kepala sekolah, wali kelas, guru kelas, guru BK, guru piket, tata usaha	Tidak ada	Informasi yang disediakan oleh sistem dapat diakses sesuai dengan identitas pengguna saat mengakses sistem.
KP-SIKA-6	Sistem harus menyediakan layanan untuk mengelola data master.	Peneliti	Tidak ada	Sistem informasi yang menyediakan layanan untuk mengelola data kelas.
KP-SIKA-7	Sistem harus menyediakan layanan untuk memperbarui data kehadiran siswa yang telah tersimpan, sehingga data yang ada di kelas dan di guru piket saling terintegrasi.	Guru Piket	Guru kelas akan meminta konfirmasi kepada guru piket melalui sekretaris kelas, jika terdapat kehadiran siswa yang tidak diketahui di kelas. Kemudian guru kelas yang akan memperbarui datanya.	Sistem informasi yang menyediakan layanan untuk memperbarui data kehadiran siswa secara langsung dari data yang sebelumnya telah tersimpan di dalam *database*.
KP-SIKA-8	Sistem harus menyediakan	Kepala sekolah, wali	Data siswa yang telah	Sistem informasi yang

Tabel 4.6 Hasil Identifikasi Kebutuhan Pengguna (lanjutan)

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
	layanan untuk memonitor data siswa.	kelas, guru BK	dikelola oleh tata usaha kemudian dilaporkan atau diberikan kepada pemangku kepentingan.	menyediakan layanan untuk menampilkan dan mengunduh data siswa, kapanpun data tersebut dibutuhkan oleh pemangku kepentingan.

4.2.3 Identifikasi Penguna

Identifikasi pengguna merupakan individu yang nantinya akan berinteraksi secara langsung dengan sistem. Identifikasi pengguna diperoleh dari kelompok-kelompok yang ada pada tipe pemangku kepentingan, yaitu pengguna dan pengembang. Hasil dari identifikasi pengguna digunakan untuk mengetahui aktor yang akan dimodelkan nantinya pada *use case diagram.* Penjelasan terkait identifikasi pengguna dapat dilihat pada Tabel 4.7.

Tabel 4.7 Hasil Identifikasi Pengguna

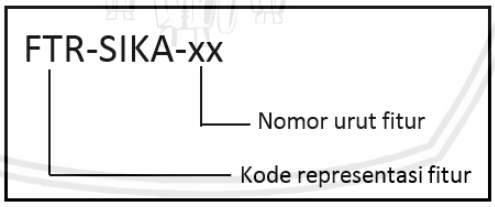
Tipe Pemangku Kepentingan	Tipe Pengguna	Deskripsi
Pengguna	Kepala sekolah	Pegawai SMA Suluh Jakarta yang memiliki tanggung jawab terhadap pihak yayasan Hortikultura, memiliki kedudukan yang tinggi pada SMA Suluh Jakarta, dan memperoleh hasil dari laporan yang telah dibuat dari tiap proses bisnis yang ada.
	Wali kelas	Pegawai SMA Suluh Jakarta yang berprofesi sebagai guru dan bertugas memantau siswa-siswa yang menjadi tanggung jawabnya seperti melihat data siswa dan kehadirannya.
	Guru kelas	Pegawai SMA Suluh Jakarta yang berprofesi sebagai guru dan bertugas mendata kehadiran siswa setiap paginya.
	Guru BK	Pegawai SMA Suluh Jakarta yang berprofesi sebagai guru dan bertugas

Tabel 4.7 Hasil Identifikasi Pengguna (lanjutan)

Tipe Pemangku Kepentingan	Tipe Pengguna	Deskripsi
		memantau siswa-siswa yang menjadi tanggung jawabnya seperti melihat data siswa dan kehadirannya.
	Guru piket	Pegawai SMA Suluh Jakarta yang berprofesi sebagai guru dan bertugas untuk mendata kehadiran siswa yang terlambat, menerima dan mencatat surat siswa yang tidak hadir.
	Tata usaha	Pegawai SMA Suluh Jakarta yang berprofesi sebagai staf dan bertugas untuk mengelola data siswa.
Pengembang	Peneliti	Pengguna yang memiliki akses untuk mengelola data kelas yang ada pada SMA Suluh Jakarta.

4.2.4 Identifikasi Fitur

Solusi yang ditawarkan untuk memenuhi kebutuhan pengguna ditunjukkan dengan bentuk fitur-fitur yang akan terdapat pada Sistem Informasi Kehadiran Siswa. Setiap fitur dalam penelitian ini diberikan kode identitas. Pada Gambar 4.9 merupakan keterangan terkait pemberian kode identitas pada setiap fitur.



Gambar 4.9 Kode Identifikasi dari Fitur

Hasil dari identifikasi fitur ini digunakan untuk pengidentifikasian persyaratan fungsional dan nonfungsional dalam pengembangan Sistem Informasi Kehadiran Siswa. Pada Tabel 4.8 merupakan penjelasan terkait dengan hasil dari identifikasi fitur.

Setelah fitur-fitur telah teridentifikasi dilakukan pengelompokkan berdasarkan pada keterkaitannya dengan kebutuhan-kebutuhan pengguna yang telah teridentifikasi sebelumnya. Penjelasan terkait pengelompokkan fitur berdasarkan kebutuhan pengguna dapat dilihat pada Tabel 4.9.

Tabel 4.8 Hasil Identifikasi Fitur

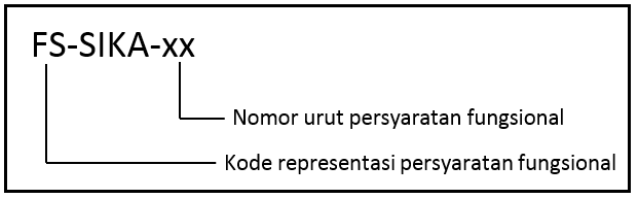
Kode Fitur	Deskripsi
FTR-SIKA-1	Sistem dapat mengenali identitas dari pengguna yang mengakses sistem dan pengguna dapat menerima informasi dari sistem sesuai dengan hak akses yang dimiliki.
FTR-SIKA-2	Sistem dapat digunakan untuk mendata kehadiran siswa yang terdiri dari penyimpanan, penyajian, dan perubahan data kehadiran siswa.
FTR-SIKA-3	Sistem dapat digunakan untuk memonitor rekap data kehadiran siswa yang terdiri dari menampilkan dan mengunduh data kehadiran sesuai kriteria yang dimasukkan.
FTR-SIKA-4	Sistem dapat digunakan untuk mengelola data siswa yang terdiri dari penyajian, penambahan, penghapusan, perubahan dan pengunduhan data siswa.
FTR-SIKA-5	Sistem dapat digunakan untuk memonitor kehadiran siswa secara menyeluruh dengan menggunakan *pie chart*.
FTR-SIKA-6	Sistem dapat digunakan untuk mengelola data terkait data kelas-kelas yang ada di SMA Suluh Jakarta.

Tabel 4.9 Hubungan Fitur dengan Kebutuhan Pengguna

Kebutuhan Pengguna	Fitur
KP-SIKA-5	FTR-SIKA-1
KP-SIKA-1	FTR-SIKA-2
KP-SIKA-7	
KP-SIKA-2	FTR-SIKA-3
KP-SIKA-3	FTR-SIKA-4
KP-SIKA-8	
KP-SIKA-4	FTR-SIKA-5
KP-SIKA-6	FTR-SIKA-6

4.2.5 Persyaratan Fungsional

Persyaratan fungsional merupakan terpenuhinya kondisi-kondisi tertentu di dalam sistem sehingga fitur dapat berjalan dan memenuhi kebutuhan-kebutuhan dari pengguna. Pada Gambar 4.10 merupakan keterangan terkait pemberian kode pada setiap persyaratan fungsional.



Gambar 4.10 Kode Persyaratan Fungsional

Untuk melihat keterkaitan antara fitur dan persyaratan fungsional dapat dilihat pada Tabel 4.10.

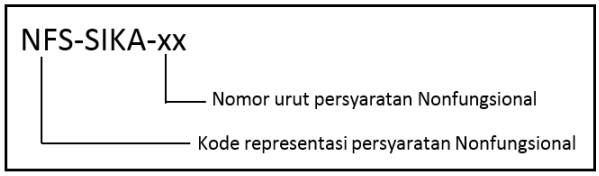
Tabel 4.10 Persyaratan Fungsional

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
FTR-SIKA-1	FS-SIKA-01	Sistem dapat menampilkan halaman untuk autentikasi pengguna.
	FS-SIKA-02	Sistem dapat melakukan autentikasi pengguna yang mengakses sistem berdasarkan pada data pengguna yang telah tersimpan didalam sistem.
	FS-SIKA-03	Sistem dapat menampilkan informasi sesuai dengan tipe dari pengguna yang sedang aktif menggunakan sistem.
	FS-SIKA-04	Sistem dapat melakukan penghentian *session* pada pengguna yang keluar dari sistem.
FTR-SIKA-2	FS-SIKA-05	Sistem dapat menampilkan antarmuka yang berkaitan dengan pengelolaan data kehadiran siswa.
	FS-SIKA-06	Sistem dapat digunakan untuk menyimpan data kehadiran siswa yang telah didata.
	FS-SIKA-07	Sistem dapat menampilkan data kehadiran yang telah dimasukkan.
	FS-SIKA-08	Sistem dapat digunakan untuk mengubah data kehadiran siswa yang telah dimasukkan.
FTR-SIKA-3	FS-SIKA-09	Sistem dapat menampilkan antarmuka yang berkaitan dengan memonitor rekap data kehadiran.
	FS-SIKA-10	Sistem dapat menampilkan rekap data kehadiran berdasarkan rentang tanggal, jumlah hari efektif sekolah, dan kelas yang dipilih.
	FS-SIKA-11	Sistem dapat digunakan untuk mengunduh hasil rekap data kehadiran yang telah ditampilkan

Tabel 4.10 Persyaratan Fungsional (lanjutan)

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
		berdasarkan rentang tanggal, jumlah hari efektif sekolah, dan kelas yang dipilih.
FTR-SIKA-4	FS-SIKA-12	Sistem dapat menampilkan antarmuka yang berkaitan dengan pengelolaan data siswa.
	FS-SIKA-13	Sistem dapat digunakan untuk menampilkan data siswa.
	FS-SIKA-14	Sistem dapat digunakan untuk menambahkan data siswa.
	FS-SIKA-15	Sistem dapat digunakan untuk menghapus data siswa.
	FS-SIKA-16	Sistem dapat digunakan untuk mengubah data siswa.
	FS-SIKA-17	Sistem dapat digunakan untuk mengunduh data siswa.
	FS-SIKA-24	Sistem dapat digunakan untuk menampilkan detail data siswa.
FTR-SIKA-5	FS-SIKA-18	Sistem dapat digunakan untuk menampilkan data kehadiran siswa secara menyeluruh dari setiap tingkatan kelas dengan menggunakan *pie chart*.
FTR-SIKA-6	FS-SIKA-19	Sistem dapat menampilkan antarmuka yang berkaitan dengan pengelolaan data kelas.
	FS-SIKA-20	Sistem dapat digunakan untuk menampilkan data kelas.
	FS-SIKA-21	Sistem dapat digunakan untuk menambah data kelas.
	FS-SIKA-22	Sistem dapat digunakan untuk menghapus data kelas.
	FS-SIKA-23	Sistem dapat digunakan untuk mengubah data kelas.

4.2.6 Persyaratan Nonfungsional



Gambar 4.11 Kode Persyaratan Nonfungsional

Persyaratan nonfungsional merupakan persyaratan yang mendukung berjalannya sistem sehingga perlu adanya pemenuhan persyaratan tersebut. Persyaratan nonfungsional memiliki nomer kodenya masing-masing, pada Gambar 4.11 merupakan keterangan terkait pemberian kode pada setiap persyaratan nonfungsional. Untuk penjelasan terkait identifikasi persyaratan nonfungsional sistem terdapat pada Tabel 4.11.

Tabel 4.11 Persyaratan Nonfungsional

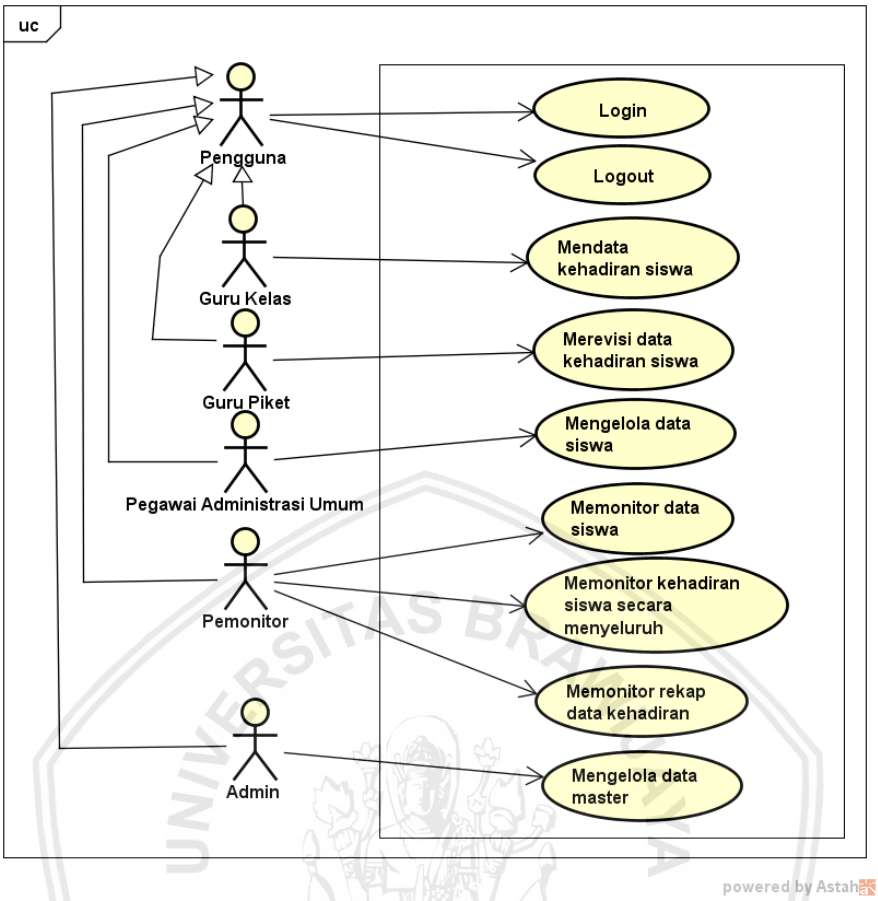
Kode Fitur	Kode Persyaratan Nonfungsional	Deskripsi
FTR-SIKA-1	NFS-SIKA-01	Sistem Informasi Kehadiran Siswa dapat berjalan pada beberapa peramban tanpa ada konten yang hilang.
FTR-SIKA-2		
FTR-SIKA-3		
FTR-SIKA-4		
FTR-SIKA-5		
FTR-SIKA-6		

4.3 Pemodelan *Use Case*

Setelah dilakukan analisis persyaratan, dibuat pemodelan *use case* yang terdiri dari identifikasi aktor, identifikasi *use case,* dan spesifikasi *use case.* Pada tahap Identifikasi aktor berisi mengenai pengguna yang akan menggunakan sistem. Setelah itu pengguna dikelompokan berdasarkan karakteristik yang dimiliki. Pada tahap identifikasi *use case* menjelaskan mengenai tujuan yang ingin dicapai aktor saat menggunakan sistem, identifikasi *use case* didapatkan dari hasil analisis persyaratan. Kemudian setiap *use case* dijelaskan dalam spesifikasi *use case*. Pada spesifikasi *use case* terdapat aktor yang menggunakan sistem, kondisi yang perlu disiapkan sebelum *use case* dapat dilakukan, tahapan alur dasar saat *use case* dilakukan, kondisi jika *use case* berhasil dilakukan dan kondisi jika *use case* gagal dilakukan (alternatif *use case*).

4.3.1 Use Case Diagram

Use case diagram merupakan penggambaran interaksi yang terjadi antara aktor dengan sistem. Aktor pada *use case diagram* diidentifikasi berdasarkan pada pengkelompokan pengguna sesuai dengan peran yang dimiliki. *Use case* diidentifikasi berdasarkan pada tujuan yang ingin dicapai aktor saat menggunakan Sistem Informasi Kehadiran Siswa. Penggambaran *use case diagram* dari Sistem Informasi Kehadiran Siswa ditunjukkan oleh Gambar 4.12.



Gambar 4.12 *Use Case Diagram* Sistem Informasi Kehadiran Siswa

Use case yang telah teridentifikasi kemudian dihubungkan dengan aktivitas pada proses bisnis *to-be* (solusi yang ditawarkan), pemangku kepentingan, dan identifikasi fitur yang sebelumnya telah dilakukan pada tahap analisis persyaratan. Penjelasan terkait hubungan antara *use case* dengan aktivitas-aktivitas pada proses bisnis *to-be* dapat dilihat pada Tabel 4.12.

Tabel 4.12 Hubungan *Use Case* dengan Aktivitas pada Proses Bisnis *To-Be*

Proses Bisnis	Aktivitas	*Use Case*
Mendata kehadiran siswa	Memasukkan data kehadiran siswa	Mendata kehadiran siswa, memonitor rekap data kehadiran, memonitor kehadiran siswa secara menyeluruh
	Menyimpan data kehadiran siswa	
	Memperbarui data kehadiran siswa	
Merevisi data kehadiran siswa	Mengecek data kehadiran siswa	Merevisi data kehadiran siswa
	Memperbarui data kehadiran siswa	
	Menyimpan data kehadiran siswa	
Mendata siswa	Memasukkan data siswa	Mengelola data siswa

Tabel 4.12 Hubungan *Use Case* dengan Aktivitas pada Proses Bisnis *To-Be* (lanjutan)

Proses Bisnis	Aktivitas	*Use Case*
	Memperbarui data siswa	
	Menyimpan data siswa	
Memonitor data siswa	Melihat data siswa	Memonitor data siswa
	Mengunduh data siswa	
	Memproses unduh data siswa	

Kemudian dijelaskan hubungan antara *use case* dengan pemangku kepentingan pada Tabel 4.13, sehingga dapat diketahui interaksi atau aktivitas yang dapat dilakukan oleh pengguna terhadap Sistem Informasi Kehadiran Siswa berdasarkan pengelompokkan tipe pemangku kepentingan.

Tabel 4.13 Hubungan *Use case* dengan Pemangku Kepentingan

Use Case	Pengguna	Tipe Pemangku Kepentingan
Login, Logout	Kepala sekolah, wali kelas, guru BK, guru kelas, guru piket, tata usaha	Pengguna
Mengelola data siswa	Tata usaha	
Mendata kehadiran siswa	Guru kelas	
Merevisi data kehadiran siswa	Guru piket	
Memonitor kehadiran siswa secara menyeluruh	Kepala sekolah, wali kelas, guru BK	
Memonitor data siswa		
Memonitor rekap data kehadiran		
Mengelola data master	Peneliti	Pengembang

Selain itu untuk memastikan bahwa analisis persyaratan yang telah dibuat sesuai dengan pemodelan *use case* yang dilakukan, dijabarkan keterkaitan hubungan antara *use case* dengan fitur yang telah teridentifikasi pada Tabel 4.14.

Tabel 4.14 Hubungan antara *Use Case* dengan Fitur

Use Case	Fitur
Login, Logout	FTR-SIKA-1

Tabel 4.14 Hubungan antara *Use Case* dengan Fitur (lanjutan)

Use Case	Fitur
Memonitor kehadiran siswa secara menyeluruh	FTR-SIKA-5
Mendata kehadiran siswa	FTR-SIKA-2
Merevisi data kehadiran siswa	
Mengelola data siswa	FTR-SIKA-4
Memonitor data siswa	
Memonitor rekap data kehadiran	FTR-SIKA-3
Mengelola data master	FTR-SIKA-6

4.3.2 Deskripsi Aktor

Pada deskripsi aktor dijelaskan mengenai tujuan aktor dalam menggunakan Sistem Informasi Kehadiran Siswa beserta dengan peranan dari aktor dijelaskan dalam deskripsi aktor. Pada Tabel 4.15 merupakan penjelasan terkait deskripsi aktor yang telah teridentifikasi.

Tabel 4.15 Deskripsi Aktor

Nama Aktor	Deskripsi
Pengguna	Aktor yang akan menggunakan sistem secara langsung sesuai dengan hak akses yang dimiliki. Seluruh pegawai yang ada pada SMA Suluh Jakarta akan berperan sebagai pengguna.
Guru Kelas	Aktor menggunakan sistem untuk mendata kehadiran siswa yang diperankan oleh guru kelas SMA Suluh Jakarta.
Guru Piket	Aktor menggunakan sistem untuk merevisi data kehadiran siswa yang telah dimasukkan sebelumnya oleh guru kelas dan diperankan oleh guru piket SMA Suluh Jakarta.
Pegawai Administrasi Umum	Aktor menggunakan sistem untuk mengelola data siswa yang diperankan oleh tata usaha SMA Suluh Jakarta.
Pemonitor	Aktor menggunakan sistem untuk memonitor rekap data kehadiran siswa, memonitor kehadiran siswa secara menyeluruh, dan memonitor data siswa yang diperankan oleh kepala sekolah, wali kelas, guru BK SMA Suluh Jakarta.

Tabel 4.15 Deskripsi Aktor (lanjutan)

Nama Aktor	Deskripsi
Admin	Aktor menggunakan sistem untuk mengelola data kelas yang diperankan oleh peneliti.

4.3.3 Spesifikasi *Use case*

Setiap *use case* yang telah teridentifikasi kemudian dijelaskan kedalam bentuk spesifikasi *use case*. Spesifikasi *use case* menjelaskan mengenai aktor yang menggunakan sistem, kondisi awal yang perlu dipenuhi sebelum *use case* dapat dilakukan, kondisi setelah *use case* dapat dijalankan, tahapan-tahapan aktor saat menjalankan *use case* dan alternatif dari *use case* yang dijalankan.

4.3.3.1 Spesifikasi Use Case Login

Spesifikasi *use case login* menjelaskan mengenai aktor pengguna yang mengakses sistem untuk mendapatkan informasi sesuai dengan hak akses yang dimiliki pada Sistem Informasi Kehadiran Siswa. Pada spesifikasi *use case login* terdapat kondisi-kondisi awal yang perlu dipenuhi agar tahapan-tahapan pada *use case login* dapat dilakukan. Selain itu terdapat juga kondisi jika *use case login* berhasil dijalankan. Penjelasan terkait spesifikasi *use case login* dapat dilihat pada Tabel 4.16.

Tabel 4.16 Spesifikasi *Use Case Login*

Brief Description	*Use case* login menjelaskan mengenai aktor pengguna yang mengakses sistem melakukan autentikasi identitas yang dimiliki untuk mendapatkan informasi yang ada didalam Sistem Informasi Kehadiran Siswa sesuai dengan hak akses yang dimiliki.
Actor	Pengguna
Pre-condition	• Komputer yang digunakan untuk mengakses sistem telah terhubung dengan internet. • Sistem telah terhubung dengan *server.* • Identitas aktor pengguna telah tersimpan di dalam sistem.
Post-condition	• Identitas dari aktor pengguna dapat terverifikasi oleh sistem. • Sistem menampilkan informasi sesuai dengan hak akses yang dimiliki oleh aktor pengguna.
Basic Flow	**{use case dimulai}** 1. *Use case* dimulai ketika aktor pengguna memasukkan *username* dan *password*. 2. Aktor pengguna mengirim *username* dan *password* kedalam sistem. **{mengecek isian}**

Tabel 4.16 Spesifikasi *Use Case Login* (lanjutan)

Basic Flow	3. Sistem mengidentifikasi identitas yang dimasukkan oleh pengguna. **{mengidentifikasi pengguna}** 4. Sistem menverifikasi informasi terkait identitas dari pengguna. **{use case selesai}** 5. *Use case* selesai.
Alternative Flow	**A1. Menangani *Username* atau *Password* Salah.** Pada **{mengidentifikasi pengguna}** di *basic flow*, jika aktor pengguna salah memasukkan *username* dan *password*, makan sistem gagal mengidentifikasi identitas pengguna dan sistem akan menampilkan pesan "Maaf, kombinasi *username* dengan *password* salah" dan *use case* dinyatakan selesai. **A2. Menangani Isian Kosong** Pada **{mengecek isian}** di *basic flow*, jika *username* atau *password* kosong maka akan muncul pesan "Data tidak boleh kosong!" dan *use case* dinyatakan selesai.

4.3.3.2 Spesifikasi Use Case Logout

Spesifikasi *use case logout* menjelaskan mengenai aktor pengguna yang keluar dari sistem setelah aktor pengguna masuk ke dalam sistem. Pada spesifikasi *use case logout* terdapat kondisi-kondisi awal yang perlu dipenuhi agar tahapan-tahapan pada *use case logout* dapat dilakukan. Selain itu terdapat juga kondisi jika *use case logout* berhasil dijalankan. Penjelasan terkait spesifikasi *use case logout* dapat dilihat pada Tabel 4.17.

Tabel 4.17 Spesifikasi *Use Case Logout*

Brief Description	*Use case logout* menjelaskan bagaimana aktor pengguna yang telah masuk ke sistem dapat keluar dari sistem.
Actor	Pengguna
Pre-condition	• Komputer yang digunakan untuk mengakses sistem telah terhubung dengan internet. • Sistem telah terhubung dengan *server.* • Aktor pengguna berhasil masuk ke dalam sistem.
Post-condition	Aktor pengguna berhasil keluar dari sistem.
Basic Flow	**{use case dimulai}** 1. *Use case* dimulai ketika aktor pengguna memilih untuk keluar dari sistem.

Tabel 4.17 Spesifikasi *Use Case Logout* (lanjutan)

Basic Flow	2. Sistem menghentikan hak akses yang dimiliki aktor pengguna. 3. Sistem menampilkan halaman login. **{use case selesai}** 4. *Use case* selesai.
Alternative Flow	-

4.3.3.3 Spesifikasi Use Case Memonitor Kehadiran siswa Secara Menyeluruh

Spesifikasi *use case* memonitor kehadiran siswa secara menyeluruh menjelaskan aktor Pemonitor yang menggunakan sistem untuk melihat data kehadiran siswa secara menyeluruh dari setiap tingkatan kelas berupa tampilan *pie chart*. Pada spesifikasi *use case* memonitor kehadiran siswa secara menyeluruh terdapat kondisi-kondisi awal yang perlu dipenuhi agar tahapan-tahapan pada *use case* memonitor kehadiran siswa secara menyeluruh dapat dilakukan. Selain itu terdapat juga kondisi jika *use case* memonitor kehadiran siswa secara menyeluruh berhasil dijalankan. Penjelasan terkait spesifikasi *use case* memonitor kehadiran siswa secara menyeluruh dapat dilihat pada Tabel 4.18.

Tabel 4.18 Spesifikasi Use Case Memonitor Kehadiran Siswa Secara Menyeluruh

Brief Description	*Use case* memonitor kehadiran siswa secara menyeluruh menjelaskan tentang aktor Pemonitor yang menggunakan sistem untuk melihat data kehadiran siswa secara menyeluruh dari setiap tingkatan kelas berupa tampilan *pie chart*.
Actor	Pemonitor
Pre-condition	• Komputer yang digunakan untuk mengakses sistem telah terhubung dengan internet. • Sistem telah terhubung dengan *server.* • Aktor Pemonitor telah berhasil masuk ke dalam sistem. • Data kelas telah tersimpan ke dalam sistem.
Post-condition	Aktor Pemonitor dapat melihat data kehadiran siswa secara menyeluruh dari setiap tingkatan kelas berupa tampilan *pie chart*.
Basic Flow	**{use case dimulai}** 1. *Use case* dimulai ketika aktor Pemonitor memilih untuk melihat kehadiran siswa secara menyeluruh. 2. Sistem melakukan proses pemuatan data kehadiran siswa secara menyeluruh dari setiap tingkatan kelas berupa tampilan *pie chart*. **{memuat data kehadiran secara menyeluruh}**

Tabel 4.18 Spesifikasi Use Case Memonitor Kehadiran Siswa Secara Menyeluruh (lanjutan)

	3. Sistem menampilkan data kehadiran siswa secara menyeluruh dari setiap tingkatan kelas berupa tampilan *pie chart*. **{use case selesai}** 4. *Use case* selesai.
Alternative Flow	**A1. Menangani Kondisi Data yang Belum Tersedia.** Pada **{memuat data kehadiran secara menyeluruh}** di *basic flow*, jika data kehadiran siswa belum tersedia maka sistem akan menampilkan pesan "Data belum tersedia pada sistem" dan *use case* dinyatakan selesai.

4.3.3.4 Spesifikasi Use Case Mendata Kehadiran Siswa

Spesifikasi *use case* mendata kehadiran siswa menjelaskan mengenai aktor Guru Kelas yang menggunakan sistem untuk menyimpan, melihat dan mengubah data kehadiran siswa. Pada spesifikasi *use case* mendata kehadiran siswa terdapat kondisi-kondisi awal yang perlu dipenuhi agar tahapan-tahapan *use case* mendata kehadiran siswa dapat dilakukan. Selain itu terdapat juga kondisi jika *use case* mendata kehadiran siswa berhasil dijalankan. Penjelasan terkait spesifikasi *use case* mendata kehadiran siswa dapat dilihat pada Tabel 4.19.

Tabel 4.19 Spesifikasi Use Case Mendata Kehadiran Siswa

Brief Description	*Use case* mendata kehadiran siwa menjelaskan tentang aktor Guru Kelas yang menggunakan sistem untuk menyimpan, melihat dan mengubah data kehadiran siswa yang telah tersimpan di dalam sistem.
Actor	Guru Kelas
Pre-condition	• Komputer yang digunakan untuk mengakses sistem telah terhubung dengan internet. • Sistem telah terhubung dengan *server.* • Aktor Guru Kelas berhasil masuk kedalam sistem. • Data kelas telah tersimpan kedalam sistem.
Post-condition	• Sistem dapat berhasil menyimpan data kehadiran siswa. • Aktor Guru Kelas dapat melihat data kehadiran siswa yang telah disimpan.
Basic Flow	**{use case dimulai}** 1. *Use case* dimulai ketika aktor Guru Kelas memilih untuk menyimpan data kehadiran siswa. 2. Sistem menampilkan halaman untuk pilih tanggal dan kelas

Tabel 4.19 Spesifikasi Use Case Mendata Kehadiran Siswa (lanjutan)

Basic Flow	untuk pendataan kehadiran. **{menampilkan halaman pilih kriteria}** 3. Aktor Guru Kelas memilih tanggal dan kelas yang akan didata kehadirannya. **{mengecek isian pilih kriteria}** 4. Aktor Guru Kelas memilih menampilkan halaman untuk pendataan kehadiran siswa. 5. Sistem melakukan proses pemuatan halaman untuk pendataan kehadiran siswa. **{memuat halaman pendataan}** 6. Sistem menampilkan halaman untuk pendataan kehadiran siswa. **{menampilkan halaman pendataan}** 7. Aktor Guru Kelas memasukkan data kehadiran siswa. **{mengonfirmasi simpan data}** 8. Aktor Guru Kelas memilih untuk menyimpan data kehadiran siswa. 9. Sistem melakukan proses menyimpan data kehadiran siswa. **{menyimpan data kehadiran}** 10. Sistem menampilkan pesan "Data berhasil disimpan!" 11. Sistem menampilkan halaman lihat data kehadiran siswa. **{menampilkan halaman lihat data}** 12. Jalankan *subflow* **Melihat Data Kehadiran.** 13. Jalankan *subflow* **Mengubah Data Kehadiran.** **{use case selesai}** 14. *Use case* selesai.
Alternative Flow	**A1. Menangani Isian Kosong** Pada **{mengecek isian pilih kriteria}** di *basic flow* dan *subflow*, jika pilih tanggal atau pilih kelas tidak diisi maka sistem akan menampilkan pesan "Data tidak boleh kosong!" dan *use case* dinyatakan selesai. **A2. Menangani Kondisi Data yang Tidak Tersedia.** Pada **{memuat halaman pendataan}** di *basic flow* jika halaman pendataan kehadiran siswa tidak tersedia maka sistem akan menampilkan pesan "Tidak ada data di *database*" dan *use case* dinyatakan selesai. **A3. Membatalkan Penyimpanan Data.** Pada **{mengonfirmasi simpan data}** di *basic flow,* jika aktor Guru Kelas memilih untuk membatalkan penyimpanan data kehadiran yang telah didata, maka:

Tabel 4.19 Spesifikasi Use Case Mendata Kehadiran Siswa (lanjutan)

	1. Sistem akan menampilkan halaman pilih tanggal dan kelas untuk pendataan kehadiran. 2. *Use case* dinyatakan selesai. **A4. Membatalkan Mengubah Data.** Pada **{memgkonfirmasi ubah data}** di *subflow,* jika aktor Guru Kelas memilih untuk membatalkan perubahan data kehadiran siswa, maka: 1. Sistem akan kembali menampilkan halaman lihat data kehadiran. 2. *Use case* dinyatakan selesai. **A5. Menangani Kegagalan Penyimpan Data.** Pada **{menyimpan data kehadiran}** di *basic flow*, jika sistem mengalami kegagalan dalam penyimpanan data, maka sistem akan menampilkan pesan "Data gagal disimpan!" dan *use case* dinyatakan selesai. **A6. Menangani Kondisi Data yang Belum Tersedia.** Pada **{memuat data kehadiran}** di sub*flow*, jika data kehadiran siswa belum tersedia maka sistem akan menampilkan pesan "Data belum tersedia di sistem" dan *use case* dinyatakan selesai. **A7. Menangani Kegagalan Mengubah Data.** Pada **{memproses perubahan data kehadiran}** di *subflow*, jika sistem mengalami kegagalan dalam mengubah data kehadiran siswa, maka sistem akan menampilkan pesan "Data gagal diubah" dan *use case* dinyatakan selesai. **A8.Melihat Data Kehadiran.** Pada **{menampilkan halaman lihat data}**, jika aktor Guru Kelas memilih melihat data hasil pendataan kehadiran siswa, maka s*ubflow* **Melihat Data Kehadiran** dijalankan. **A9.Mengubah Data Kehadiran.** Pada **{menampilkan halaman lihat data}**, jika aktor Guru Kelas memilih mengubah data hasil pendataan kehadiran siswa, maka s*ubflow* **Mengubah Data Kehadiran** dijalankan.
SubFlow	**S1. Melihat Data Kehadiran.** 1. Aktor Guru Kelas memilih untuk melihat data kehadiran siswa.

Tabel 4.19 Spesifikasi Use Case Mendata Kehadiran Siswa (lanjutan)

	2. Aktor Guru Kelas memilih tanggal dan kelas yang akan dilihat data kehadirannya. **{mengecek isian pilih kriteria}** 3. Aktor Guru Kelas memilih menampilkan halaman data kehadiran siswa yang telah dipilih. 4. Sistem melakukan proses pemuatan data kehadiran siswa. **{memuat data kehadiran}** 5. Sistem menampilkan data kehadiran siswa. **{menampilkan data kehadiran}** 6. *Use case* selesai. **S2. Mengubah Data Kehadiran.** 1. Aktor Guru Kelas memilih mengubah data kehadiran siswa yang telah tersimpan. 2. Aktor Guru Kelas memilih tanggal dan kelas yang akan dilihat data kehadirannya. **{mengecek isian pilih kriteria}** 3. Sistem melakukan proses pemuatan data kehadiran siswa. **{memuat data kehadiran}** 4. Sistem menampilkan data kehadiran siswa. **{menampilkan data kehadiran}** 5. Aktor Guru Kelas mengubah data kehadiran siswa. **{mengonfirmasi ubah data}** 6. Aktor Guru Kelas memilih menyimpan data kehadiran yang telah diubah. 7. Sistem melakukan proses mengubah data kehadiran siswa. **{memproses perubahan data kehadiran}** 8. Sistem menampilkan pesan “Data berhasil diubah!” **{use case selesai}** 9. *Use case* selesai.

4.3.3.5 Spesifikasi Use Case Merevisi Data Kehadiran Siswa

Spesifikasi *use case* merevisi data kehadiran siswa menjelaskan mengenai aktor Guru Piket yang menggunakan sistem untuk mengubah data kehadiran siswa yang telah dimasukkan oleh aktor Guru Kelas dan tersimpan di dalam sistem. Pada spesifikasi *use case* merevisi data kehadiran siswa terdapat kondisi-kondisi awal yang perlu dipenuhi agar tahapan-tahapan *use case* merevisi data kehadiran siswa dapat dilakukan. Selain itu terdapat juga kondisi jika *use case* merevisi data kehadiran siswa berhasil dijalankan. Penjelasan terkait spesifikasi *use case* merevisi data kehadiran siswa dapat dilihat pada Tabel 4.20.

Tabel 4.20 Spesifikasi Use Case Merevisi Data Kehadiran Siswa

Brief Description	*Use case* merevisi data kehadiran siwa menjelaskan tentang aktor Guru Piket yang menggunakan sistem untuk mengubah data kehadiran siswa yang telah tersimpan di dalam sistem.
Actor	Guru Piket
Pre-condition	• Komputer yang digunakan untuk mengakses sistem telah terhubung dengan internet. • Sistem telah terhubung dengan *server.* • Aktor Guru Piket berhasil masuk kedalam sistem. • Data kelas telah tersimpan kedalam sistem.
Post-condition	• Sistem dapat berhasil menyimpan data kehadiran siswa yang telah diubah. • Aktor Guru Piket dapat melihat data kehadiran siswa yang telah disimpan.
Basic Flow	**{use case dimulai}** 1. *Use case* dimulai ketika aktor Guru Piket ingin mengubah data kehadiran siswa. 2. Sistem menampilkan halaman untuk pilih tanggal dan kelas untuk melihat data kehadiran siswa. **{menampilkan halaman pilih kriteria}** 3. Aktor Guru Piket memilih tanggal dan kelas yang akan diubah data kehadirannya. **{mengecek isian pilih kriteria}** 4. Sistem melakukan proses pemuatan data kehadiran siswa. **{memuat data kehadiran}** 5. Sistem menampilkan data kehadiran siswa. **{menampilkan data kehadiran}** 6. Aktor Guru Piket mengubah data kehadiran siswa. **{mengonfirmasi ubah data}** 7. Aktor Guru Piket memilih menyimpan data kehadiran yang telah diubah. 8. Sistem melakukan proses mengubah data kehadiran siswa. **{memproses perubahan data kehadiran}** 9. Sistem menampilkan pesan “Data berhasil diubah!” **{*use case* selesai}** 10. *Use case* selesai.
Alternative Flow	**A1. Menangani Isian Kosong** Pada **{mengecek isian pilih kriteria}** di *basic flow,* jika pilih tanggal atau pilih kelas tidak diisi maka sistem akan menampilkan pesan “Data tidak boleh kosong!” dan *use case* dinyatakan selesai.

Tabel 4.20 Spesifikasi Use Case Merevisi Data Kehadiran Siswa (lanjutan)

Alternative Flow	**A2. Menangani Kondisi Data yang Belum Tersedia.** Pada **{memuat data kehadiran}** di *basic flow*, jika data kehadiran siswa belum tersedia maka sistem akan menampilkan pesan "Data belum tersedia di sistem" dan *use case* dinyatakan selesai. **A3. Membatalkan Mengubah Data.** Pada **{mengonfirmasi ubah data}** di *basic flow,* jika aktor Guru Piket memilih untuk membatalkan perubahan data kehadiran siswa, maka: 1. Sistem akan kembali menampilkan halaman lihat data kehadiran. 2. *Use case* dinyatakan selesai. **A4. Menangani Kegagalan Mengubah Data.** Pada **{memproses perubahan data kehadiran}** di *basic flow*, jika sistem mengalami kegagalan dalam mengubah data kehadiran siswa, maka sistem akan menampilkan pesan "Data gagal diubah" dan *use case* dinyatakan selesai.

4.3.3.6 Spesifikasi Use Case Mengelola Data Siswa

Spesifikasi *use case* mengelola data siswa menjelaskan mengenai aktor Pegawai Administrasi Umum yang menggunakan sistem untuk melihat, menambah, mengubah, menghapus dan mengunduh data siswa. Pada spesifikasi *use case* mengelola data siswa terdapat kondisi-kondisi awal yang perlu dipenuhi agar tahapan-tahapan pada *use case* mengelola data siswa dapat dilakukan. Selain itu terdapat juga kondisi jika *use case* mengelola data siswa berhasil dijalankan. Penjelasan terkait spesifikasi *use case* mengelola data siswa dapat dilihat pada Tabel 4.21.

Tabel 4.21 Spesifikasi *Use Case* Mengelola Data Siswa

Brief Description	*Use case* mengelola data siswa menjelaskan mengenai aktor Pegawai Administrasi Umum yang menggunakan sistem untuk melihat, menambah, mengubah, menghapus, dan mengunduh data siswa yang telah tersimpan di dalam sistem.
Actor	Pegawai Administrasi Umum
Pre-condition	• Komputer yang digunakan untuk mengakses sistem telah terhubung dengan internet. • Sistem telah terhubung dengan *server.* • Aktor Pegawai Administrasi Umum berhasil masuk kedalam sistem. • Data kelas telah tersimpan kedalam sistem.

Tabel 4.21 Spesifikasi *Use Case* Mengelola Data Siswa

Post-condition	• Sistem dapat berhasil menyimpan hasil pengelolaan data siswa. • Aktor Pegawai Administrasi Umum dapat melihat dan mengunduh data siswa yang telah disimpan.
Basic Flow	**{use case dimulai}** 1. *Use case* dimulai ketika aktor Pegawai Administrasi Umum memilih untuk menambah data siswa dengan jumlah yang banyak. 2. Sistem melakukan proses pemuatan halaman data siswa. **{memuat data siswa}** 3. Sistem menampilkan halaman data siswa. **{menampilkan data siswa}** 4. Aktor Pegawai Administrasi Umum memilih tambah banyak data. 5. Sistem menampilkan halaman untuk tambah data banyak siswa. **{menampilkan halaman tambah data banyak siswa}** 6. Aktor Pegawai Administrasi Umum mengunduh format untuk mendata siswa. 7. Aktor Pegawai Administrasi Umum mengisi data siswa pada file format pendataan siswa yang telah diunduh. 8. Aktor Pegawai Administrasi Umum memilih untuk menyimpan data siswa. 9. Aktor Pegawai Administrasi Umum memilih file yang akan diunggah. 10. Aktor Pegawai Administrasi Umum memilih meninjau file yang telah dipilih. **{mengonfirmasi unggah data}** 11. Aktor Pegawai Administrasi Umum memilih mengunggah data. 12. Sistem melakukan proses menyimpan data siswa. **{menyimpan data siswa}** 13. Sistem menampilkan pesan “Data berhasil disimpan!” **{use case selesai}** 14. *Use case* selesai.
Alternative Flow	**A1. Mengubah Data Siswa.** Pada **{menampilkan data siswa}** di *basic flow,* jika aktor Pegawai Administrasi Umum memilih untuk mengubah data siswa, maka lakukan subflow **Mengubah Data Siswa**.

Tabel 4.21 Spesifikasi *Use Case* Mengelola Data Siswa (lanjutan)

	A2. Menghapus Data Siswa. Pada **{menampilkan data siswa}** di *basic flow,* jika aktor Pegawai Administrasi Umum memilih untuk menghapus data siswa, maka: 1. Aktor Pegawai Administrasi Umum memilih untuk menghapus data siswa. 2. Sistem menampilkan pesan "Apakah anda yakin untuk menghapus data ini?" **{mengonfirmasi hapus data}** 3. Aktor Pegawai Administrasi Umum memilih untuk menghapus data siswa yang telah dipilih. 4. Sistem memproses penghapusan data. **{memproses penghapusan data siswa}** 5. Sistem menampilkan pesan "Data berhasil dihapus!" 6. *Use case* selesai. **A3. Membatalkan Menambah Data Siswa.** Pada **{mengonfirmasi simpan data}** di *subflow,* jika aktor Pegawai Administrasi Umum memilih untuk membatalkan menambah data siswa, maka: 1. Sistem akan kembali pada halaman data siswa. 2. *Use case* dinyatakan selesai. **A4. Membatalkan Mengubah Data Siswa.** Pada **{mengonfirmasi ubah data}** di *subflow,* jika aktor Pegawai Administrasi Umum memilih untuk membatalkan mengubah data siswa, maka: 1. Sistem akan kembali pada halaman data siswa. 2. *Use case* dinyatakan selesai. **A5. Membatalkan Menghapus Data Siswa.** Pada **{mengonfirmasi hapus data}** di *alternative flow,* jika aktor Pegawai Administrasi Umum memilih untuk membatalkan menghapus data siswa, maka: 1. Sistem akan kembali pada halaman data siswa. 2. *Use case* dinyatakan selesai. **A6. Menangani Kegagalan Menghapus Data Siswa.** Pada **{memproses penghapusan data siswa}** di *alternative flow*, jika sistem mengalami kegagalan dalam menghapus data siswa, maka sistem akan menampilkan pesan "Data gagal dihapus!" dan *use case* dinyatakan selesai.

Tabel 4.21 Spesifikasi Use Case Mengelola Data Siswa (lanjutan)

	A7. Menambah Data Satu Siswa. Pada **{menampilkan data siswa}** di *basic flow,* jika aktor Pegawai Administrasi Umum memilih untuk menambah data satu siswa, maka lakukan subflow **Menambah Data Satu Siswa**. **A8. Mengunduh Data Siswa.** Pada **{menampilkan data siswa}** di *basic flow,* jika aktor Pegawai Administrasi Umum memilih untuk mengunduh data siswa, maka lakukan subflow **Mengunduh Data Siswa**. **A9. Membatalkan Mengunggah Data Siswa.** Pada **{mengonfirmasi unggah data}** di *basic flow,* jika aktor Pegawai Administrasi Umum memilih untuk membatalkan mengunggah data siswa, maka: 1. Sistem akan kembali pada halaman data siswa. 2. *Use case* dinyatakan selesai. **A10. Menampilkan Detail Data Siswa.** Pada **{menampilkan data siswa}** di *basic flow,* jika aktor Pegawai Administrasi Umum memilih untuk melihat detail data siswa, maka : 1. Aktor Pegawai Administrasi Umum memilih untuk melihat detail data siswa. 2. Sistem menampilkan detail data siswa. 3. *Use case* dinyatakan selesai.
SubFlow	**S1. Mengubah Data Siswa.** 1. Aktor Pegawai Administrasi Umum memilih untuk mengubah data siswa. 2. Sistem melakukan proses pemuatan data siswa yang akan diubah. **{memuat data siswa}** 3. Sistem menampilkan data siswa yang akan diubah. **{menampilkan data siswa}** 4. Aktor Pegawai Administrasi Umum mengubah data siswa yang telah tersimpan. **{mengonfirmasi ubah data}** 5. Aktor Pegawai Administrasi Umum memilih untuk menyimpan data siswa yang telah diubah. 6. Sistem memproses perubahan data untuk dapat disimpan. **{memproses perubahan data siswa}** 7. Sistem menampilkan pesan “Data berhasil diubah!” 8. *Use case* selesai.

Tabel 4.21 Spesifikasi *Use Case* Mengelola Data Siswa (lanjutan)

	S2. Menambah Data Satu Siswa. 1. *Use case* dimulai ketika aktor Pegawai Administrasi Umum memilih untuk menambah data satu siswa. 2. Sistem melakukan proses pemuatan data siswa. **{memuat data siswa}** 3. Sistem menampilkan data siswa. **{menampilkan data siswa}** 4. Aktor Pegawai Administrasi Umum memilih tambah satu data. 5. Sistem menampilkan isian untuk tambah data. **{menampilkan isian tambah data satu siswa}** 6. Aktor Pegawai Administrasi Umum mengisi data siswa. **{mengonfirmasi simpan data}** 7. Aktor Pegawai Administrasi Umum memilih untuk menyimpan data siswa. 8. Sistem melakukan proses menyimpan data siswa. **{menyimpan data siswa}** 9. Sistem menampilkan pesan "Data berhasil disimpan!" 10. *Use case* selesai. **S3. Mengunduh Data Siswa.** 1. *Use case* dimulai ketika aktor Pegawai Administrasi Umum memilih untuk mengunduh data siswa. 2. Sistem melakukan proses pemuatan data siswa. **{memuat data siswa}** 3. Sistem menampilkan data siswa. **{menampilkan data siswa}** 4. Aktor Pegawai Administrasi Umum memilih mengunduh data siswa. 5. Sistem melakukan proses pemuatan halaman mengunduh data siswa. **{memuat halaman mengunduh data siswa}** 6. Sistem menampilkan halaman mengunduh data siswa. **{menampilkan halaman mengunduh data siswa}** 7. Aktor Pegawai Administrasi Umum memilih kelas yang akan diunduh data siswanya. 8. Aktor Pegawai Administrasi Umum memilih untuk mengunduh data siswa. 9. Sistem melakukan proses mengunduh data siswa. 10. *Use case* selesai.

4.3.3.7 Spesifikasi Use Case Memonitor Data Siswa

Spesifikasi *use case* memonitor data siswa menjelaskan mengenai aktor Pemonitor yang menggunakan sistem untuk melihat dan mengunduh data siswa yang telah dimasukkan oleh aktor Pegawai Administrasi Umum ke dalam sistem. Pada spesifikasi *use case* memonitor data siswa terdapat kondisi-kondisi awal yang perlu dipenuhi agar tahapan-tahapan pada *use case* memonitor data siswa dapat dilakukan. Selain itu terdapat juga kondisi jika *use case* memonitor data siswa berhasil dijalankan. Penjelasan terkait spesifikasi *use case* memonitor data siswa dapat dilihat pada Tabel 4.22.

Tabel 4.22 Spesifikasi *Use Case* Memonitor Data Siswa

Brief Description	*Use case* memonitor data siswa menjelaskan mengenai aktor Pemonitor yang menggunakan sistem untuk melihat dan mengunduh data siswa yang telah tersimpan di dalam sistem.
Actor	Pemonitor
Pre-condition	• Komputer yang digunakan untuk mengakses sistem telah terhubung dengan internet. • Sistem telah terhubung dengan *server.* • Aktor Pemonitor berhasil masuk kedalam sistem. • Data kelas telah tersimpan kedalam sistem.
Post-condition	Aktor Pemonitor dapat melihat dan mengunduh data siswa yang telah disimpan
Basic Flow	**{use case dimulai}** 1. *Use case* dimulai ketika aktor Pemonitor memilih untuk melihat data siswa. 2. Sistem melakukan proses pemuatan halaman data siswa. **{memuat data siswa}** 3. Sistem menampilkan halaman data siswa. **{menampilkan data siswa}** **{use case selesai}** 4. *Use case* selesai.
Alternative Flow	**A1. Menampilkan Detail Data Siswa.** Pada **{menampilkan data siswa}** di *basic flow,* jika aktor Pemonitor memilih untuk melihat detail data siswa, maka : 1. Aktor Pemonitor memilih untuk melihat detail data siswa. 2. Sistem menampilkan detail data siswa. 3. *Use case* dinyatakan selesai. **A2. Mengunduh Data Siswa.** Pada **{menampilkan data siswa}** di *basic flow,* jika aktor Pemonitor memilih untuk mengunduh data siswa, maka lakukan subflow **Mengunduh Data Siswa**.

Tabel 4.22 Spesifikasi *Use Case* Memonitor Data Siswa (lanjutan)

SubFlow	**S1. Mengunduh Data Siswa.** 1. *Use case* dimulai ketika aktor Pemonitor memilih untuk mengunduh data siswa. 2. Sistem melakukan proses pemuatan data siswa. **{memuat data siswa}** 3. Sistem menampilkan data siswa. **{menampilkan data siswa}** 4. Aktor Pemonitor memilih mengunduh data siswa. 5. Sistem melakukan proses pemuatan halaman mengunduh data siswa. **{memuat halaman mengunduh data siswa}** 6. Sistem menampilkan halaman mengunduh data siswa. 7. **{menampilkan halaman mengunduh data siswa}** 8. Aktor Pemonitor memilih kelas yang akan diunduh data siswanya. 9. Aktor Pemonitor memilih untuk mengunduh data siswa. 10. Sistem melakukan proses mengunduh data siswa. 11. *Use case* selesai.

4.3.3.8 Spesifikasi Use Case Memonitor Rekap Data Kehadiran

Spesifikasi *use case* memonitor rekap data kehadiran menjelaskan aktor Pemonitor yang menggunakan sistem untuk melihat dan mengunduh rekap data kehadiran siswa. Pada spesifikasi *use case* memonitor rekap data kehadiran terdapat kondisi-kondisi awal yang perlu dipenuhi agar tahapan-tahapan *use case* memonitor rekap data kehadiran dapat dilakukan. Selain itu terdapat juga kondisi jika *use case* memonitor rekap data kehadiran berhasil dijalankan. Penjelasan terkait spesifikasi *use case* memonitor rekap data kehadiran dapat dilihat pada Tabel 4.23.

Tabel 4.23 Spesifikasi *Use Case* Memonitor Rekap Data Kehadiran

Brief Description	*Use case* memonitor rekap data kehadiran siswa menjelaskan tentang aktor Pemonitor yang menggunakan sistem untuk melihat dan mengunduh data kehadiran siswa yang telah tersimpan di dalam sistem.
Actor	Pemonitor
Pre-condition	• Komputer yang digunakan untuk mengakses sistem telah terhubung dengan internet. • Sistem telah terhubung dengan *server.* • Aktor Pemonitor berhasil masuk kedalam sistem. • Data kelas telah tersimpan kedalam sistem.

Tabel 4.23 Spesifikasi *Use Case* Memonitor Rekap Data Kehadiran (lanjutan)

Post-condition	Aktor Pemonitor dapat melihat dan mengunduh data kehadiran siswa.
Basic Flow	**{use case dimulai}** 1. *Use case* dimulai ketika aktor Pemonitor memilih untuk melihat rekap data kehadiran siswa. 2. Sistem menampilkan halaman pilih tanggal, pilih kelas, dan jumlah hari efektif sekolah untuk melihat rekap data kehadiran. **{menampilkan halaman pilih kriteria}** 3. Aktor Pemonitor memilih rentang tanggal rekap data kehadiran siswa, kelas yang ingin dilihat kehadirannya dan memasukkan jumlah hari efektif sekolah. **{mengecek isian pilih kriteria}** 4. Aktor Pemonitor memilih menampilkan rekap data kehadiran siswa. 5. Sistem melakukan proses pemuatan rekap data kehadiran siswa. **{memuat rekap data kehadiran}** 6. Sistem menampilkan rekap data kehadiran siswa. **{menampilkan rekap data kehadiran}** 7. Jalankan *subflow* **Mengunduh Rekap Data Kehadiran.** **{use case selesai}** 8. *Use case* selesai.
Alternative Flow	**A1. Menangani Kondisi Data yang Belum Tersedia.** Pada **{memuat rekap data kehadiran}** pada *basic flow*, jika data kehadiran siswa belum tersedia maka sistem akan menampilkan pesan "Data belum tersedia di sistem" dan *use case* dinyatakan selesai. **A2. Menangani isian Kosong.** Pada **{mengecek isian pilih kriteria}** di *basic flow* dan *subflow*, jika pilih tanggal, pilih kelas atau hari efektif sekolah tidak diisi maka sistem akan menampilkan pesan "Data tidak boleh kosong!" dan *use case* dinyatakan selesai. **A3.Mengunduh Rekap Data Kehadiran.** Pada **{menampilkan rekap data kehadiran}**, jika Pemonitor memilih mengunduh rekap data kehadiran siswa, maka s*ubflow* **Mengunduh Rekap Data Kehadiran** dijalankan.
SubFlow	**S1. Mengunduh Rekap Data Kehadiran.** 1. Aktor Pemonitor memilih untuk mengunduh rekap data

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 4.23 Spesifikasi *Use Case* Memonitor Rekap Data Kehadiran (lanjutan)

	kehadiran siswa. 2. Aktor Pemonitor memilih rentang waktu rekap data kehadiran siswa, kelas yang ingin dilihat kehadirannya dan memasukkan jumlah hari efektif sekolah. **{mengecek isian pilih kriteria}** 3. Aktor Pemonitor memilih mengunduh rekap data kehadiran siswa. 4. Sistem memproses pengunduhan data. 5. *Use case* selesai

4.3.3.9 Spesifikasi Use Case Mengelola Data Master

Spesifikasi *use case* mengelola data *master* menjelaskan mengenai aktor Admin yang menggunakan sistem untuk melihat, menambah, mengubah dan menghapus data kelas. Pada spesifikasi *use case* mengelola data *master* terdapat kondisi-kondisi awal yang perlu dipenuhi agar tahapan-tahapan *use case* mengelola data *master* dapat dilakukan. Selain itu terdapat juga kondisi jika *use case* mengelola data *master* berhasil dijalankan. Penjelasan terkait spesifikasi *use case* mengelola data *master* dapat dilihat pada Tabel 4.24.

Tabel 4.24 Spesifikasi *Use Case* Mengelola Data *Master*

Brief Description	*Use case* mengelola data *master* menjelaskan mengenai aktor Admin yang menggunakan sistem untuk melihat, menambah, mengubah, dan menghapus data kelas.
Actor	Admin
Pre-condition	• Komputer yang digunakan untuk mengakses sistem telah terhubung dengan internet. • Sistem telah terhubung dengan *server.* • Aktor Admin berhasil masuk kedalam sistem.
Post-condition	• Sistem dapat berhasil menyimpan hasil pengelolaan data kelas. • Aktor Admin dapat melihat data kelas.
Basic Flow	**{use case dimulai}** 1. *Use case* dimulai ketika aktor Admin memilih untuk menambah data kelas. 2. Sistem melakukan proses pemuatan data kelas. **{memuat data kelas}** 3. Sistem menampilkan data kelas. **{menampilkan data kelas}** 4. Aktor Admin memilih tambah data. 5. Sistem menampilkan isian untuk menambah data kelas. **{menampilkan isian tambah data kelas}**

Tabel 4.24 Spesifikasi *Use Case* Mengelola Data *Master* (lanjutan)

	6. Aktor Admin memasukkan data kelas. **{mengonfirmasi simpan data}** 7. Aktor Admin memilih untuk menyimpan data kelas. 8. Sistem melakukan proses menyimpan data kelas. **{menyimpan data kelas}** 9. Sistem menampilkan pesan "Data berhasil disimpan!" **{use case selesai}** 10. *Use case* selesai.
Alternative Flow	**A1. Mengubah Data Kelas.** Pada **{menampilkan data kelas}** di basic *flow,* jika aktor Admin memilih untuk mengubah data kelas, maka lakukan *subflow* **Mengubah Data Kelas**. **A2. Menghapus Data Kelas.** Pada **{menampilkan data kelas}** di basic *flow,* jika aktor Admin memilih untuk menghapus data kelas, maka: 1. Aktor Admin memilih untuk menghapus data kelas. 2. Sistem menampilkan pesan "Apakah anda yakin untuk menghapus data ini?" **{mengonfirmasi hapus data}** 3. Aktor Admin memilih untuk menghapus data kelas yang telah dipilih. 4. Sistem melakukan proses penghapusan data kelas. **{memproses penghapusan data kelas}** 5. Sistem menampilkan pesan "Data berhasil dihapus!" 6. *Use case* selesai. **A3. Membatalkan Menambah Data Kelas.** Pada **{mengonfirmasi simpan data}** di *basic flow,* jika aktor Admin memilih membatalkan untuk menambah data kelas, maka: 1. Sistem akan kembali pada halaman data kelas. 2. *Use case* dinyatakan selesai. **A4. Membatalkan Mengubah Data Kelas.** Pada **{mengonfirmasi ubah data}** di *subflow,* jika aktor Admin memilih untuk membatalkan mengubah data kelas, maka: 1. Sistem akan kembali pada halaman data kelas. 2. *Use case* dinyatakan selesai. **A5. Membatalkan Menghapus Data Kelas.** Pada **{mengonfirmasi hapus data}** di *alternative flow,* jika

Tabel 4.24 Spesifikasi *Use Case* Mengelola Data *Master* (lanjutan)

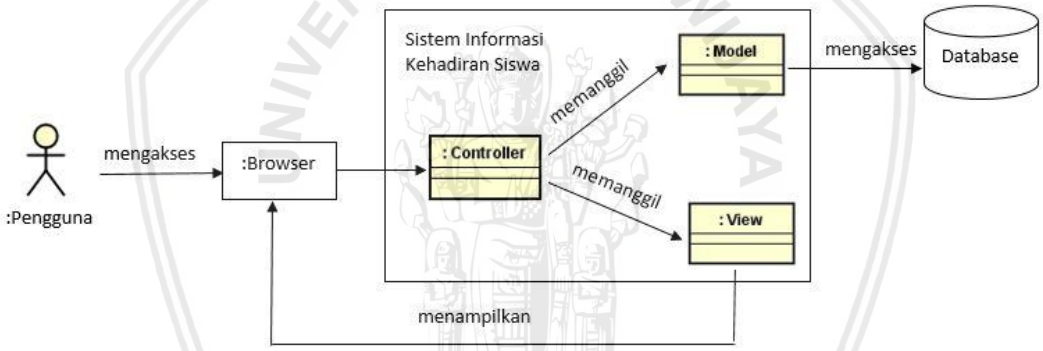
	aktor Admin memilih untuk membatalkan menghapus data kelas, maka: 1. Sistem akan kembali pada halaman data kelas. 2. *Use case* dinyatakan selesai. **A6. Menangani Kegagalan Penyimpan Data Kelas.** Pada **{menyimpan data kelas}** di *basic flow*, jika sistem mengalami kegagalan dalam penyimpanan data kelas, maka sistem akan menampilkan pesan "Data gagal disimpan!" dan *use case* dinyatakan selesai. **A7. Menangani Kegagalan Menghapus Data Kelas.** Pada **{memproses penghapusan data kelas}** di *alternative flow*, jika sistem mengalami kegagalan dalam menghapus data kelas, maka sistem akan menampilkan pesan "Data gagal dihapus!" dan *use case* dinyatakan selesai. **A8. Menangani Kegagalan Mengubah Data Kelas.** Pada **{memproses perubahan data kelas}** di *subflow*, jika sistem mengalami kegagalan dalam mengubah data kelas, maka sistem akan menampilkan pesan "Data gagal diubah!" dan *use case* dinyatakan selesai.
Subflow	**S1. Mengubah Data Kelas.** 1. Aktor Admin memilih untuk mengubah data kelas. 2. Sistem melakukan proses pemuatan data kelas yang akan diubah. **{memuat data kelas}** 3. Sistem menampilkan data kelas yang akan diubah. **{menampilkan data kelas}** 4. Aktor Admin mengubah data kelas yang telah tersimpan. **{mengonfirmasi ubah data}** 5. Aktor Admin memilih untuk menyimpan data kelas yang telah diubah. 6. Sistem memproses perubahan data untuk dapat disimpan. **{memproses perubahan data kelas}** 7. Sistem menampilkan pesan "Data berhasil diubah!" 8. *Use case* selesai.

BAB 5 PERANCANGAN

Bab ini menjelaskan mengenai pemodelan terhadap analisis pesyaratan dengan menggunakan UML (*Unified Modeling Language*). Hasil dari perancangan Sistem Informasi Kehadiran Siswa yang dilakukan berupa perancangan arsitektur sistem, pemodelan objek, pemodelan interaksi antar objek, perancangan basis data, perancangan algoritme, dan perancangan antar muka pengguna.

5.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem menjelaskan mengenai komponen-komponen yang menyusun sistem, serta hubungan antar komponen-komponen tersebut. Dengan adanya perancangan arsitektur sistem dapat membantu pemangku kepentingan dalam memahami bagaimana komponen-komponen yang ada pada Sistem Informasi Kehadiran Siswa dapat membantu mereka dalam melakukan aktivitas bisnis mereka. Perancangan arsitektur Sistem Informasi Kehadiran Siswa dapat dilihat pada Gambar 5.1.



Gambar 5.1 Arsitektur Sistem

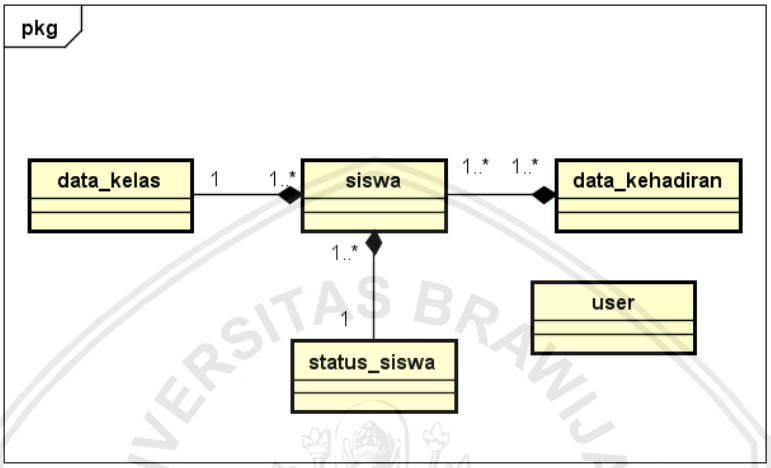
Seperti yang dapat dilihat pada Gambar 5.1 Sistem Informasi Kehadiran Siswa dikembangkan dengan menggunakan konsep MVC (*Model View Controller*). Kerangka kerja yang digunakan untuk perancangan sistem menggunakan kerangka kerja Codeigniter. Ketika pengguna mengakses Sistem Informasi Kehadiran Siswa melalui *browser* maka *controller* akan memanggil model untuk mengakses data yang tersimpan di dalam *database* dan memanggil *view* untuk menampilkan tampilan yang akan dlihat oleh pengguna.

5.2 Pemodelan Objek

Pada pemodelan objek berisi kelas-kelas yang nantinya akan digunakan dalam pengembangan Sistem Informasi Kehadiran Siswa. Pemodelan terhadap kelas-kelas tersebut digambarkan dengan menggunakan diagram kelas. Diagram kelas yang akan dijelaskan pada bagian ini berupa diagram kelas analisis dan diagaram kelas perancangan.

5.2.1 Diagram Kelas Analisis

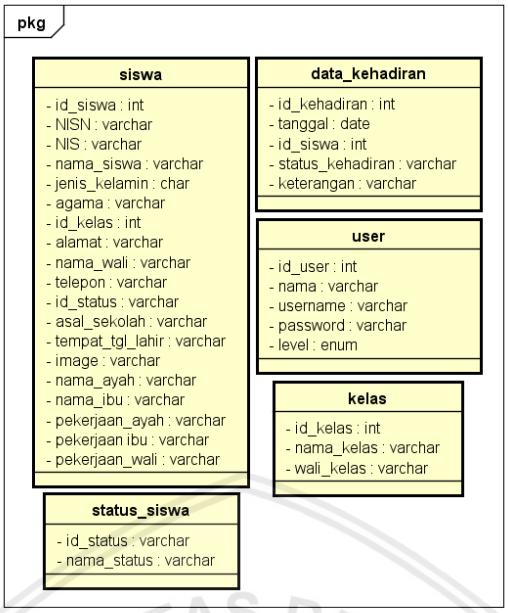
Diagram kelas analisis merupakan objek yang menjadi komponen penyusun sistem yang digambarkan menggunakan relasi antar kelas. Diagram kelas menggambarkan kelas-kelas yang terdapat ada sistem dan relasi diantara kelas-kelas tersebut. Kelas-kelas yang digambarkan merupakan hasil dari *use case diagram* yang telah dilakukan pada tahap sebelumnya. Berikut ini adalah penggambaran diagram kelas analisis yang dapat dilihat pada Gambar 5.2.



Gambar 5.2 Diagram Kelas Analisis

Pada Gambar 5.2 menunjukkan 5 tabel dengan 4 tabel yang saling berhubungan. Hubungan antar kelas-kelas yang pada sistem digambarkan dengan menggunakan komposit untuk menunjukkan hubungan antar kelas dengan tingkat ketergantungan yang kuat. Kelas status_siswa memiliki hubungan komposit dengan kelas siswa yang menggambarkan objek status dari setiap siswa. Hubungan komposit juga terdapat pada kelas siswa dengan kelas data_kelas dan kelas data_kehadiran. Angka 1 pada hubungan komposit data_kelas dan siswa menunjukkan siswa hanya dapat memiliki satu data_kelas dan tanda 1..* menunjukkan data_kelas dapat dimiliki oleh banyak siswa.

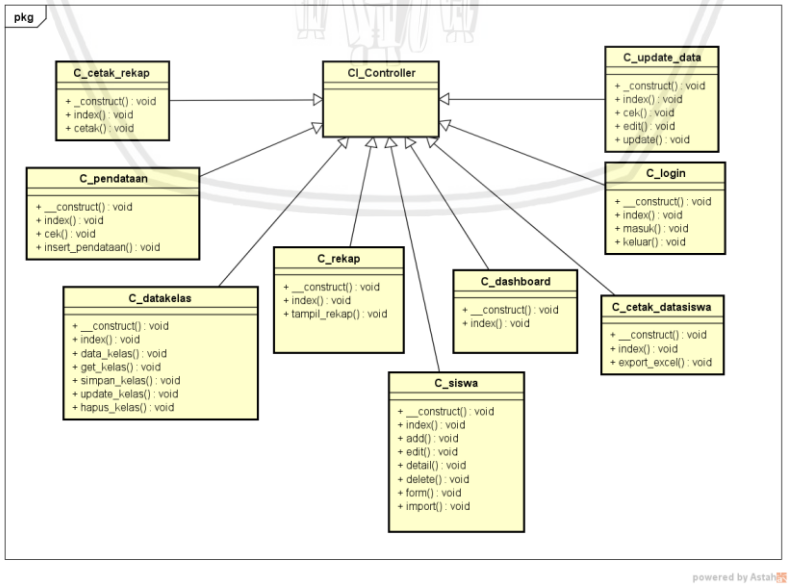
Setiap kelas pada diagram kelas memiliki atributnya masing-masing. Atribut menggambarkan karakteristik dari setiap objek yang ada pada kelas dan setiap atribut memiliki tipe datanya masing-masing. Tipe data dari atribut terletak disebelah kanan dari nama atributnya. Pada Gambar 5.3 menunjukkan atribut-atribut yang terdapat pada setiap kelas yang ada pada diagram kelas.



Gambar 5.3 Atribut pada Diagram Kelas Analisis

5.2.2 Diagram Kelas Perancangan

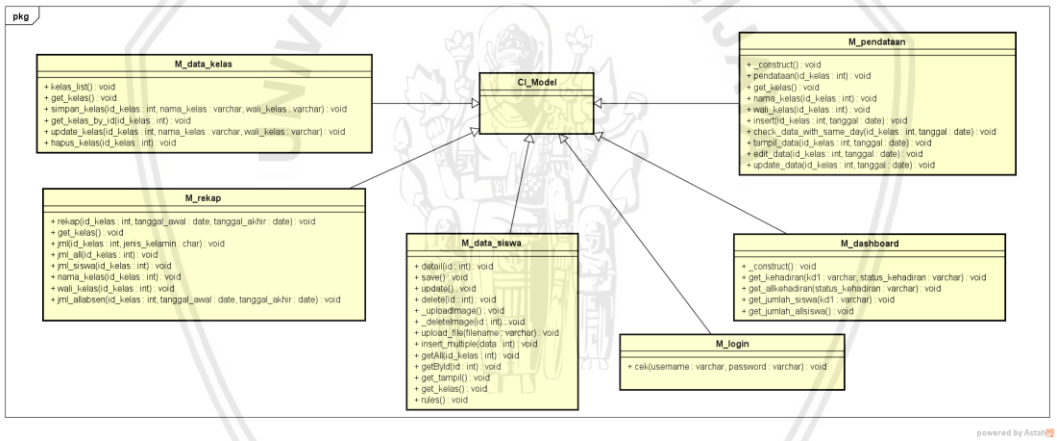
Diagram kelas perancangan merupakan tahapan yang dilakukan sebagai hasil dari diagram kelas analisis yang telah dilakukan pada tahap sebelumnya. Diagram kelas perancangan digambarkan sesuai dengan diagram kelas analisis yang telah teridentifikasi dengan menggunakan pola kerangka keja Codeigniter yaitu MVC. Diagram kelas perancangan memiliki CI_Model dan CI_Controller sebagai kelas klasifikasi umumnya (*superclass)*.



Gambar 5.4 Diagram Kelas Perancangan *Controller*

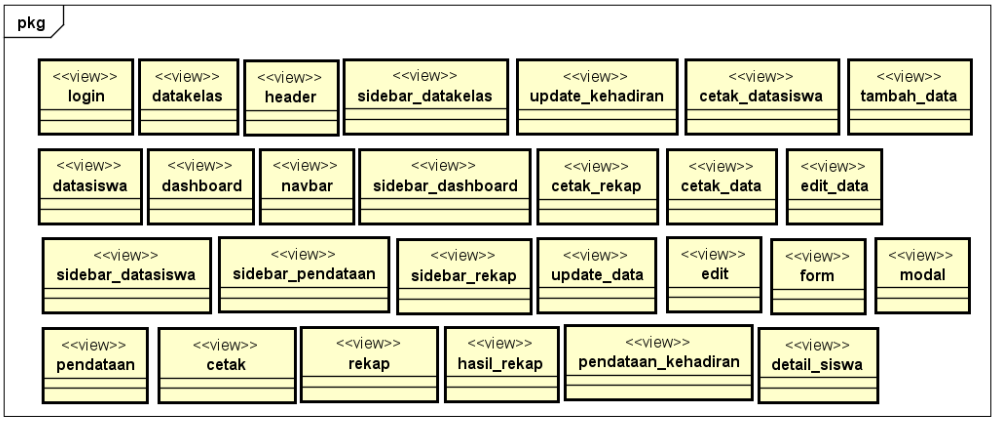
Relasi antar kelas pada *controller* ditunjukkan oleh Gambar 5.4. Pada diagram kelas perancangan *controller* terdapat 9 subkelas yang merupakan

turunan dari kelas klasifikasi umumnya (*superclass*) yaitu CI_Controller. Pada subkelas *controller* memiliki fungsi logikanya masing-masing yang digunakan untuk memenuhi kebutuhan pengguna berdasarkan pada Tabel 4.10 yang menjelaskan mengenai persyaratan fungsional sistem. Pada subkelas *controller* C_login berisi fungsi yang digunakan untuk mengakses sistem agar mendapatkan informasi yang ada pada sistem sesuai dengan hak akses yang dimiliki. Fungsi yang digunakan untuk melihat data kehadiran siswa secara menyeluruh menggunakan subkelas *controller* C_dashboard. Fungsi yang digunakan untuk menyimpan data kehadiran siswa menggunakan subkelas *controller* C_pendataan, sedangkan untuk mengubah serta menampilkan data kehadiran siswa menggunakan subkelas *controller* C_update_data. Fungsi yang digunakan untuk mengelola data siswa menggunakan subkelas *controller* C_siswa. Fungsi yang digunakan untuk mencetak data siswa menggunakan subkelas *controller* C_cetak_datasiswa. Fungsi yang digunakan untuk mengelola data kelas menggunakan subkelas *controller* C_datakelas. Pada subkelas *controller* C_rekap berisi fungsi yang digunakan untuk melihat data kehadiran siswa sesuai rentang tanggal yang dipilih sedangkan untuk mencetak data kehadiran siswa sesuai rentang tanggal yang dipilih menggunakan fungsi pada subkelas *controller* C_cetak_rekap.



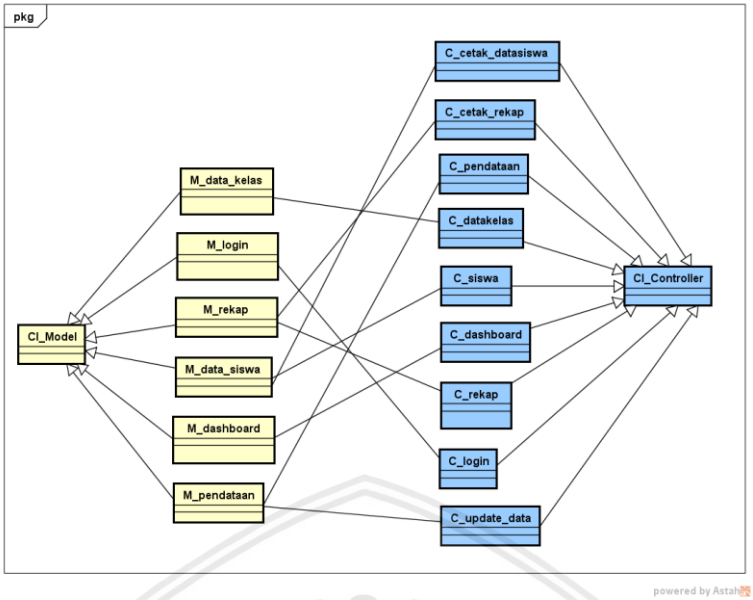
Gambar 5.5 Diagram Kelas Perancangan *Model*

Pada Gambar 5.5 menunjukkan relasi antar kelas *model* dengan 6 subkelas yang memiliki akses ke *database*. Subkelas *model* merupakan turunan dari CI_Model yang merupakan kelas dengan spesifikasi umumnya (*superclass*). Setiap subkelas *model* memiliki fungsi masing-masing. Subkelas *model* M_data_kelas memiliki fungsi yang digunakan untuk mengakses data di *database* untuk mengelola data kelas, M_rekap memiliki fungsi untuk mengakses data di *database* yang digunakan untuk menampilkan dan mengunduh data kehadiran siswa, M_data_siswa memiliki fungsi untuk mengakses data di *database* yang digunakan untuk mengelola data siswa, M_login memiliki fungsi untuk mengakses data di *database* yang digunakan untuk memverifikasi pengguna yang mengakses sistem, M_dashboard memiliki fungsi untuk mengakses data di *database* yang digunakan untuk menampilkan data kehadiran siswa secara menyeluruh, M_pendataan memiliki fungsi untuk mengakses data di *database* yang digunakan untuk menyimpan, menampilkan dan mengubah data kehadiran siswa.



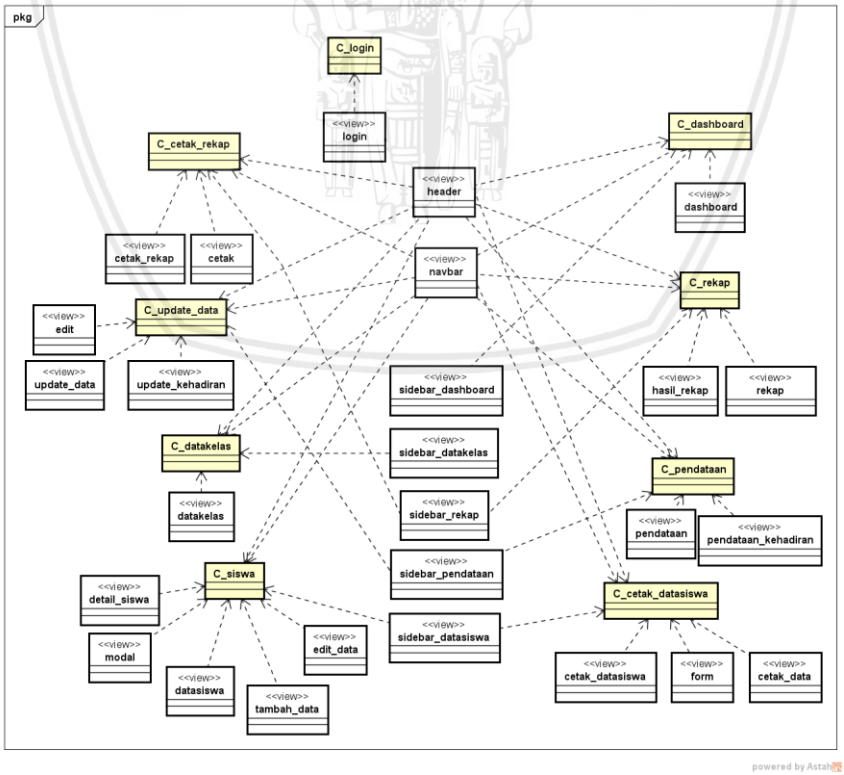
Gambar 5.6 Diagram Kelas Perancangan *Stereotype View*

Pada Gambar 5.6 menunujukkan diagram kelas perancangan *stereotype view* yang terdiri dari 27 kelas yang menggambarkan *view* yang nantinya akan ditampilkan kepada pengguna. Penamaan file *view* berdasarkan pada perancangan *stereotype view*. Pada *view* header, navbar, sidebar_datakelas, sidebar_dashboard, sidebar_datasiswa, sidebar_pendataan, sidebar_rekap menggambarkan tampilan layout dalam sistem. Pada *view* login digunakan untuk tampilan login. Pada *view* datakelas digunakan untuk tampilan mengelola data kelas. Pada *view* datasiswa, tambah_data, edit_data, detail_siswa, dan modal digunakan untuk tampilan mengelola data siswa. Pada *view* dashboard digunakan untuk tampilan data kehadiran siswa secara menyeluruh berupa *pie chart*. Pada *view* cetak_rekap digunakan untuk tampilan mengunduh data kehadiran siswa sesuai tanggal dan kelas yang dipilih. Pada *view* update_data digunakan untuk tampilan data kehadiran siswa yang telah didata. Pada *view* edit digunakan untuk tampilan mengubah data kehadiran siswa. Pada *view* pendataan digunakan untuk tampilan pilih tanggal dan kelas yang akan didata kehadiran siswanya. Pada *view* cetak digunakan untuk tampilan pilih tanggal, kelas dan hari efektif sekolah yang akan dicetak data kehadiran siswanya. Pada *view* rekap digunakan untuk tampilan pilih tanggal, kelas dan hari efektif sekolah yang akan ditampilkan data kehadiran siswanya. Pada *view* hasil_rekap digunakan untuk tampilan data kehadiran siswa sesuai tanggal dan kelas yang dipilih. Pada *view* pendataan_kehadiran digunakan untuk tampilan pendataan kehadiran siswa sesuai tanggal dan kelas yang dipilih. Pada *view* cetak_data digunakan untuk tampilan mengunduh data siswa sesuai kelas yang dipilih. Pada *view* cetak_datasiswa digunakan untuk tampilan pilih kelas dari data siswa yang akan diunduh. Pada *view form* digunakan untuk tampilan data siswa yang akan disimpan di *database*.



Gambar 5.7 Diagram Kelas Perancangan Relasi *Model* dan *Controller*

Relasi antar kelas *controller* dengan kelas *model* ditunjukkan oleh Gambar 5.7. Pemberian nama pada kelas *controller* menggunakan huruf "C" sebagai awalan huruf pada kelas sedangkan untuk pemberian nama pada kelas *model* menggunakan huruf "M" sebagai awalan huruf pada kelas. Masing-masing subkelas merupakan turunan dari kelas dengan spesifikasi umumnya (*superclass*).



Gambar 5.8 Diagram Kelas Perancangan Relasi *Controller* dan *View*

Relasi antara kelas *controller* dengan kelas *view* ditunjukkan oleh Gambar 5.8 dengan 9 subkelas *controller* dan 27 kelas *view* didalamnya. Penggambaran relasi antara kelas *view* dan *controller* sesuai dengan yang telah teridentifikasi sebelumnya.

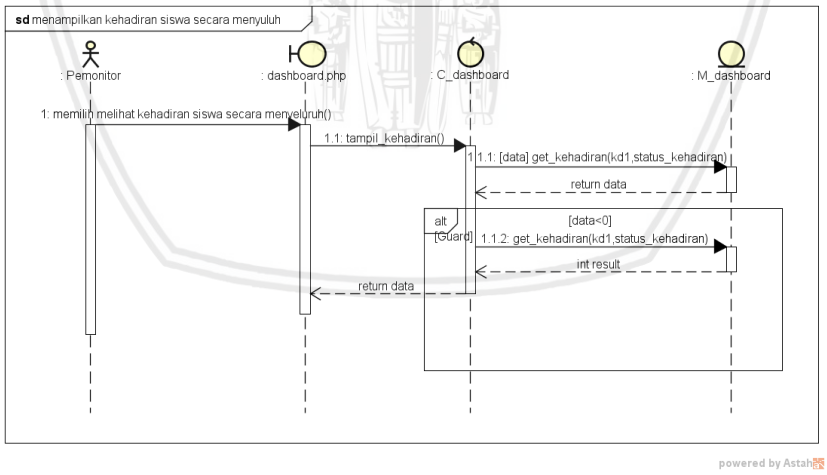
5.3 Pemodelan Interaksi Antar Objek

Pemodelan terhadap interaksi antar objek digambarkan dengan menggunakan *sequence diagram*. *Sequence diagram* menggambarkan interaksi yang terjadi antar objek-objek yang ada di sistem untuk menyampaikan *message* pada suatu proses tertentu. *Sequence diagram* dibuat berdasarkan spesifikasi *use case* yang telah dibuat sebelumnya pada tahap analisis persyaratan. Berdasarkan *use case* yang telah dibuat sebelumnya, akan dijelaskan beberapa *sequence diagram* yang terdapat pada Sistem Informasi Kehadiran Siswa yaitu memonitor kehadiran siswa secara menyeluruh, mendata kehadiran siswa, mengelola data siswa, memonitor rekap data kehadiran, dan mengelola data master.

5.3.1 Memonitor Kehadiran Siswa Secara Menyeluruh

Pada proses memonitor kehadiran siswa secara menyeluruh terdapat *sequence diagram* menampilkan kehadiran siswa secara menyeluruh yang menggambarkan interaksi aktor Pemonitor dengan *model, view, dan controller* untuk dapat menampilkan data kehadiran siswa dari setiap tingkatan kelas secara menyeluruh.

5.3.1.1 Menampilkan Data Kehadiran Siswa Secara Menyeluruh



Gambar 5.9 *Sequence Diagram* Menampilkan Data Kehadiran Siswa Secara Menyeluruh

Pada sequence diagram menampilkan data kehadiran secara menyeluruh yang ditunjukkan pada Gambar 5.9, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menampilkan data kehadiran siswa secara menyeluruh dari tiap tingkatan kelas. Interaksi dimulai ketika aktor Pemonitor mengakses sistem untuk menampilkan kehadiran siswa secara menyeluruh melalui *view* dashboard.php, maka *controller* C_dashboard akan mengirim

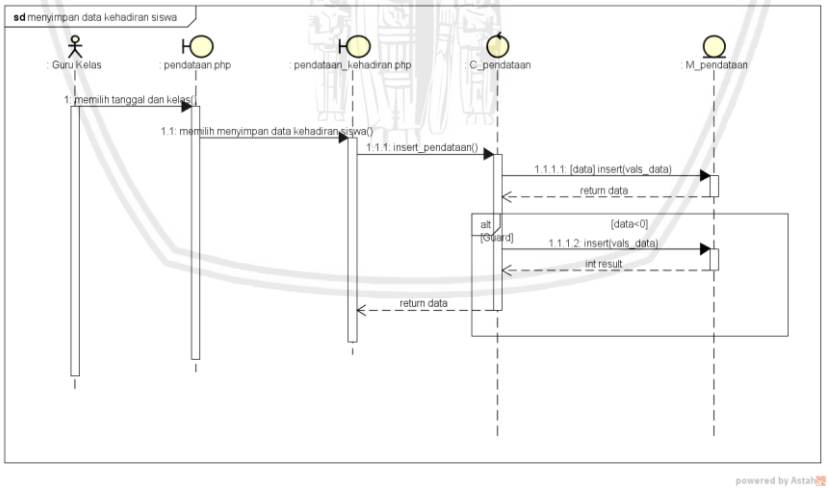
message get_kehadiran() pada *model* M_dashboard untuk meminta data yang akan ditampilkan di dashboard.php. Setelah itu *model* M_dashboard mengirim *message* ke *controller* C_dashboard berupa data yang diminta oleh *controller* C_dashboard dan *controller* C_dashboard mengirim *message* tersebut ke *view* dashboard.php untuk dapat ditampilkan dan dilihat oleh aktor Pemonitor.

5.3.2 Mendata Kehadiran Siswa

Pada proses mendata kehadiran siswa terdapat *sequence diagram* menyimpan data kehadiran, menampilkan data kehadiran, dan mengubah data kehadiran siswa. Setiap *sequence diagram* pada proses mendata kehadiran siswa menggambarkan interaksi aktor Guru Kelas dengan *model, view, dan controller* untuk dapat mendata kehadiran siswa.

5.3.2.1 Sequence Diagram Menyimpan Data Kehadiran

Pada sequence diagram menyimpan data kehadiran yang ditunjukkan pada Gambar 5.10, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menyimpan data hasil pendataan kehadiran siswa. Interaksi dimulai ketika aktor Guru Kelas mengakses sistem untuk menyimpan data kehadiran siswa melalui *view* pendataan.php dan pendataan_kehadiran.php, maka *controller* C_pendataan akan mengirim *message* insert() pada *model* M_pendataan untuk menyimpan data di *database*. Setelah itu *model* M_pendataan mengirim *message* ke *controller* C_pendataan berupa data yang telah tersimpan dan *controller* C_pendataan mengirim *message* tersebut ke *view* pendataan_kehadiran.php untuk dapat ditampilkan dan dilihat oleh aktor Guru Kelas.

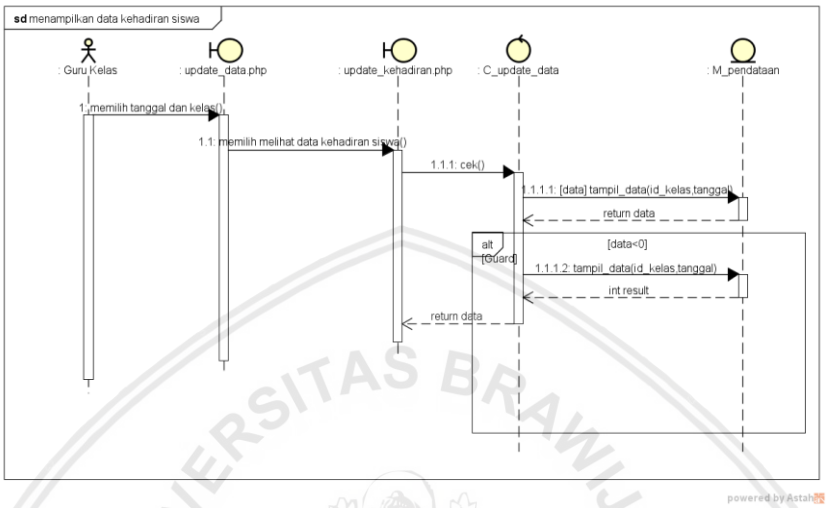


Gambar 5.10 *Sequence Diagram* Menyimpan Data Kehadiran

5.3.2.2 Sequence Diagram Menampilkan Data Kehadiran

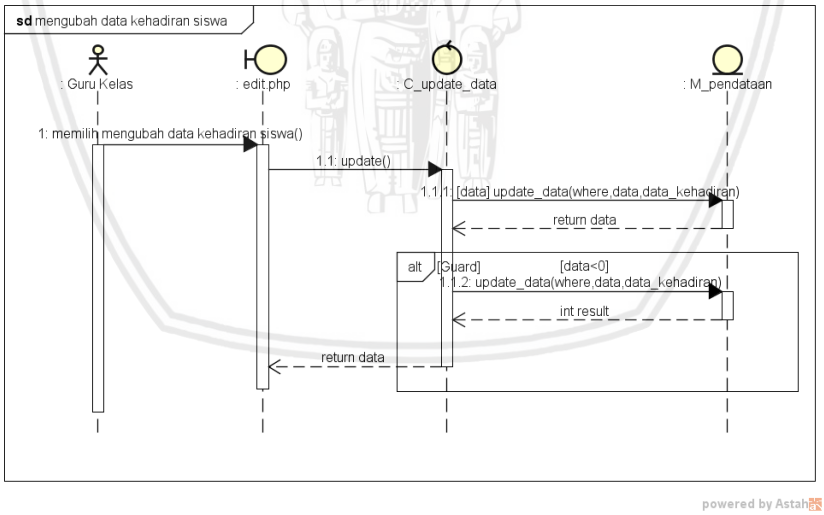
Pada sequence diagram menampilkan data kehadiran yang ditunjukkan pada Gambar 5.11, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menampilkan data kehadiran siswa yang telah didata. Interaksi dimulai ketika aktor Guru Kelas mengakses sistem untuk menampilkan data kehadiran siswa melalui *view* update_data.php dan update_kehadiran.php,

maka *controller* C_update_data akan mengirim *message* tampil_data() pada *model* M_pendataan untuk meminta data yang akan ditampilkan di update_kehadiran.php. Setelah itu *model* M_pendataan mengirim *message* ke *controller* C_update_data berupa data yang diminta oleh *controller* C_update_data dan *controller* C_update_data mengirim *message* tersebut ke *view* update_kehadiran.php untuk dapat ditampilkan dan dilihat oleh aktor Guru Kelas.



Gambar 5.11 *Sequence Diagram* Menampilkan Data Kehadiran

5.3.2.3 Sequence Diagram Mengubah Data Kehadiran



Gambar 5.12 *Sequence Diagram* Mengubah Data Kehadiran

Pada sequence diagram mengubah data kehadiran yang ditunjukkan pada Gambar 5.12, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk mengubah data kehadiran siswa yang telah tersimpan di *database*. Interaksi dimulai ketika aktor Guru Kelas mengakses sistem untuk mengubah data kehadiran siswa melalui *view* edit.php, maka *controller* C_update_data akan mengirim *message* update_data() pada *model* M_pendataan untuk mengubah data yang ada di *database*. Setelah itu *model* M_pendataan mengirim *message* ke

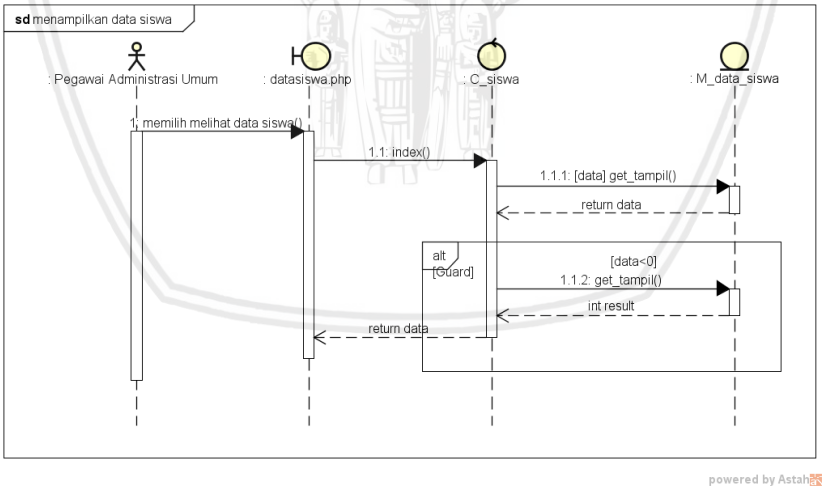
controller C_update_data berupa data yang telah diubah dan *controller* C_update_data mengirim *message* tersebut ke *view* edit.php untuk dapat ditampilkan dan dilihat oleh aktor Guru Kelas.

5.3.3 Mengelola Data Siswa

Pada proses mengelola data siswa terdapat *sequence diagram* menampilkan data siswa, menambah data siswa, menghapus data siswa, mengubah data siswa, mengunduh data siswa, dan menampilkan detail data siswa. Setiap *sequence diagram* pada proses mengelola data siswa menggambarkan interaksi aktor Pegawai Administrasi Umum dengan *model, view,* dan *controller* untuk dapat mengelola data siswa.

5.3.3.1 Sequence Diagram Menampilkan Data Siswa

Pada sequence diagram menampilkan data siswa yang ditunjukkan pada Gambar 5.13, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menampilkan data siswa. Interaksi dimulai ketika aktor Pegawai Administrasi Umum mengakses sistem untuk menampilkan data siswa melalui *view* datasiswa.php, maka *controller* C_siswa akan mengirim *message* get_tampil() pada *model* M_data_siswa untuk meminta data yang akan ditampilkan di datasiswa.php. Setelah itu *model* M_data_siswa mengirim *message* ke *controller* C_siswa berupa data yang diminta oleh *controller* C_siswa dan *controller* C_siswa mengirim *message* tersebut ke *view* datasiswa.php untuk dapat ditampilkan dan dilihat oleh aktor Pegawai Administrasi Umum.

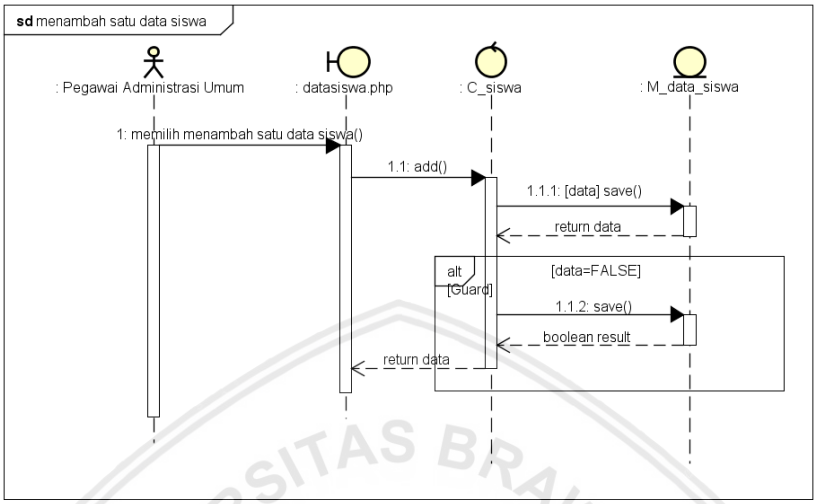


Gambar 5.13 *Sequence Diagram* Menampilkan Data Siswa

5.3.3.2 Sequence Diagram Menambah Data Satu Siswa

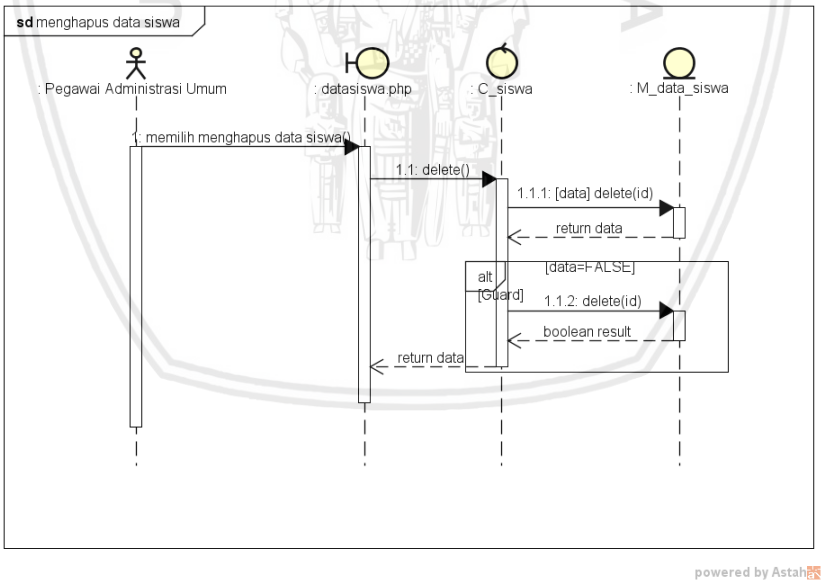
Pada sequence diagram menambah data satu siswa yang ditunjukkan pada Gambar 5.14, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menambah data siswa. Interaksi dimulai ketika aktor Pegawai Administrasi Umum mengakses sistem untuk menambahkan data satu siswa melalui *view* datasiswa.php, maka *controller* C_siswa akan mengirim *message* save() pada *model* M_data_siswa untuk menyimpan data siswa di *database*.

Setelah itu *model* M_data_siswa mengirim *message* ke *controller* C_siswa berupa data yang telah ditambah dan *controller* C_siswa mengirim *message* tersebut ke *view* datasiswa.php untuk dapat ditampilkan dan dilihat oleh aktor Pegawai Administrasi Umum.



Gambar 5.14 *Sequence Diagram* Menambah Data Satu Siswa

5.3.3.3 Sequence Diagram Menghapus Data Siswa

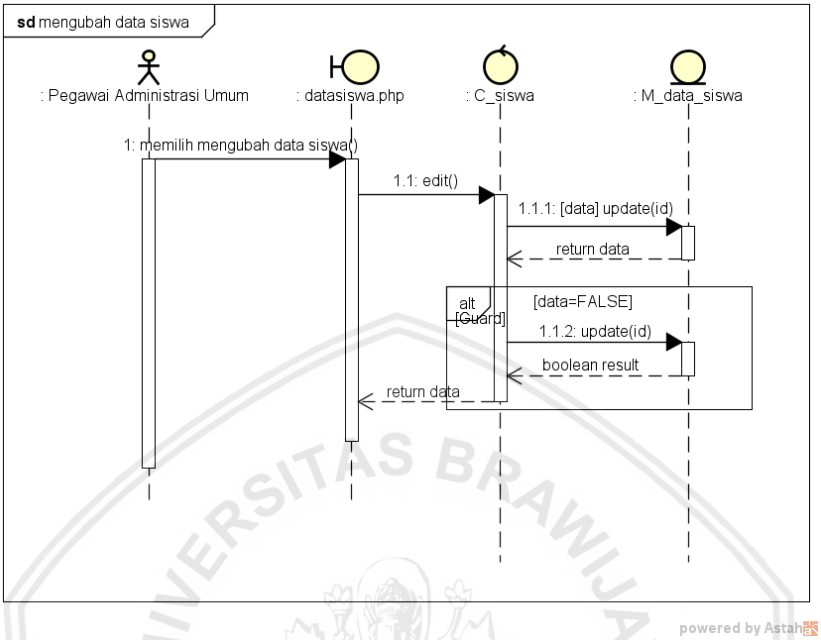


Gambar 5.15 *Sequence Diagram* Menghapus Data Siswa

Pada sequence diagram menghapus data siswa yang ditunjukkan pada Gambar 5.15, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menghapus data siswa. Interaksi dimulai ketika aktor Pegawai Administrasi Umum mengakses sistem untuk menghapus data siswa melalui *view* datasiswa.php, maka *controller* C_siswa akan mengirim *message* delete() pada *model* M_data_siswa untuk menghapus data siswa di *database*. Setelah itu *model* M_data_siswa mengirim *message* ke *controller* C_siswa berupa data yang telah

dihapus dan *controller* C_siswa mengirim *message* tersebut ke *view* datasiswa.php untuk dapat ditampilkan dan dilihat oleh aktor Pegawai Administrasi Umum.

5.3.3.4 Sequence Diagram Mengubah Data Siswa

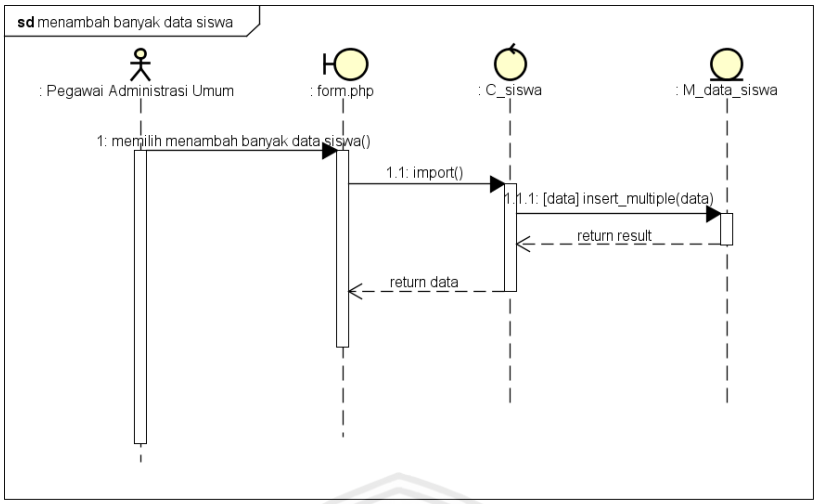


Gambar 5.16 *Sequence Diagram* Mengubah Data Siswa

Pada sequence diagram mengubah data siswa yang ditunjukkan pada Gambar 5.16, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk mengubah data siswa. Interaksi dimulai ketika aktor Pegawai Administrasi Umum mengakses sistem untuk mengubah data siswa melalui *view* datasiswa.php, maka *controller* C_siswa akan mengirim *message* update() pada *model* M_data_siswa untuk memperbarui data siswa di *database*. Setelah itu *model* M_data_siswa mengirim *message* ke *controller* C_siswa berupa data yang telah diubah dan *controller* C_siswa mengirim *message* tersebut ke *view* datasiswa.php untuk dapat ditampilkan dan dilihat oleh aktor Pegawai Administrasi Umum.

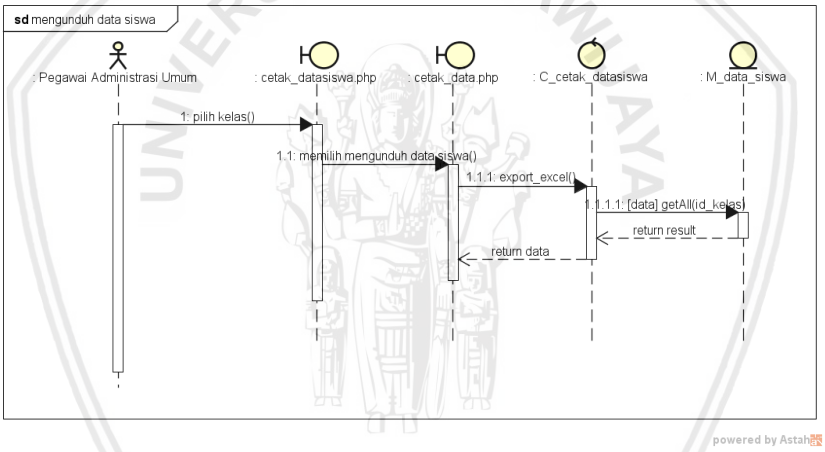
5.3.3.5 Sequence Diagram Menambah Data Banyak Siswa

Pada sequence diagram menambah data banyak siswa yang ditunjukkan pada Gambar 5.17, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menambah data banyak siswa. Interaksi dimulai ketika aktor Pegawai Administrasi Umum mengakses sistem untuk menambahkan data banyak siswa melalui *view* form.php, maka *controller* C_siswa akan mengirim *message* insert_multiple() pada *model* M_data_siswa untuk menyimpan data siswa di *database*. Setelah itu *model* M_data_siswa mengirim *message* ke *controller* C_siswa berupa data yang telah ditambah dan *controller* C_siswa mengirim *message* tersebut ke *view* datasiswa.php untuk dapat ditampilkan dan dilihat oleh aktor Pegawai Administrasi Umum.



Gambar 5.17 *Sequence Diagram* Menambah Data Banyak Siswa

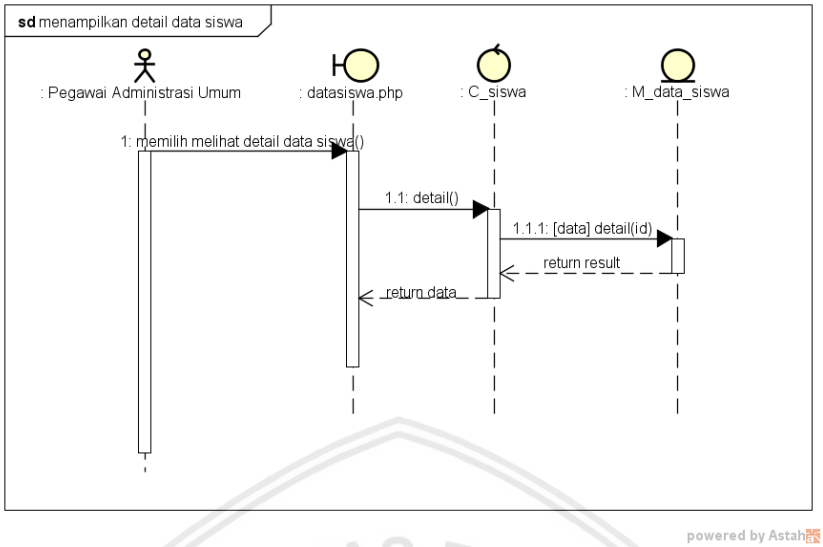
5.3.3.6 Sequence Diagram Mengunduh Data Siswa



Gambar 5.18 *Sequence Diagram* Mengunduh Data Siswa

Pada sequence diagram mengunduh data siswa yang ditunjukkan pada Gambar 5.18, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk mengunduh data siswa. Interaksi dimulai ketika aktor Pegawai Administrasi Umum mengakses sistem untuk mengunduh data siswa melalui *view* cetak_datasiswa.php dan cetak_data.php, maka *controller* C_cetak_datasiswa akan mengirim *message* getAll() pada *model* M_data_siswa. Setelah itu *model* M_data_siswa mengirim *message* ke *controller* C_cetak_datasiswa berupa data yang diminta oleh *controller* C_cetak_datasiswa dan *controller* C_cetak_datasiswa mengirim *message* tersebut ke *view* cetak_data.php untuk dapat diunduh oleh aktor Pegawai Administrasi Umum.

5.3.3.7 Sequence Diagram Menampilkan Detail Data Siswa



Gambar 5.19 *Sequence Diagram* Menampilkan Detail Data Siswa

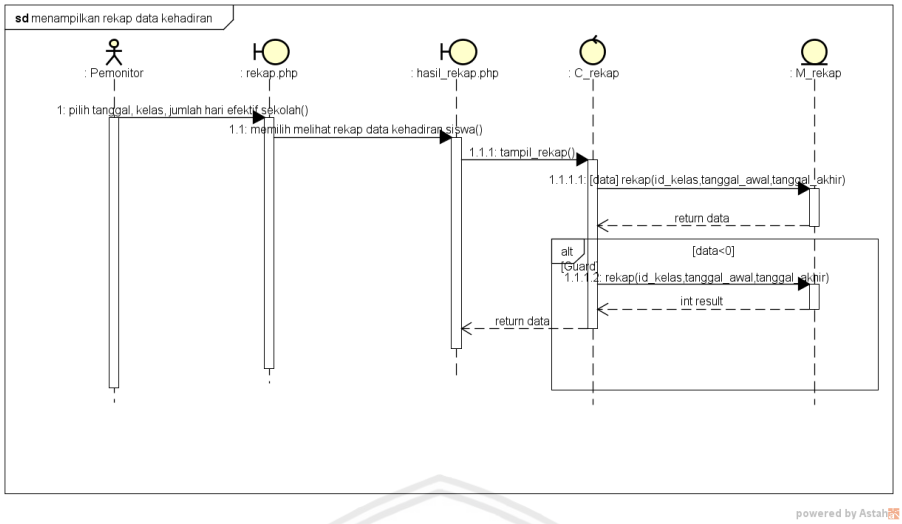
Pada sequence diagram melihat detail data siswa yang ditunjukkan pada Gambar 5.19, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menampilkan detail data siswa. Interaksi dimulai ketika aktor Pegawai Administrasi Umum mengakses sistem untuk menampilkan detail data siswa melalui *view* datasiswa.php maka *controller* C_siswa akan mengirim *message* detail() pada *model* M_data_siswa. Setelah itu *model* M_data_siswa mengirim *message* ke *controller* C_siswa berupa data yang diminta oleh *controller* C_siswa dan *controller* C_siswa mengirim *message* tersebut ke *view* datasiswa.php untuk dapat dilihat oleh aktor Pegawai Administrasi Umum.

5.3.4 Memonitor Rekap Data Kehadiran

Pada proses memonitor rekap data kehadiran terdapat *sequence diagram* menampilkan rekap data kehadiran dan mengunduh rekap data kehadiran. Setiap *sequence diagram* pada proses memonitor rekap data kehadiran menggambarkan interaksi aktor Pemonitor dengan *model, view, dan controller* untuk dapat memonitor rekap data kehadiran.

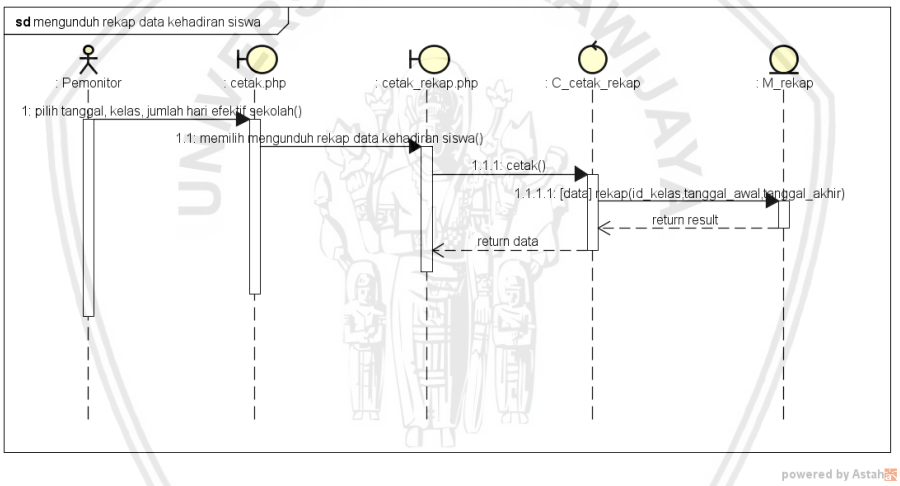
5.3.4.1 Sequence Diagram Menampilkan Rekap Data Kehadiran

Pada sequence diagram menampilkan rekap data kehadiran yang ditunjukkan pada Gambar 5.20, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menampilkan rekap data kehadiran siswa. Interaksi dimulai ketika aktor Pemonitor mengakses sistem untuk menampilkan rekap data kehadiran siswa melalui *view* rekap.php dan hasil_rekap.php, maka *controller* C_rekap akan mengirim *message* rekap() pada *model* M_rekap untuk meminta data yang akan ditampilkan di hasil_rekap.php. Setelah itu *model* M_rekap mengirim *message* ke *controller* C_rekap berupa data yang diminta oleh *controller* C_rekap dan *controller* C_rekap mengirim *message* tersebut ke *view* hasil_rekap.php untuk dapat ditampilkan dan dilihat oleh aktor Pemonitor.



Gambar 5.20 *Sequence Diagram* Menampilkan Rekap Data Kehadiran

5.3.4.2 Sequence Diagram Mengunduh Rekap Data Kehadiran



Gambar 5.21 *Sequence Diagram* Mengunduh Rekap Data Kehadiran

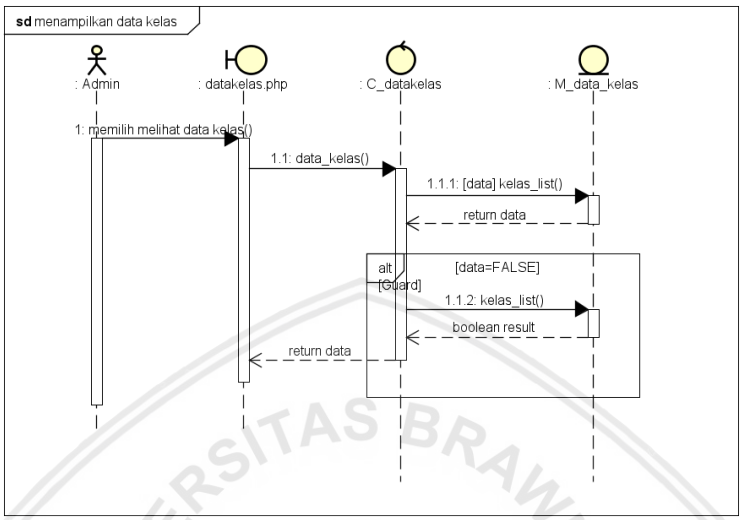
Pada sequence diagram mengunduh rekap data kehadiran yang ditunjukkan pada Gambar 5.21, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk mengunduh rekap data kehadiran siswa. Interaksi dimulai ketika aktor Pemonitor mengakses sistem untuk mengunduh rekap data kehadiran siswa melalui *view* cetak.php dan cetak_rekap.php, maka *controller* C_cetak_rekap akan mengirim *message* rekap() pada *model* M_rekap untuk meminta data yang akan diunduh. Setelah itu *model* M_rekap mengirim *message* ke *controller* C_cetak_rekap berupa data yang diminta oleh *controller* C_cetak_rekap dan *controller* C_cetak_rekap mengirim *message* tersebut ke *view* cetak_rekap.php untuk dapat diunduh oleh aktor Pemonitor.

5.3.5 Mengelola Data *Master*

Pada proses mengelola data *master* terdapat *sequence diagram* menampilkan data kelas, menambah data kelas, menghapus data kelas, dan

mengubah data kelas. Setiap *sequence diagram* pada proses mengelola data *master* menggambarkan interaksi aktor Admin dengan *model, view, dan controller* untuk dapat mengelola data kelas.

5.3.5.1 Sequence Diagram Menampilkan Data Kelas

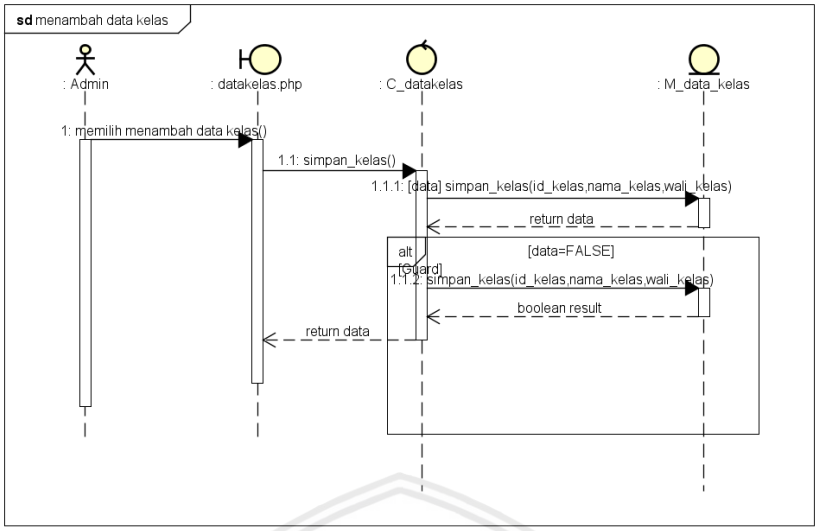


Gambar 5.22 *Sequence Diagram* Menampilkan Data Kelas

Pada sequence diagram menampilkan data kelas yang ditunjukkan pada Gambar 5.22, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menampilkan data kelas. Interaksi dimulai ketika aktor Admin mengakses sistem untuk menampilkan data kelas melalui *view* datakelas.php, maka *controller* C_datakelas akan mengirim *message* kelas_list() pada *model* M_data_kelas untuk meminta data yang akan ditampilkan di datakelas.php. Setelah itu *model* M_data_kelas mengirim *message* ke *controller* C_datakelas berupa data yang diminta oleh *controller* C_datakelas dan *controller* C_datakelas mengirim *message* tersebut ke *view* datakelas.php untuk dapat ditampilkan dan dilihat oleh aktor Admin.

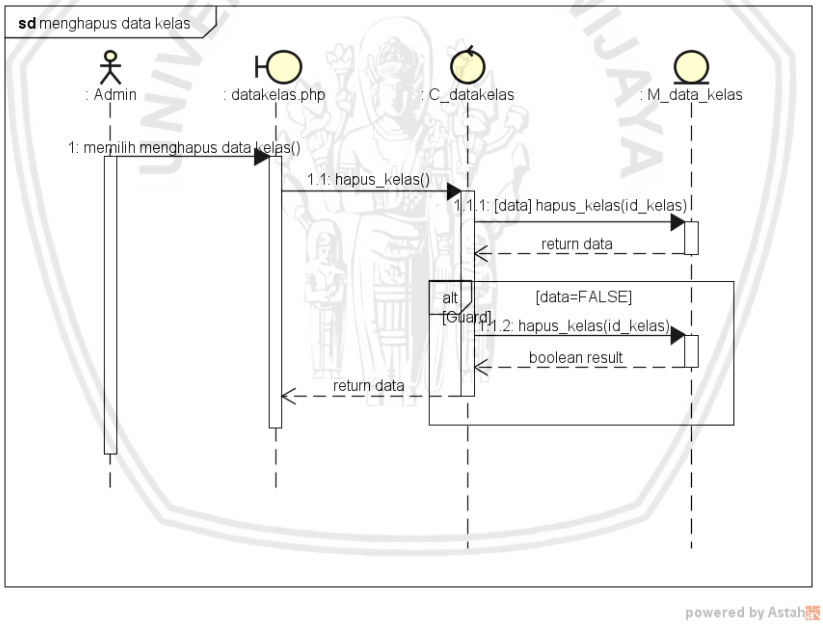
5.3.5.2 Sequence Diagram Menambah Data Kelas

Pada sequence diagram menambah data kelas yang ditunjukkan pada Gambar 5.23, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menambah data kelas. Interaksi dimulai ketika aktor Admin mengakses sistem untuk menambah data kelas melalui *view* datakelas.php, maka *controller* C_datakelas akan mengirim *message* simpan_kelas() pada *model* M_data_kelas untuk menyimpan data kelas di *database*. Setelah itu *model* M_data_kelas mengirim *message* ke *controller* C_datakelas berupa data yang telah ditambah dan *controller* C_datakelas mengirim *message* tersebut ke *view* datakelas.php untuk dapat ditampilkan dan dilihat oleh aktor Admin.



Gambar 5.23 *Sequence Diagram* Menambah Data Kelas

5.3.5.3 Sequence Diagram Menghapus Data Kelas

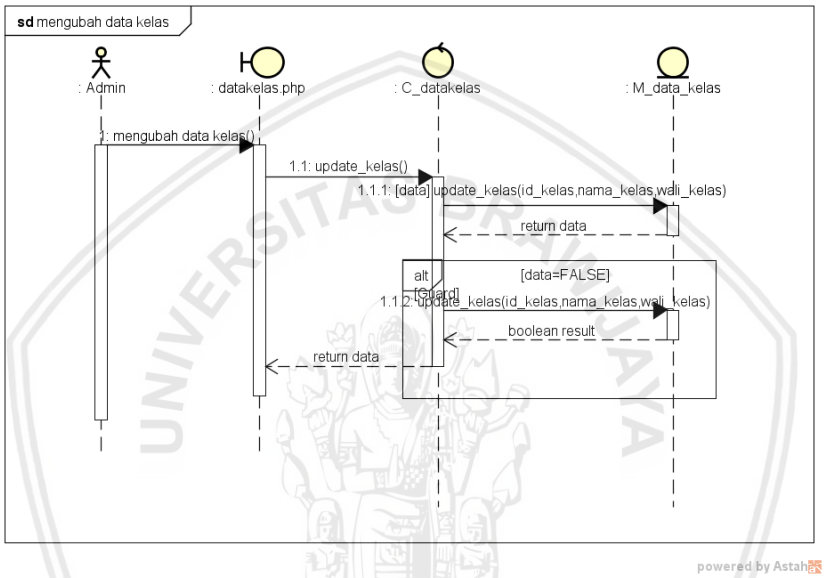


Gambar 5.24 *Sequence Diagram* Menghapus Data Kelas

Pada sequence diagram menghapus data kelas yang ditunjukkan pada Gambar 5.24, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk menghapus data kelas. Interaksi dimulai ketika aktor Admin mengakses sistem untuk menghapus data kelas melalui *view* datakelas.php, maka *controller* C_datakelas akan mengirim *message* hapus_kelas() pada *model* M_data_kelas untuk menghapus data kelas di *database*. Setelah itu *model* M_data_kelas mengirim *message* ke *controller* C_datakelas berupa data yang telah dihapus dan *controller* C_datakelas mengirim *message* tersebut ke *view* datakelas.php untuk dapat ditampilkan dan dilihat oleh aktor Admin.

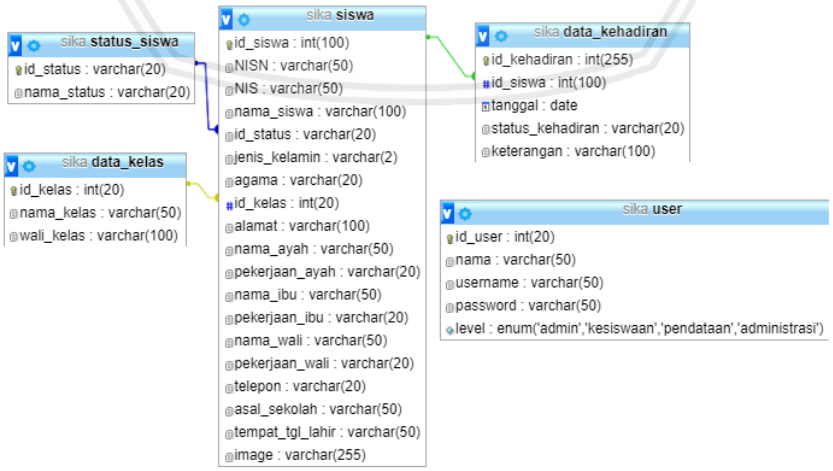
5.3.5.4 Sequence Diagram Mengubah Data Kelas

Pada sequence diagram mengubah data kelas yang ditunjukkan pada Gambar 5.25, menggambarkan interaksi yang terjadi antar objek-objek dalam sistem untuk mengubah data kelas. Interaksi dimulai ketika aktor Admin mengakses sistem untuk mengubah data kelas melalui *view* datakelas.php, maka *controller* C_datakelas akan mengirim *message* update_kelas() pada *model* M_data_kelas untuk mengubah data kelas di *database*. Setelah itu *model* M_data_kelas mengirim *message* ke *controller* C_datakelas berupa data yang telah diubah dan *controller* C_datakelas mengirim *message* tersebut ke *view* datakelas.php untuk dapat ditampilkan dan dilihat oleh aktor Admin.



Gambar 5.25 *Sequence Diagram* Mengubah Data Kelas

5.4 Perancangan Basis Data



Gambar 5.26 *Physical Data Model* Sistem Informasi Kehadiran Siswa

Perancangan basis data merupakan perancangan yang digambarkan berdasarkan pada perancagan kelas analisis yang telah dilakukan pada tahap sebelumnya, yang ditunjukkan oleh Gambar 5.2. Setiap kelas yang terdapat pada diagram kelas analisis digambarkan pada *physical* data model berupa kelas-kelas dengan atribut-atribut yang ada pada kelas, serta relasi antar kelas digambarkan pada *physical* data model. Sistem Informasi Kehadiran Siswa yang dikembangkan menggunakan MySQL sebagai basis datanya. Pada Gambar 5.26 menunjukkan *physical* data model pada Sistem Informasi Kehadiran Siswa yang dikembangkan. Pada *physical* data model Sistem Informasi Kehadiran Siswa terdapat 5 tabel yaitu tabel siswa, data_kelas, status_siswa, data_kehadiran, dan user. Untuk menyimpan data dari kelas yang ada pada SMA Suluh Jakarta menggunakan tabel data_kelas. Untuk menyimpan data siswa menggunakan tabel siswa. Untuk menyimpan data terkait kehadiran siswa menggunakan tabel data_kehadiran. Tabel yang digunakan untuk menyimpan macam-macam status siswa yaitu tabel status_siswa. Sedangkan untuk autentikasi pengguna yang melakukan login kedalam sisem menggunakan tabel user.

5.5 Perancangan Algoritme

Perancangan algoritme merupakan tahapan yang dilakukan untuk menentukan dan menyusun operasi-operasi logika yang digunakan dalam pengembangan Sistem Informasi Kehadiran Siswa. Perancangan algoritme kemudian dituliskan kedalam beberapa *pseudocode*. Perancangan algoritme yang akan dijelaskan berupa perancangan algoritme mendata kehadiran siswa, mengelola data siswa, dan memonitor rekap data kehadiran.

5.5.1 Mendata Kehadiran Siswa

Pada perancangan algoritme mendata kehadiran siswa yang ditunjukkan Tabel 5.1 menjelaskan mengenai aktor Guru Kelas yang menggunakan sistem untuk menyimpan data kehadiran siswa melalui fungsi insert_pendataan() pada *controller* C_pendataan dan data disimpan ke *database* melalui fungsi insert() pada *model* M_pendataan.

Tabel 5.1 *Pseudocode* Mendata Kehadiran Siswa

No.	***Pseudocode***
	Mulai Fungsi insert_pendataan
1	Init data = post;
2	Perulangan ($i=0; $i < jumlah id siswa ; $i++)
3	Init id_siswa=data id_siswa nilai i;
4	Init data_now=data tanggal;
5	Init check_kehadiran = Mengambil data dari database melalui model m_pendataan dengan memanggil fungsi check_data_with_same_day dengan parameter id_siswa dan date_now;
6	if($check_kehadiran==null)
7	Init vals_data=array 'id_siswa' => $data['id_siswa'][$i], 'status_kehadiran' => $data['status_kehadiran'.$data['id_siswa'][$i]], 'keterangan' => $data['keterangan'.$data['id_siswa'][$i]], 'tanggal' => $data['tanggal'],);

Tabel 5.1 *Pseudocode* Mendata Kehadiran Siswa (lanjutan)

No.	*Pseudocode*
8	Init result = Mengambil data dari database melalui model m_pendataan dengan memanggil fungsi insert dengan parameter vals_data;
9	if ($result > 0) Menampilkan data berhasil disimpan dan menampilkan halaman 'pendataan/c_update_data';
10	else Menampilkan data gagal disimpan dan menampilkan halaman 'pendataan/c_update_data'; Selesai

5.5.2 Mengelola Data Siswa

Pada perancangan algoritme mengelola data siswa yang ditunjukkan oleh Tabel 5.2 menjelaskan mengenai aktor Pegawai Admininistrasi Umum yang menggunakan sistem untuk menambah data siswa. Untuk menambah data siswa melalui fungsi add() pada *controller* C_siswa dan data yang ditambahkan tersimpan kedalam *database* melalui fungsi save() pada *model* M_data_siswa.

Tabel 5.2 *Pseudocode* Mengelola Data Siswa

No.	Pseudocode
	Mulai Fungsi add
1	Init siswa = mengakses database m_data_siswa;
2	Init validation = mengakses form_validation;
3	Menetapkan aturan pada validation dengan memanggil fungsi rules pada siswa;
4	Jika validation dapat berjalan data tersimpan pada database dengan memanggil fungsi save;
5	Menampilkan pesan Data berhasil disimpan! dan halaman akan beralih ke list siswa dengan diaksesnya kesiswaan/c_siswa;
6	Init data nama_navbar = 'Tambah Data Siswa';
7	Init data kelas dengan mengakses fungsi get_kelas pada database melalui model m_data_siswa;
8	Init data status dengan mengakses fungsi get_status_siswa pada database melalui model m_data_siswa;
9	Menampilkan halaman 'layout/header';
10	Menampilkan halaman 'layout/sidebar_datasiswa';
11	Menampilkan halaman 'layout/navbar' dengan mengirimkan nilai data;
12	Menampilkan halaman 'administrasi/tambah_data';

5.5.3 Memonitor Rekap Data Kehadiran

Pada perancangan algoritme memonitor rekap data kehadiran yang ditunjukkan Tabel 5.3 menjelaskan mengenai aktor Pemonitor yang menggunakan sistem untuk mengunduh rekap data kehadiran siswa. Untuk mengunduh rekap data kehadiran siswa melalui fungsi cetak() pada *controller* C_cetak_rekap dan mengambil data yang akan diunduh oleh aktor Pemonitor melalui fungsi rekap() pada *model* M_rekap.

Tabel 5.3 *Pseudocode* Memonitor Rekap Data Kehadiran

No.	*Pseudocode*
	Mulai Fungsi cetak

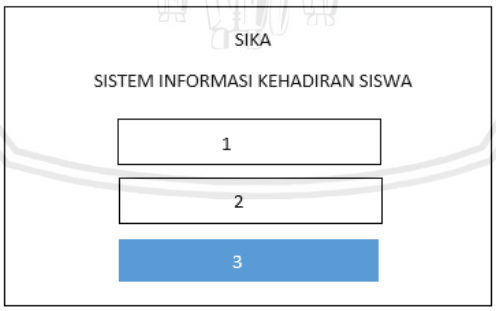
Tabel 5.3 *Pseudocode* Memonitor Rekap Data Kehadiran (lanjutan)

No.	*Pseudocode*
1	`Init id_kelas = post('id_kelas');`
2	`Init tanggal_awal = post('tanggal_awal');`
3	`Init tanggal_akhir = post('tanggal_akhir');`
4	`Init he = post('he');`
5	`Init rekap rekap dengan mengambil data dari database melalui model m_rekap dengan fungsi rekap dengan parameter id_kelas, tanggal_awal, dan tanggal_akhir;`
6	`Init rekap cetak dengan mengambil data dari database melalui model m_rekap dengan fungsi jml_siswa dengan parameter id_kelas;`
7	`Init rekap jml_all dengan mengambil data dari database melalui model m_rekap dengan fungsi jml_all dengan parameter id_kelas;`
8	`Init rekap jml_allabsen dengan mengambil data dari database melalui model m_rekap dengan fungsi jml_allabsen dengan parameter id_kelas, tanggal_awal, dan tanggal_akhir;`
9	`Init rekap nama_kelas dengan mengambil data dari database melalui model m_rekap dengan fungsi nama_kelas dengan parameter id_kelas;`
10	`Init rekap wali_kelas dengan mengambil data dari database melalui model m_rekap dengan fungsi wali_kelas dengan parameter id_kelas;`
11	`Menampilkan halaman 'pemonitor/cetak' dengan mengirimkan nilai rekap;`

5.6 Perancangan Antarmuka Pengguna

Pada bagian perancangan antarmuka pengguna menjelaskan mengenai gambaran dari tampilan antarmuka pengguna yang nantinya ada pada Sistem Informasi Kehadiran Siswa yang terdiri dari antarmuka *login*, antarmuka kehadiran siswa secara menyeluruh, antarmuka pendataaan kehadiran siswa, antarmuka mengelola data kelas, antarmuka mengelola data siswa, dan antarmuka rekap data kehadiran.

5.6.1 Antarmuka *Login*

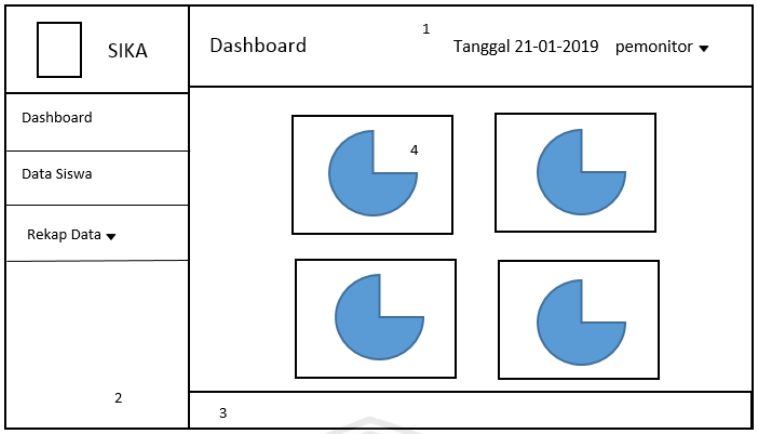


Gambar 5.27 Perancangan Antamuka Login

Antarmuka yang akan ditampilkan kepada aktor Pengguna saat mengakses Sistem Informasi Kehadiran Siswa ditunjukkan oleh Gambar 5.27 yaitu perancangan antarmuka *login*. Keterangan yang terdapat pada Gambar 5.27 dijelaskan sebagai berikut:

1. Kotak *input* untuk memasukkan *username*.
2. Kotak *input* untuk memasukkan *password*.
3. Tombol *sign in* untuk masuk ke dalam sistem.

5.6.2 Antarmuka Kehadiran Siswa Secara Menyeluruh

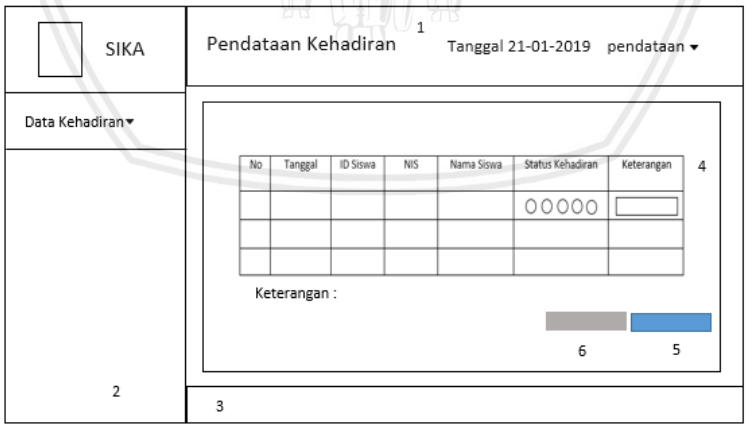


Gambar 5.28 Perancangan Antamuka Kehadiran Siswa Secara Menyeluruh

Antarmuka kehadiran siswa secara menyeluruh akan ditampilkan kepada aktor Pemonitor ketika telah masuk ke dalam sistem atau memilih *dashboard* pada *sidebar*. Perancangan antarmuka kehadiran siswa secara menyeluruh ditunjukkan oleh Gambar 5.28. Keterangan yang terdapat pada Gambar 5.28 dijelaskan sebagai berikut:

1. *Header.*
2. *Sidebar*.
3. *Footer*.
4. *Pie chart* kehadiran.

5.6.3 Antarmuka Pendataan Kehadiran Siswa

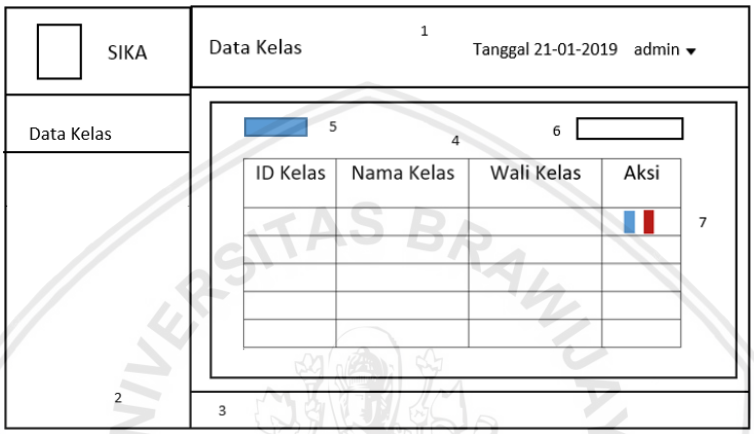


Gambar 5.29 Perancangan Antamuka Pendataan Kehadiran Siswa

Antarmuka pendataan kehadiran akan ditampilkan kepada aktor Guru Kelas ketika memilih untuk memasukkan data kehadiran siswa, dengan memilih data kehadiran pada *sidebar*. Perancangan antarmuka pendataan kehadiran siswa ditunjukkan oleh Gambar 5.29. Keterangan yang terdapat pada Gambar 5.29 dijelaskan sebagai berikut:

1. *Header.*
2. *Sidebar*.
3. *Footer.*
4. Tabel pendataan kehadiran siswa.
5. Tombol simpan data.
6. Tombol batal simpan data.

5.6.4 Antarmuka Mengelola Data kelas



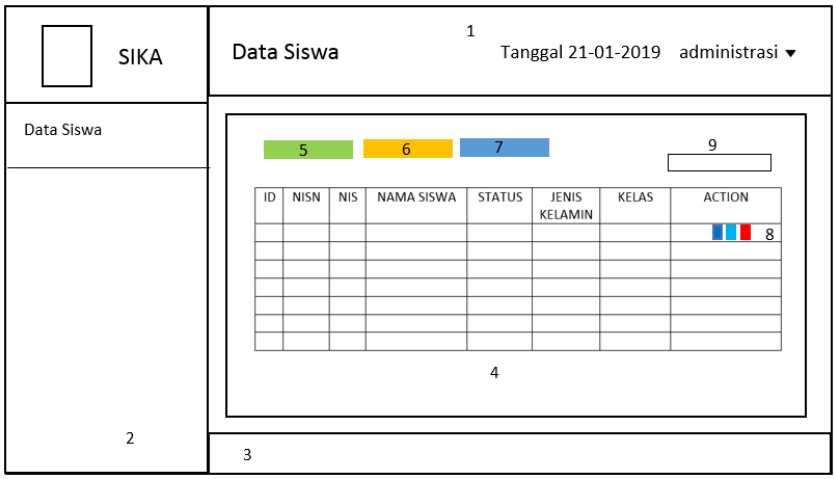
Gambar 5.30 Perancangan Antamuka Mengelola Data Kelas

Antarmuka mengelola data kelas akan ditampilkan kepada aktor Admin ketika telah berhasil masuk kedalam sistem dan memilih untuk memasukkan data kelas. Perancangan antarmuka mengelola data kelas ditunjukkan oleh Gambar 5.30. Keterangan yang terdapat pada Gambar 5.30 dijelaskan sebagai berikut:

1. *Header.*
2. *Sidebar*.
3. *Footer*.
4. Tabel data Kelas.
5. Tombol tambah data kelas.
6. Form pencarian data kelas.
7. Tombol untuk *edit* dan hapus data kelas.

5.6.5 Antarmuka Mengelola Data Siswa

Antarmuka mengelola data siswa akan ditampilkan kepada aktor Pegawai Administrasi Umum ketika memilih untuk memasukkan data siswa, dengan memilih data siswa pada *sidebar*. Perancangan antarmuka mengelola data siswa ditunjukkan oleh Gambar 5.31.

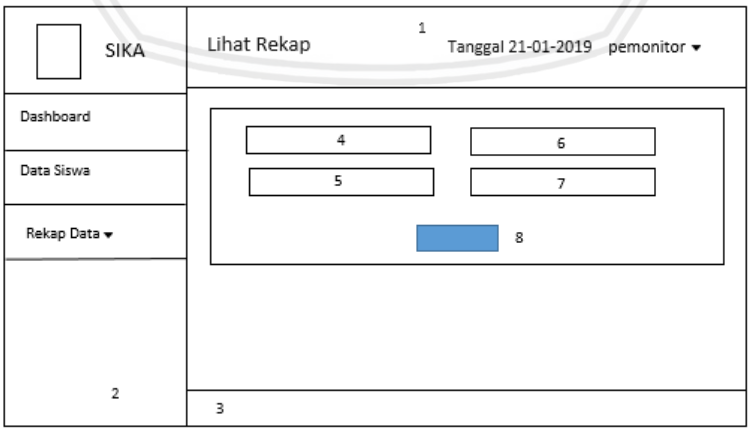


Gambar 5.31 Perancangan Antamuka Mengelola Data Siswa

Keterangan yang terdapat pada Gambar 5.31 dijelaskan sebagai berikut:

1. *Header.*
2. *Sidebar*.
3. *Footer.*
4. Tabel data siswa.
5. Tombol tambah data satu siswa.
6. Tombol tambah data banyak siswa.
7. Tombol cetak data siswa.
8. Tombol untuk detail, *edit*, dan hapus data siswa.
9. Form pencarian data siswa.

5.6.6 Antarmuka Rekap Data Kehadiran



Gambar 5.32 Perancangan Antamuka Rekap Data Kehadiran

Antarmuka rekap data kehadiran akan ditampilkan kepada aktor Pemonitor ketika memilih untuk melihat hasil rekap data kehadiran siswa, dengan

memilih rekap data pada *sidebar*. Perancangan antarmuka rekap data kehadiran ditunjukkan oleh Gambar 5.32. Keterangan yang terdapat pada Gambar 5.32 dijelaskan sebagai berikut:

1. *Header*.
2. *Sidebar*.
3. *Footer*.
4. Kotak *input* untuk pilih tanggal awal.
5. Kotak *input* untuk pilih kelas.
6. Kotak *input* untuk pilih tanggal akhir.
7. Kotak *input* untuk memasukkan jumlah hari efektif sekolah.
8. Tombol tampilkan data.

BAB 6 IMPLEMENTASI

Bab ini menjelaskan mengenai kebutuhan yang diperlukan dalam implementasi Sistem Informasi Kehadiran Siswa dan hasil dari implementasi Sistem Informasi Kehadiran Siswa yang dilakukan. Implementasi sistem dilakukan berdasarkan pada perancangan sistem yang telah dilakukan pada tahap sebelumnya. Pada bab implementasi ini berisi spesifikasi lingkungan implementasi sistem, struktur artefak dari sistem, implementasi algoritme, dan implementasi antarmuka pengguna.

6.1 Spesifikasi Lingkungan Implementasi

Pada bagian spesifikasi lingkungan implementasi menjelaskan mengenai spesifikasi perangkat keras dan kebutuhan perangkat lunak yang digunakan untuk pengembangan Sistem Informasi Kehadiran Siswa, serta terdapat spesifikasi lingkungan pengembangan yang ditunjukkan oleh Tabel 6.3. Untuk spesifikasi perangkat keras ditunjukan oleh Tabel 6.1 dan Tabel 6.2 menunjukkan spesifikasi kebutuhan perangkat lunak.

Tabel 6.1 Spesifikasi Perangkat Keras

Unit Komputasi	Laptop SONY SVE14136CVB
CPU	Core (TM) i5-3230M 2.60GHz
Kapasitas RAM	4 Gb
Kapasitas Penyimpanan	500 Gb
Kartu Grafis	AMD Radeon HD 7500M
Resolusi Layar	1366 x 768 pixels

Tabel 6.2 Spesifikasi Kebutuhan Perangkat Lunak

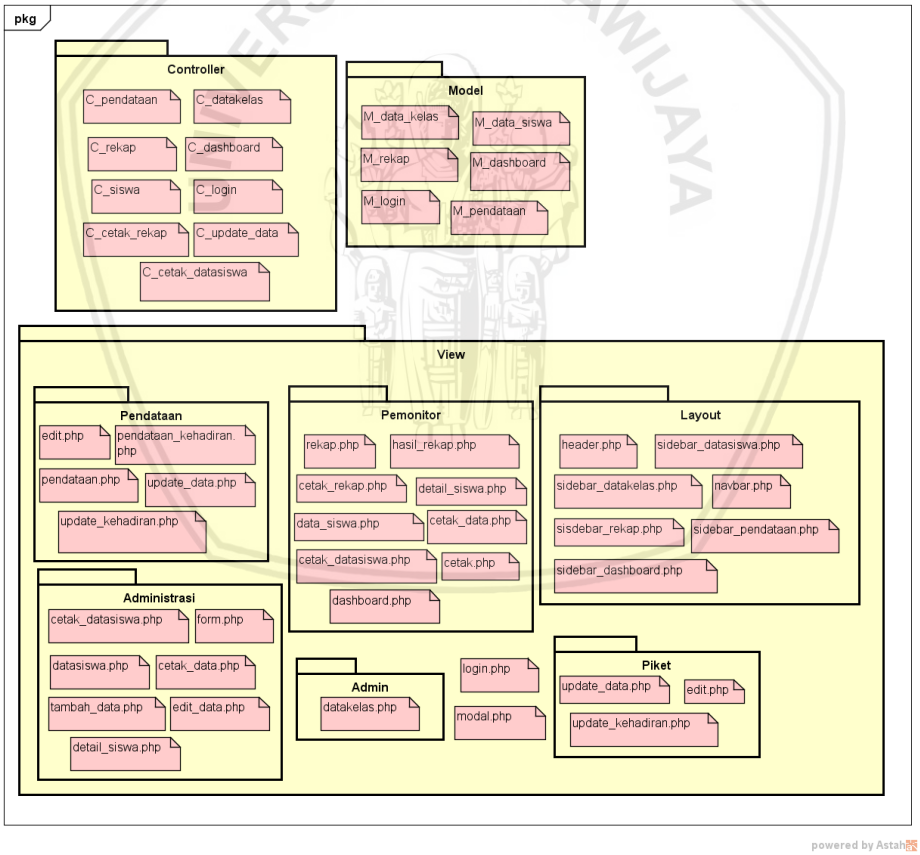
Sistem Operasi	Windows 8.1 Enterprise 64-bit
Web Server	Apache 2.0
DBMS	MariaDB 10.1.19
Bahasa Pemrograman	PHP 5.6.28
Editor Kode Progrm	Sublime Text 3
Peramban	Google Chrome Version 69.0.3497.100 (Official Build) (64-bit)
Perangkat Lunak atau Bahasa Pemrograman Pendukung	Javascript, HTML
***Library* Pendukung**	Tcpdf

Tabel 6.3 Spesifikasi Lingkungan Pengembangan

Kapasitas RAM (Minimal)	256 MB
Kapasistas Memori (Minimal)	85 MB
Versi PHP	5.3 atau lebih
Library Pendukung	Php-font-lib v0.3.x

6.2 Struktur Artefak Sistem

Pada bagian struktur artefak sistem menjelaskan mengenai penggambaran struktur artefak yang digunakan dalam pengembangan Sistem Informasi Kehadiran Siswa berdasarkan pada perancangan diagaram kelas *controller, model* dan *view* yang telah dibuat sebelumnya. Penggambaran struktur artefak dari Sistem Informasi Kehadiran Siswa menggunakan *package* yang terbagi menjadi *package controller, model* dan *view*. Struktur artefak dari Sistem Informasi Kehadiran Siswa ditunjukkan oleh Gambar 6.1.



Gambar 6.1 Struktur Artefak Sistem Informasi Kehadiran Siswa

Struktur artefak dari Sistem Informasi Kehadiran Siswa yang ditunjukkan oleh Gambar 6.1 berisi 3 *package* di dalamnya yaitu *controller, model* dan *view. Package controller* pada struktur artefak sistem terdiri dari 9 file yang menggambarkan *controller* yang ada pada Sistem Informasi Kehadiran Siswa sesuai dengan diagram kelas perancangan *controller*. *Package model* pada

struktur artefak sistem terdiri dari 6 file yang menggambarkan *model* yang ada pada Sistem Informasi Kehadiran Siswa sesuai dengan diagram kelas perancangan *model*. *Package view* pada struktur artefak sistem terdiri dari 27 file yang menggambarkan *view* yang ada pada Sistem Informasi Kehadiran Siswa sesuai dengan diagram kelas perancangan *stereotype view* yang telah dilakukan pada tahap sebelumnya.

6.3 Implementasi Algoritme

Pada bagian implementasi algoritme menjelaskan mengenai hasil implementasi dari peracangan yang telah dibuat pada tahap sebelumnya berupa kode program. Implementasi algoritme yang dijelaskan terdiri dari implementasi algoritme mendata kehadiran siswa, mengelola data siswa, dan memonitor rekap data kehadiran.

6.3.1 Implementasi Algoritme Mendata Kehadiran Siswa

Implementasi algoritme mendata kehadiran siswa merupakan fitur pada sistem yang digunakan untuk menyimpan data kehadiran siswa berdasarkan pada rancangan algoritme mendata kehadiran siswa yang telah dibuat pada tahap sebelumnya. Pada Tabel 6.4 merupakan hasil implementasi algoritme mendata kehadiran siswa yang digunakan untuk menyimpan data kehadiran siswa.

Tabel 6.4 Implementasi Algoritme Mendata Kehadiran Siswa

No.	Kode Program
	`class C_pendataan extends CI_Controller {`
	`public function insert_pendataan(){`
1	`$data = $this->input->post();`
2	`for ($i=0; $i < count($data['id_siswa']); $i++) {`
3	`$id_siswa = $data['id_siswa'][$i];`
4	`$date_now = $data['tanggal'];`
5	`$check_kehadiran = $this->m_pendataan->check_data_with_same_day($id_siswa,$date_now);`
6	`if(is_null($check_kehadiran)){`
7	`$vals_data = array( 'id_siswa' => $data['id_siswa'][$i], 'status_kehadiran' => $data['status_kehadiran'.$data['id_siswa'][$i]], 'tanggal' => $data['tanggal'], );`
8	`$result = $this->m_pendataan->insert($vals_data); } }`
9	`if ($result > 0) { echo "<script>alert('Data berhasil disimpan!');window.location='/SIKA/index.php/pendataan/c_update_data';</script>";`
10	`}else{ echo "<script>alert('Data gagal disimpan!');history.go(-2);</script>";}}`

6.3.2 Implementasi Algoritme Mengelola Data Siswa

Implementasi algoritme mengelola data siswa merupakan fitur pada sistem yang digunakan untuk menambah data siswa berdasarkan pada rancangan

algoritme mengelola data siswa yang telah dibuat pada tahap sebelumnya. Pada Tabel 6.5 merupakan hasil implementasi algoritme mengelola data siswa.

Tabel 6.5 Implementasi Algoritme Mengelola Data Siswa

No.	Kode Program
	class C_siswa extends CI_Controller{
	public function add(){
1	$siswa = $this->m_data_siswa;
2	$validation = $this->form_validation;
3	$validation->set_rules($siswa->rules());
4	if ($validation->run()) {
	$siswa->save();
5	echo "<script>alert('Data berhasil disimpan!');window.location='/SIKA/index.php/kesiswaan/c_siswa'; </script>";
	}
6	$data['nama_navbar'] = 'Tambah Data Siswa';
7	$data['kelas']=$this->m_data_siswa->get_kelas();
8	$data['status']=$this->m_data_siswa->get_status_siswa();
9	$this->load->view('layout/header');
10	$this->load->view('layout/sidebar_datasiswa');
11	$this->load->view('layout/navbar',$data);
12	$this->load->view('administrasi/tambah_data');}}

6.3.3 Implementasi Algoritme Memonitor Rekap Data Kehadiran

Implementasi algoritme memonitor rekap data kehadiran merupakan fitur pada sistem yang digunakan untuk mencetak hasil rekap data kehadiran siswa berdasarkan pada rancangan algoritme memonitor rekap data kehadiran yang telah dibuat pada tahap sebelumnya. Pada Tabel 6.6 merupakan hasil implementasi algoritme memonitor rekap data kehadiran.

Tabel 6.6 Implementasi Algoritme Memonitor Rekap Data Kehadiran

No.	Kode Program
	class C_cetak_rekap extends CI_Controller {
	public function cetak(){
1	$id_kelas=$this->input->post('id_kelas');
2	$tanggal_awal=$this->input->post('tanggal_awal');
3	$tanggal_akhir=$this->input->post('tanggal_akhir');
4	$he=$this->input->post('he');
5	$rekap['rekap']=$this->m_rekap->rekap($id_kelas,$tanggal_awal,$tanggal_akhir);
6	$rekap['cetak']=$this->m_rekap->jml_siswa($id_kelas);
7	$rekap['jml_all']=$this->m_rekap->jml_all($id_kelas);
8	$rekap['jml_allabsen']=$this->m_rekap->jml_allabsen($id_kelas,$tanggal_awal,$tanggal_akhir);
9	$rekap['nama_kelas']=$this->m_rekap->nama_kelas($id_kelas);
10	$rekap['wali_kelas']=$this->m_rekap->wali_kelas($id_kelas);
11	$this->load->view('pemonitor/cetak',$rekap); }}

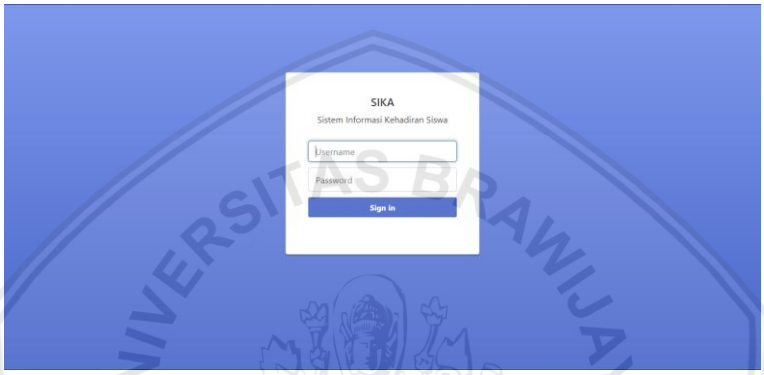
6.4 Implementasi Antarmuka Pengguna

Pada bagian implementasi antarmuka pengguna menjelaskan mengenai beberapa hasil dari implementasi perancangan antarmuka pengguna yang telah

dibuat pada tahap sebelumnya. Hasil dari implemtasi perancangan antarmuka pengguna berupa antarmuka *login*, antarmuka kehadiran siswa secara menyeluruh, antarmuka pendataan kehadiran siswa, antarmuka mengelola data kelas, antarmuka mengelola data siswa, dan antarmuka rekap data kehadiran.

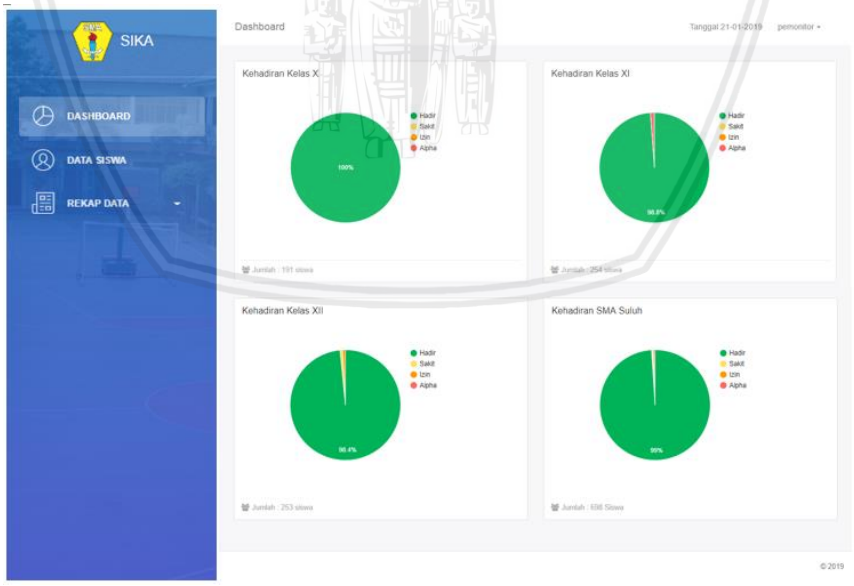
6.4.1 Implementasi Antarmuka *Login*

Implementasi antarmuka *login* merupakan tampilan yang akan pertama kali dilihat oleh Pengguna ketika mengakses Sistem Informasi Kehadiran Siswa. Pada implementasi antarmuka *login* yang ditunjukkan oleh Gambar 6.2 terdapat masukkan (*input*) yang perlu diisi yaitu *username* dan *password* untuk dapat masuk ke dalam sistem dengan menggunakan fungsi *sign in*.



Gambar 6.2 Implementasi Antarmuka Login

6.4.2 Implementasi Antarmuka Kehadiran Siswa Secara Menyeluruh

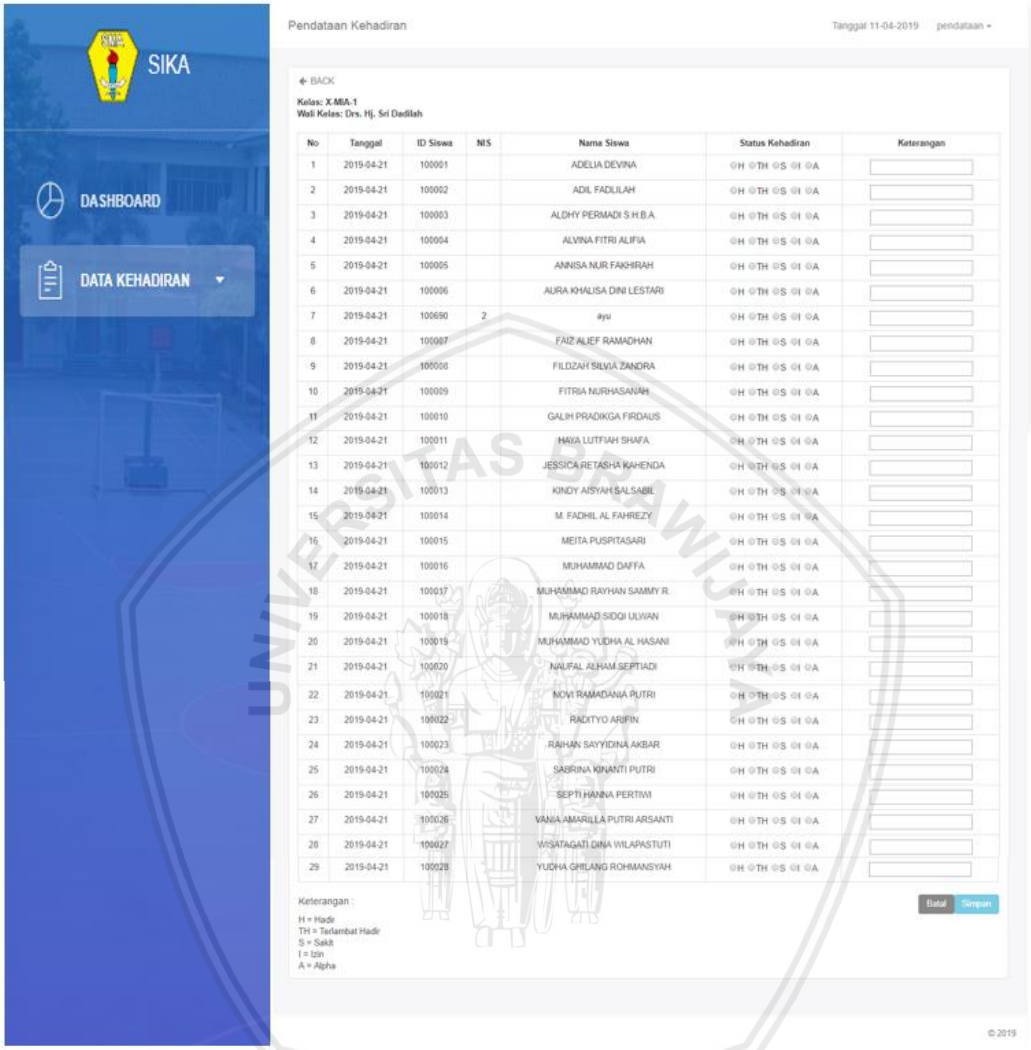


Gambar 6.3 Implementasi Antarmuka Kehadiran Siswa Secara Menyeluruh

Implementasi antarmuka kehadiran siswa secara menyeluruh merupakan tampilan yang akan dilihat oleh kepala sekolah, wali kelas dan guru BK setelah berhasil melakukan *login* dan masuk kedalam sistem atau saat kepala sekolah, wali kelas dan guru BK memilih *dashboard* pada *sidebar*. Pada implementasi antarmuka

kehadiran siswa secara menyeluruh yang ditunjukkan oleh Gambar 6.3 terdapat *pie chart* yang berisi kehadiran seluruh siswa dari setiap tingkatan kelas.

6.4.3 Implementasi Antarmuka Pendataan Kehadiran Siswa

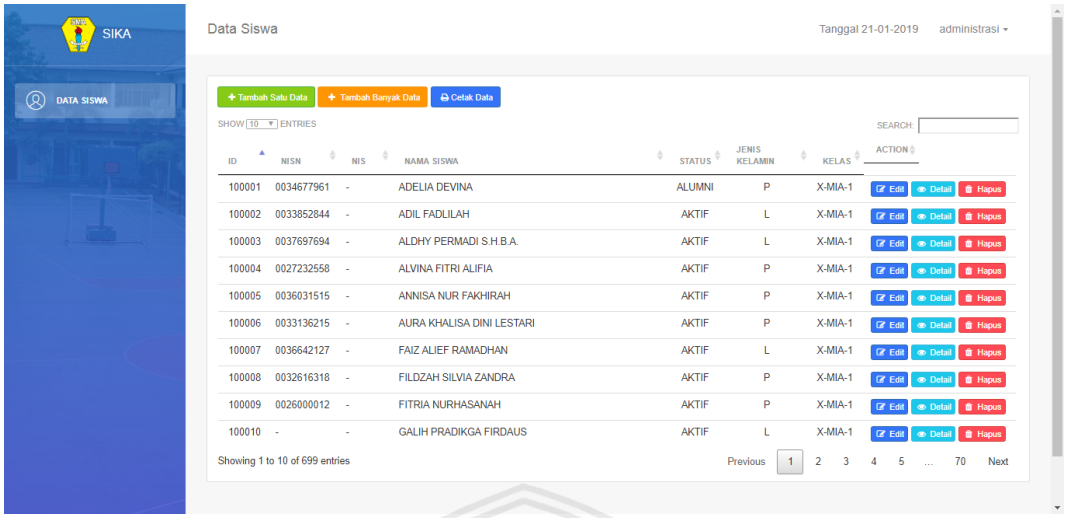


Gambar 6.4 Implementasi Antarmuka Pendataan Kehadiran Siswa

Implementasi antarmuka pendataan kehadiran merupakan tampilan yang akan dilihat oleh guru kelas saat memilih data kehadiran pada *sidebar*. Pada implementasi antarmuka pendataan kehadiran yang ditunjukkan oleh Gambar 6.4 terdapat nama-nama siswa untuk didata kehadirannya, serta terdapat juga keterangan untuk menuliskan keterangan kehadiran siswa jika diperlukan.

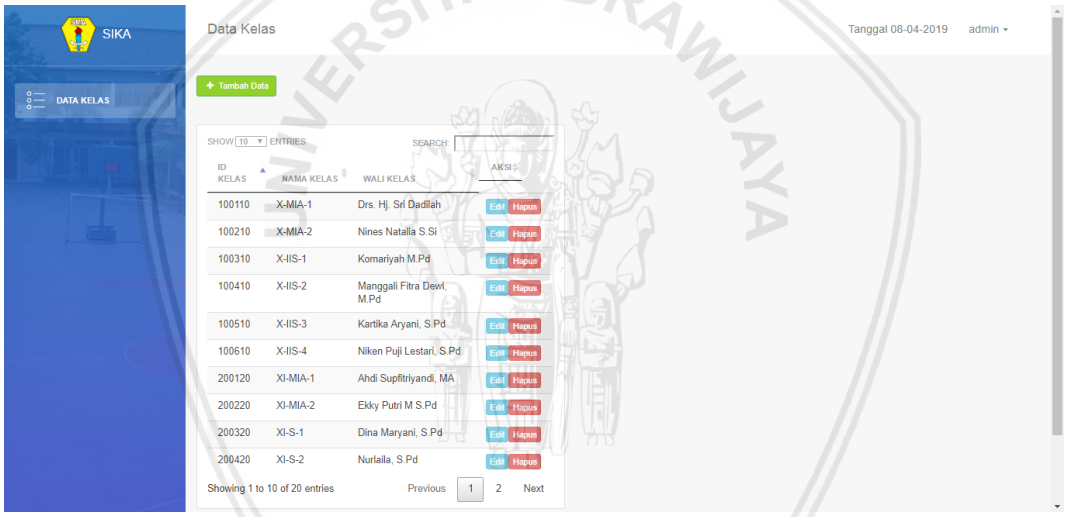
6.4.4 Implementasi Antarmuka Mengelola Data Siswa

Implementasi antarmuka mengelola data siswa merupakan tampilan yang akan dilihat oleh tata usaha saat memilih data siswa pada *sidebar*. Pada Implementasi antarmuka mengelola data siswa yang ditunjukkan oleh Gambar 6.5 terdapat daftar nama-nama siswa yang datanya telah tersimpan didalam *database*, serta terdapat pilihan untuk menambah, mencetak, menghapus, mengubah, dan melihat detail data siswa.



Gambar 6.5 Implementasi Antarmuka Mengelola Data Siswa

6.4.5 Implementasi Antarmuka Mengelola Data Kelas

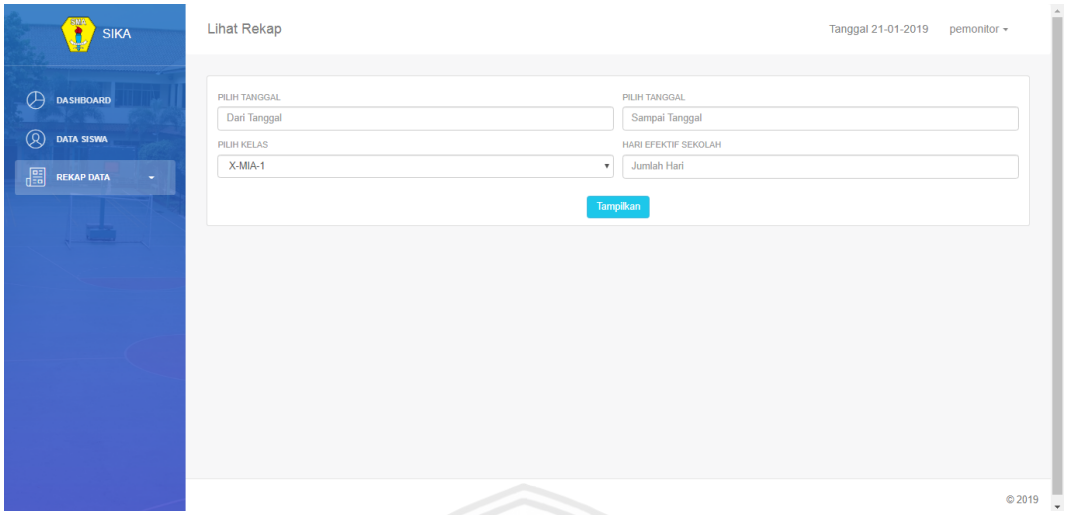


Gambar 6.6 Implementasi Antarmuka Mengelola Data Kelas

Implementasi antarmuka mengelola data kelas merupakan tampilan yang akan dilihat oleh pengembang saat berhasil masuk kedalam sistem. Pada Implementasi antarmuka mengelola data kelas yang ditunjukkan oleh Gambar 6.6 terdapat daftar nama-nama kelas yang telah tersimpan didalam *database*, serta terdapat pilihan untuk menambah, mengubah, dan menghapus data kelas.

6.4.6 Implementasi Antarmuka Rekap Data Kehadiran

Implementasi antarmuka rekap data kehadiran merupakan tampilan yang akan dilihat oleh kepala sekolah, wali kelas dan guru BK saat memilih rekap data pada *sidebar*. Pada Implementasi antarmuka rekap data kehadiran yang ditunjukkan oleh Gambar 6.7 terdapat masukkan (*input*) yang perlu diisi yaitu tanggal awal rekap, tanggal akhir rekap, kelas dan jumlah hari efektif sekolah yang ingin dilihat rekap data kehadirannya.



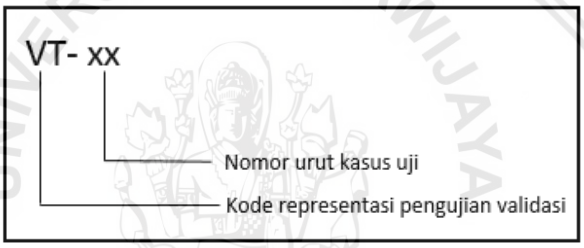
Gambar 6.7 Implementasi Antarmuka Rekap Data Kehadiran

BAB 7 PENGUJIAN

Bab ini menjelaskan mengenai hasil pengujian dari implementasi dan perancangan Sistem Informasi Kehadiran Siswa yang telah dilakukan. Pengujian yang dilakukan pada Sistem Informasi Kehadiran Siswa menggunakan pengujian *black-box* dengan teknik validasi dan pengujian kompatibilitas pada beberapa peramban yang berbeda.

7.1 Pengujian Validasi

Pengujian validasi pada Sistem Informasi Kehadiran Siswa dilakukan untuk memastikan sistem yang dikembangkan telah sesuai dengan persyaratan fungsional sistem yang diidentifikasi sebelumnya dan sistem telah sesuai dengan harapan pengguna. Fungsi-fungsi yang terdapat pada Sistem Informasi Kehadiran Siswa yang diuji dengan menggunakan pengujian Validasi. Keterangan terkait pemberian kode pada pengujian validasi yang dilakukan dapat dilihat pada Gambar 7.1.



Gambar 7.1 Kode Pengujian *Black-Box* dengan Teknik Validasi

Teknik validasi pada pengujian *black-box* dilakukan berdasarkan pada skenario uji yang didapatkan dari alur *use case login,* mendata kehadiran siswa, memonitor rekap data kehadiran, mengelola data siswa, dan memonitor kehadiran siswa secara menyeluruh. Kemudian berdasarkan pada skenario uji yang telah dibuat, kasus uji dapat teridentifikasi dan pengujian *black-box* dengan teknik validasi dapat dilakukan. Kasus uji pada pengujian *black-box* mempunyai kode pengujian yang dapat dilihat pada Gambar 7.1.

7.1.1 Pengujian *Black-Box Login*

Pengujian *black-box login* dilakukan untuk menguji fungsi *login* yang ada pada Sistem Informasi Kehadiran Siswa berdasarkan pada alur *use case* login. Pengujian *black-box login* dilakukan oleh ibu Dina Apriliyati sebagai Pemonitor. Pada Tabel 7.1 menunjukkan skenario pengujian untuk mengidentifikasi kasus uji berdasarkan pada alur *use case login.*

Tabel 7.1 Matriks Skenario Pengujian *Black-Box Login*

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	*Basic Flow*	-

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box login* yang ditunjukkan oleh Tabel 7.2. Pengujian pada fungsi *login* dilakukan berdasarkan pada skenario 1 pada Tabel 7.1 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Pemonitor. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat menampilkan informasi sesuai dengan hak akses yang dimiliki oleh pengguna yang melakukan *login*.

Tabel 7.2 Rencana Pengujian dan Kasus Uji *Black-Box Login*

Kode Pengujian	VT-01	
Nomor Skenario	Skenario 1	
Kode Persyaratan	FS-SIKA-03	
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem dapat menampilkan informasi sesuai dengan hak akses pengguna.	
Prosedur Uji	1. Penguji mengakses Sistem Informasi Kehadiran Siswa. 2. Penguji memasukkan *username* dan *password*.	
Masukkan (*Input*)	*Username*	pemonitor
	Password	pemonitor
Hasil yang diharapkan	Sistem menampilkan informasi sesuai dengan hak akses dari Pemonitor yaitu dengan tampilan pada *sidebar* berupa *dashboard,* data siswa, dan rekap data.	
Hasil Pengujian	Sistem berhasil menampilkan informasi sesuai dengan hak akses dari Pemonitor yaitu dengan tampilan pada *sidebar* berupa *dashboard,* data siswa, dan rekap data.	
Status Uji	Valid	

7.1.2 Pengujian *Black-Box* Mendata Kehadiran Siswa

Pengujian *black-box* mendata kehadiran siswa dilakukan untuk menguji fungsi menyimpan dan mengubah data kehadiran siswa yang ada pada Sistem Informasi Kehadiran Siswa. Pengujian dilakukan oleh ibu Dessy Akhriani sebagai Guru Kelas berdasarkan pada alur *use case* mendata kehadiran siswa. Pada Tabel 7.3 menunjukkan skenario pengujian untuk mengidentifikasi kasus uji berdasarkan pada alur *use case* mendata kehadiran siswa.

Tabel 7.3 Matriks Skenario Pengujian *Black-Box* Mendata Kehadiran Siswa

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	*Basic Flow*	-

Tabel 7.3 Matriks Skenario Pengujian *Black-Box* Mendata Kehadiran Siswa (lanjutan)

No. Skenario	Alur Awal	Alur Alternatif
Skenario 2	*Basic Flow*	A5. Menangani Kegagalan Penyimpan Data
Skenario 3	*Basic Flow*	A9. Mengubah Data Kehadiran

7.1.2.1 Menyimpan Data Kehadiran

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box* menyimpan data kehadiran siswa yang ditunjukkan oleh Tabel 7.4. Pengujian pada fungsi menyimpan data kehadiran siswa dilakukan berdasarkan pada skenario 1 pada Tabel 7.3 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Guru Kelas. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat menyimpan data kehadiran siswa.

Tabel 7.4 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menyimpan Data Kehadiran

Kode Pengujian	VT-02	
Nomor Skenario	Skenario 1	
Kode Persyaratan	FS-SIKA-06	
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem dapat menyimpan data kehadiran siswa.	
Prosedur Uji	1. Penguji teridentifikasi sebagai Guru Kelas. 2. Penguji memilih pendataan pada opsi data kehadiran. 3. Penguji memilih tanggal dan kelas yang akan didata kehadirannya. 4. Penguji memilih fungsi tampilkan. 5. Penguji memasukkan data kehadiran siswa. 6. Penguji memilih fungsi simpan.	
Masukkan (*Input*)	Pilih Kelas	X-MIA-1
	Pilih Tanggal	2019-01-21
Hasil yang diharapkan	Sistem menampilkan pesan data berhasil disimpan dan data tersedia di *database*.	
Hasil Pengujian	Sistem berhasil menampilkan pesan data berhasil disimpan dan data tersedia di *database*.	
Status Uji	Valid	

7.1.2.2 Menyimpan Data Kehadiran Alternatif A5 (Menangani Kegagalan Penyimpan Data)

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box* menyimpan data kehadiran siswa alternatif A5 (menangani kegagalan menyimpan data) yang ditunjukkan oleh Tabel 7.5. Pengujian pada fungsi menyimpan data kehadiran siswa alternatif A5 dilakukan berdasarkan pada skenario 2 pada Tabel 7.3 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Guru Kelas. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat menampilkan peringatan data gagal disimpan ketika data yang dimasukkan telah tersimpan sebelumnya di dalam *database*.

Tabel 7.5 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menyimpan Data Kehadiran Alternatif A5 (Menangani Kegagalan Penyimpan Data)

Kode Pengujian	VT-03	
Nomor Skenario	Skenario 2	
Kode Persyaratan	FS-SIKA-06	
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem menampilkan peringatan data gagal disimpan jika sebelumnya data yang dimasukkan telah tersimpan di *database.*	
Prosedur Uji	1. Penguji teridentifikasi sebagai Guru Kelas. 2. Penguji memilih pendataan pada opsi data kehadiran. 3. Penguji memilih tanggal dan kelas yang akan didata kehadirannya. 4. Penguji memilih fungsi tampilkan. 5. Penguji memasukkan data kehadiran siswa. 6. Penguji memilih fungsi simpan.	
Masukkan (*Input*)	Pilih Kelas	X-MIA-1
	Pilih Tanggal	2019-01-21
Hasil yang diharapkan	Sistem menampilkan pesan data gagal disimpan.	
Hasil Pengujian	Sistem berhasil menampilkan pesan data gagal disimpan.	
Status Uji	Valid	

7.1.2.3 Mengubah Data Kehadiran

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box* mengubah data kehadiran siswa yang ditunjukkan oleh Tabel 7.6. Pengujian pada fungsi mengubah data kehadiran siswa dilakukan berdasarkan pada skenario 3 pada Tabel 7.3 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Guru Kelas. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat mengubah data kehadiran siswa yang telah tersimpan di dalam *database*.

Tabel 7.6 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Mengubah Data Kehadiran

Kode Pengujian	VT-04
Nomor Skenario	Skenario 3
Kode Persyaratan	FS-SIKA-08
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem dapat mengubah data kehadiran siswa.
Prosedur Uji	1. Penguji teridentifikasi sebagai Guru Kelas. 2. Penguji memilih lihat data pada opsi data kehadiran. 3. Penguji memilih tanggal dan kelas yang akan diubah kehadirannya. 4. Penguji memilih fungsi tampilkan. 5. Penguji memilih edit pada data yang akan diubah. 6. Penguji mengubah data kehadiran. 7. Penguji memilih fungsi simpan.
Hasil yang diharapkan	Sistem menampilkan pesan data berhasil diubah dan data di *database* terubah.
Hasil Pengujian	Sistem berhasil menampilkan pesan data berhasil diubah dan data di *database* terubah.
Status Uji	Valid

7.1.3 Pengujian *Black-Box* Memonitor Rekap Data Kehadiran

Pengujian *black-box* memonitor rekap data kehadiran dilakukan untuk menguji fungsi melihat dan mengunduh rekap data kehadiran siswa yang ada pada Sistem Informasi Kehadiran Siswa. Pengujian dilakukan oleh ibu Dina Apriliyati sebagai Pemonitor berdasarkan pada alur *use case* memonitor rekap data kehadiran. Pada Tabel 7.7 menunjukkan skenario pengujian untuk mengidentifikasi kasus uji berdasarkan pada alur *use case* memonitor rekap data kehadiran.

Tabel 7.7 Matriks Skenario Pengujian *Black-Box* Memonitor Rekap Data Kehadiran

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	*Basic Flow*	-
Skenario 2	*Basic Flow*	A1. Menangani Kondisi Data yang Belum Tersedia.
Skenario 3	*Basic Flow*	A3.Mengunduh Rekap Data Kehadiran

7.1.3.1 Melihat Rekap Data Kehadiran

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box* melihat rekap data kehadiran siswa yang ditunjukkan oleh Tabel 7.8. Pengujian pada fungsi melihat rekap data kehadiran siswa dilakukan berdasarkan pada skenario 1 pada Tabel 7.7 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Pemonitor. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat menampilkan rekap data kehadiran siswa.

Tabel 7.8 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Melihat Rekap Data Kehadiran

Kode Pengujian	VT-05	
Nomor Skenario	Skenario 1	
Kode Persyaratan	FS-SIKA-10	
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem dapat menampilkan rekap data kehadiran siswa.	
Prosedur Uji	1. Penguji teridentifikasi sebagai Pemonitor. 2. Penguji memilih lihat rekap pada opsi rekap data. 3. Penguji memasukkan data pilih tanggal, kelas, dan jumlah hari efektif sekolah. 4. Penguji memilih fungsi tampilkan.	
Masukkan (*Input*)	Pilih Kelas	X-MIA-1
	Pilih Dari Tanggal	2019-01-21
	Pilih Sampai Tanggal	2019-01-25
	Hari Efektif sekolah	5
Hasil yang diharapkan	Sistem menampilkan rekap data kehadiran siswa sesuai dengan kriteria data yang dimasukkan (*input*).	
Hasil Pengujian	Sistem berhasil menampilkan rekap data kehadiran siswa sesuai dengan kriteria data yang dimasukkan (*input*).	

Tabel 7.8 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Melihat Rekap Data Kehadiran (lanjutan)

Status Uji	Valid

7.1.3.2 Melihat Rekap Data Kehadiran dengan Alternatif A1 (Menangani Kondisi Data yang Belum Tersedia)

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box* melihat rekap data kehadiran dengan alternatif A1 (menangani kondisi yang belum tersedia) yang ditunjukkan oleh Tabel 7.9. Pengujian pada fungsi melihat rekap data kehadiran siswa dengan alternatif A1 dilakukan berdasarkan pada skenario 2 pada Tabel 7.7 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Pemonitor. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat menampilkan halaman rekap data kehadiran siswa dengan keterangan data belum tersedia ketika data terkait kehadiran siswa yang ingin ditampilkan belum tersedia di *database*.

Tabel 7.9 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Melihat Rekap Data Kehadiran dengan Alternatif A1

Kode Pengujian	VT-06	
Nomor Skenario	Skenario 2	
Kode Persyaratan	FS-SIKA-10	
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem dapat menangani kondisi data yang belum tersedia.	
Prosedur Uji	1. Penguji teridentifikasi sebagai Pemonitor. 2. Penguji memilih lihat rekap pada opsi rekap data. 3. Penguji memasukkan data pilih tanggal, kelas, dan jumlah hari efektif sekolah. 4. Penguji memilih fungsi tampilkan.	
Masukkan (*Input*)	Pilih Kelas	X-MIA-1
	Pilih Dari Tanggal	2019-01-28
	Pilih Sampai Tanggal	2019-02-01
	Hari Efektif sekolah	5
Hasil yang diharapkan	Sistem menampilkan halaman rekap data kehadiran siswa dengan keterangan data belum tersedia.	
Hasil Pengujian	Sistem berhasil menampilkan halaman rekap data kehadiran siswa dengan keterangan data belum tersedia.	
Status Uji	Valid	

7.1.3.3 Mengunduh Rekap Data Kehadiran

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box* mengunduh rekap data kehadiran siswa yang ditunjukkan oleh Tabel 7.10. Pengujian pada fungsi mengunduh rekap data kehadiran siswa dilakukan berdasarkan pada skenario 3 pada Tabel 7.7 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Pemonitor. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat mengunduh rekap data kehadiran siswa.

Tabel 7.10 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Mengunduh Rekap Data Kehadiran

Kode Pengujian	VT-07	
Nomor Skenario	Skenario 3	
Kode Persyaratan	FS-SIKA-11	
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem dapat mengunduh rekap data kehadiran siswa.	
Prosedur Uji	1. Penguji teridentifikasi sebagai Pegawai Pemonitor. 2. Penguji memilih cetak rekap pada opsi rekap data. 3. Penguji memasukkan data pilih tanggal, kelas, dan jumlah hari efektif sekolah. 4. Penguji memilih fungsi *export to* pdf.	
Masukkan (*Input*)	Pilih Kelas	X-MIA-1
	Pilih Dari Tanggal	2019-01-21
	Pilih Sampai Tanggal	2019-01-25
	Hari Efektif sekolah	5
Hasil yang diharapkan	Sistem mengunduh rekap data kehadiran sesuai dengan kriteria data yang dimasukkan (*input*).	
Hasil Pengujian	Sistem berhasil mengunduh rekap data kehadiran sesuai dengan kriteria data yang dimasukkan (*input*).	
Status Uji	Valid	

7.1.4 Pengujian *Black-Box* Mengelola Data Siswa

Pengujian *black-box* mengelola data siswa dilakukan untuk menguji fungsi menambah, mengubah dan menghapus data siswa yang ada pada Sistem Informasi Kehadiran Siswa. Pengujian dilakukan oleh ibu Dina Apriliyati sebagai Pegawai Administrasi Umum berdasarkan pada alur *use case* mengelola data siswa. Pada Tabel 7.11 menunjukkan skenario pengujian untuk mengidentifikasi kasus uji berdasarkan pada alur *use case* mengelola data siswa.

Tabel 7.11 Matriks Skenario Pengujian *Black-Box* Mengelola Data Siswa

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	*Basic Flow*	-
Skenario 2	*Basic Flow*	A1. Mengubah Data Siswa
Skenario 3	*Basic Flow*	A2. Menghapus Data Siswa

7.1.4.1 Menambah Data Banyak Siswa

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box* menambah data banyak siswa yang ditunjukkan oleh Tabel 7.12. Pengujian pada fungsi menambah data banyak siswa dilakukan berdasarkan pada skenario 1 pada Tabel 7.11 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Pegawai Administrasi Umum. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat menambah data banyak siswa.

Tabel 7.12 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menambah Data Banyak Siswa

Kode Pengujian	VT-08
Nomor Skenario	Skenario 1
Kode Persyaratan	FS-SIKA-14
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem dapat menambah data siswa dalam jumlah banyak.
Prosedur Uji	1. Penguji teridentifikasi sebagai Pegawai Administrasi Umum. 2. Penguji memilih opsi data siswa. 3. Penguji memilih fungsi tambah banyak data. 4. Penguji mengunduh format pengisian data. 5. Penguji mengisi informasi biodata siswa. 6. Penguji menyimpan data siswa dalam format xlsx. 7. Penguji mengunggah data siswa dalam format xlsx. 8. Penguji memilih fungsi preview. 9. Penguji memilih fungsi *import*.
Hasil yang diharapkan	Sistem menampilkan pesan data berhasil disimpan dan data tersedia di *database*.
Hasil Pengujian	Sistem berhasil menampilkan pesan data berhasil disimpan dan data tersedia di *database*.

Status Uji	Valid

7.1.4.2 Mengubah Data Siswa

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box* mengubah data siswa yang ditunjukkan oleh Tabel 7.13. Pengujian pada fungsi mengubah data siswa dilakukan berdasarkan pada skenario 2 pada Tabel 7.11 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Pegawai Administrasi Umum. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat mengubah data siswa yang sebelumnya telah tersimpan di dalam *database*.

Tabel 7.13 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Mengubah Data Siswa

Kode Pengujian	VT-09
Nomor Skenario	Skenario 2
Kode Persyaratan	FS-SIKA-16
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem dapat mengubah data siswa.
Prosedur Uji	1. Penguji teridentifikasi sebagai Pegawai Administrasi Umum. 2. Penguji memilih opsi data siswa. 3. Penguji memilih fungsi *edit*. 4. Penguji mengubah data siswa. 5. Penguji memilih fungsi *update*.
Hasil yang diharapkan	Sistem menampilkan pesan data berhasil diubah dan data di *database* terubah.
Hasil Pengujian	Sistem berhasil menampilkan pesan data berhasil diubah dan data di *database* terubah.
Status Uji	Valid

7.1.4.3 Menghapus Data Siswa

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box* menghapus data siswa yang ditunjukkan oleh Tabel 7.14. Pengujian pada fungsi menghapus data siswa dilakukan berdasarkan pada skenario 3 pada Tabel 7.11 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Pegawai Administrasi Umum. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat menghapus data siswa yang sebelumnya telah tersimpan di dalam *database*.

Tabel 7.14 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menghapus Data Siswa

Kode Pengujian	VT-10
Nomor Skenario	Skenario 3
Kode Persyaratan	FS-SIKA-15
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem dapat menghapus data siswa.
Prosedur Uji	1. Penguji teridentifikasi sebagai Pegawai Administrasi Umum. 2. Penguji memilih opsi data siswa. 3. Penguji memilih fungsi hapus pada kolom aksi. 4. Penguji memilih fungsi hapus pada pesan peringatan yang ditampilkan .
Hasil yang diharapkan	Sistem menampilkan pesan data berhasil dihapus dan data tidak tersedia di *database*.
Hasil Pengujian	Sistem berhasil menampilkan pesan data berhasil dihapus dan data tidak tersedia di *database*.
Status Uji	Valid

7.1.5 Pengujian *Black-Box* Memonitor Kehadiran Siswa Secara Menyeluruh

Pengujian *black-box* memonitor kehadiran siswa secara menyeluruh dilakukan untuk menguji fungsi menampilkan data kehadiran siswa dari tiap tingkatan kelas secara menyeluruh pada Sistem Informasi Kehadiran Siswa. Pengujian dilakukan oleh ibu Dina Apriliyati sebagai Pemonitor berdasarkan pada alur *use case* memonitor kehadiran siswa secara menyeluruh. Pada Tabel 7.15 menunjukkan skenario pengujian untuk mengidentifikasi kasus uji berdasarkan pada alur *use case* memonitor kehadiran siswa secara menyeluruh.

Tabel 7.15 Matriks Skenario Pengujian *Black-Box* Memonitor Kehadiran Siswa Secara Menyeluruh

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	*Basic Flow*	-
Skenario 2	*Basic Flow*	A1. Menangani Kondisi Data yang Belum Tersedia.

7.1.5.1 Menampilkan Data Kehadiran Siswa Secara Menyeluruh

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box* menampilkan data kehadiran siswa secara menyeluruh yang ditunjukkan oleh Tabel 7.16. Pengujian pada fungsi menampilkan data kehadiran siswa secara menyeluruh dilakukan berdasarkan pada skenario 1 pada Tabel 7.15 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Pemonitor. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat menampilkan data kehadiran siswa dari setiap tingkatan kelas secara menyeluruh.

Tabel 7.16 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menampilkan Data Kehadiran Siswa Secara Menyeluruh

Kode Pengujian	VT-11
Nomor Skenario	Skenario 1
Kode Persyaratan	FS-SIKA-17
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem dapat menampilkan data kehadiran siswa secara menyeluruh dari setiap tingkatan kelas.
Prosedur Uji	1. Penguji teridentifikasi sebagai Pemonitor. 2. Penguji memilih opsi *dashboard.*
Hasil yang diharapkan	Sistem menampilkan data kehadiran siswa secara menyeluruh dari setiap tingkatan kelas dengan tampilan berupa *pie chart.*
Hasil Pengujian	Sistem berhasil menampilkan data kehadiran siswa secara menyeluruh dari setiap tingkatan kelas dengan tampilan berupa *pie chart.*
Status Uji	Valid

7.1.5.2 *Menampilkan Data Kehadiran Siswa Secara Menyeluruh* dengan Alternatif A1 (*Menangani Kondisi Data yang Belum Tersedia*)

Pada bagian ini menjelaskan mengenai rencana pengujian dan kasus uji *black-box* menampilkan data kehadiran siswa secara menyeluruh dengan alternatif A1 (menangani kondisi data yang belum tersedia) yang ditunjukkan oleh Tabel 7.17. Pengujian pada fungsi menampilkan data kehadiran siswa secara menyeluruh dengan Alternatif A1 dilakukan berdasarkan pada skenario 2 pada Tabel 7.15 dan dilakukan oleh penguji yang telah teridentifikasi sebagai Pemonitor. Pengujian dilakukan untuk memastikan Sistem Informasi Kehadiran Siswa dapat menampilkan keterangan data belum tersedia ketika data yang akan ditampilkan belum tersedia di dalam *database*.

Tabel 7.17 Rencana Pengujian dan Kasus Uji *Black-Box* Fungsi Menampilkan Data Kehadiran Siswa Secara Menyeluruh dengan Alternatif A1

Kode Pengujian	VT-12
Nomor Skenario	Skenario 2
Kode Persyaratan	FS-SIKA-17
Tujuan Pengujian	Dilakukan pengujian untuk memastikan sistem dapat menampilkan keterangan data belum tersedia pada halaman *dashboard* jika data belum tersedia di *database.*
Prosedur Uji	1. Penguji teridentifikasi sebagai Pegawai Pemonitor. 2. Penguji memilih opsi *dashboard.*
Hasil yang diharapkan	Sistem menampilkan halaman *dashboard* dengan keterangan data belum tersedia di sistem.
Hasil Pengujian	Sistem berhasil menampilkan halaman *dashboard* dengan keterangan data belum tersedia di sistem.
Status Uji	Valid

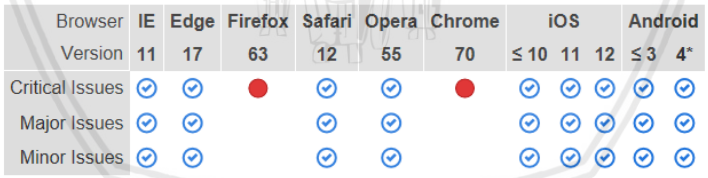
7.2 Pengujian Kompatibilitas

Pada Sistem Informasi Kehadiran Siswa yang dikembangkan, selain dilakukan pengujian validasi dilakukan juga pengujian kompatibilitas untuk memastikan sistem yang dikembangkan dapat berjalan pada beberapa peramban yang berbeda dan sistem dapat berfungsi dengan baik tanpa adanya konten yang menghilang. Pengujian kompatibilitas dilakukan berdasarkan persyaratan nonfungsional sistem yaitu NFS-SIKA-01. Beberapa jenis peramban yang digunakan dalam pengujian kompatibiltas Sistem Informasi Kehadiran Siswa ditunjukkan oleh Tabel 7.18.

Tabel 7.18 Jenis Peramban Pada Pengujian Kompatibilitas

Nama Peramban	Versi Peramban
Internet Explorer	11
Microsoft Edge	17
Firefox	63
Safari	12
Opera	55
Chrome	70
iOS	≤10, 11, 12
Android	≤3, 4”

Aplikasi yang digunakan untuk pengujian kompablitas adalah SortSite dengan versi 5.31.830.0. Pada aplikasi SortSite yang digunakan untuk melakukan pengujian kompatibilitas terdapat 3 kategori masalah yang digunakan yaitu *critical issues*, *major issues*, dan *minor issues*. Pada kategori *critical issues* yang dilambangkan dengan lingkaran berwarana merah penuh yang menggambarkan adanya konten atau fungsi sistem yang hilang pada peramban. Pada kategori *major issues* dilambangkan dengan lingkaran berwarna jingga sebanyak tiga per empat lingkaran yang menggambarkan adanya masalah utama pada tampilan atau performa dari sistem di peramban. Pada kategori *minor issues* dilambangkan dengan lingkaran berwarna jingga sebanyak setengah lingkaran yang menggambarkan adanya masalah minor pada tampilan atau performa dari sistem di peramban.



Gambar 7.2 Hasil Pengujian Kompatibilitas

Pada Gambar 7.2 menunjukkan hasil pengujian kompatibilitas pada Sistem Informasi Kehadiran Siswa dengan menggunakan aplikasi SortSite. Dari hasil pengujian yang telah dilakukan terdapat *critical issues* pada peramban Firefox dan Chrome, hal ini terjadi karena pada peramban Firefox versi 63 dan Chrome versi 70 terdapat peringatan keamanan pada kata sandi (*password*) yang tidak menggunakan *Hypertext Transfer Protocol Secure* (HTTPS). Pada saat pengujian Sistem Informasi Kehadiran Siswa masih menggunakan *Hypertext Transfer Protocol* (HTTP) sehingga terjadinya *critical issues*.

BAB 8 PENUTUP

8.1 Kesimpulan

Berdasarkan pada penelitian yang telah dilakukan, dapat disimpulkan bahwa Sistem Informasi Kehadiran Siswa pada SMA Suluh Jakarta dapat dikembangkan dengan melakukan kegiatan pengembangan sistem. Dari kegiatan pengembangan sistem yang dilakukan, diperoleh hasil sebagai berikut:

1. Dari analisis persyaratan yang telah dilakukan diperoleh hasil berupa identifikasi tipe pemangku kepentingan yang nantinya akan menggunakan Sistem Informasi Kehadiran siswa, serta analisis permasalahan yang dihadapi oleh pemangku kepentingan. Dari analisis permasalahan kemudian dilakukan identifikasi dan diperoleh 8 kebutuhan pemangku kepentingan dan pengguna terkait dengan pengelolaan data kehadiran siswa. Hasil dari identifikasi kebutuhan digunakan untuk melakukan identifikasi fitur yang nantinya ada pada Sistem Informasi Kehadiran Siswa dan diperoleh hasil 6 fitur, agar fitur pada Sistem Informasi Kehadiran Siswa dapat berjalan terdapat kemampuan yang harus dimiliki oleh sistem yaitu persyaratan fungsional sistem dan diperoleh hasil 24 persyaratan fungsional, untuk mendukung Sistem Informasi Kehadiran Siswa yang dikembangkan terdapat persyaratan nonfungsional sistem dan diperoleh hasil 1 persyaratan nonfungsional. Kemudian interaksi yang dapat dilakukan aktor terhadap Sistem Informasi Kehadiran Siswa berupa pemodelan *use* case, diperoleh hasil 10 *use case* dan 5 aktor pengguna.
2. Dari perancangan sistem yang telah dilakukan diperoleh pemodelan interaksi yang terdapat pada Sistem Informasi Kehadiran Siswa menggunakan pola MVC, pemodelan objek berupa kelas-kelas yang digunakan dalam pengembangan berupa diagram kelas analisis dan perancangan, interaksi antar objek yang digambarkan dengan *sequence diagram*, relasi antar tabel-tabel yang digunakan oleh sistem digambarkan kedalam *physical data model*, operasi-operasi logika dari fungsi sistem dituliskan kedalam *pseudocode*, dan tampilan-tampilan sistem digambarkan dalam bentuk sketsa. Hasil dari perancangan Sistem Informasi Kehadiran Siswa berupa, 16 *sequence diagram*, 5 diagram kelas analisis, 9 diagram kelas perancangan *controller,* 6 diagram kelas perancangan *model,* 27 diagram kelas perancangan *view,* dan 5 tabel pada *physical data model*.
3. Hasil dari analisis persyaratan dan perancangan yang telah dilakukan dapat dikembangkannya Sistem Informasi Kehadiran Siswa untuk digunakan pada SMA Suluh Jakarta dengan menggunakan kerangka kerja Codeigniter dan berbasis web. Pada Sistem Informasi Kehadiran Siswa terdapat beberapa fitur yang dapat digunakan, yaitu fitur memonitor kehadiran siswa secara menyeluruh, mendata kehadiran siswa, mengelola data siswa, memonitor rekap data kehadiran, dan mengelola data *master*.

4. Dari sistem yang dikembangkan kemudian dilakukan pengujian dengan menggunakan pengujian validasi dan kompatibilitas pada beberapa fungsi yang ada pada login, mendata kehadiran siswa, memonitor rekap data kehadiran, mengelola data siswa dan memonitor kehadiran siswa secara menyeluruh. Hasil dari pengujian validasi yang dilakukan berdasarkan pada persyaratan fungsional dinyatakan valid. Sedangkan pada pengujian kompatibilitas diperoleh hasil berupa *critical issues* di peramban Chrome versi 70 dan Firefox versi 63 pada bagian keamanan sistem. Hal ini terjadi karena Sistem Informasi Kehadiran Siswa saat diuji masih menggunakan protokol HTTP sedangkan pada peramban Chrome dan Firefox untuk sistem keamanan kata sandi atau *password* tanpa menggunakan protokol HTTPS akan menimbulkan peringatan.

8.2 Saran

Pemberian saran dilakukan sebagai bahan pertimbangan dalam mengembangkan Sistem Informasi Kehadiran Siswa lebih lanjut pada SMA Suluh Jakarta, antara lain sebagai berikut:

1. Sistem Informasi Kehadiran Siswa perlu dikembangkan lebih lanjut dengan memanfaatkan penggunaan berbagai perangkat pendukung sistem presensi sehingga proses memasukkan data ke dalam sistem tidak perlu lagi dilakukan.
2. Perlu adanya pengembangan sistem lebih lanjut berupa penambahan hak akses bagi orang tua siswa untuk dapat memantau kehadiran anaknya disekolah, serta orang tua siswa dapat memberikan surat ketidakhadiran siswa melalui sistem, sehingga orang tua siswa tidak perlu datang kesekolah atau menunggu siswa dapat kembali bersekolah lagi.

DAFTAR REFERENSI

1keydata, 2009. *Physical Data Model.* [Online] Tersedia di: <https://www.1keydata.com/datawarehousing/physical-data-model.html> [Diakses 26 Februari 2019].

A Comparison Between Three SDLC Models Waterfall, 12(1), pp. 106-111. [Online] Tersedia di: <https://www.ijcsi.org/papers/IJCSI-12-1-1-106-111.pdf> [Diakses 09 Februari 2019].

Aryasworo, W. V., Aknuranda, I. & Pramono, D., 2018. Pengembangan Sistem Informasi Pencatatan dan Pemantauan Distribusi. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* 2(4), pp. 1534-1541. [e-journal]. Tersedia melalui: <http://jtiik.ub.ac.id> [Diakses 09 Februari 2019]

Bittner, K. & Spence, I., 2002. *Use Case Modeling.* s.l.:Addison Wesley.

Booch, G. et al., 2007. *Object-Oriented Analysis and Design with Applications.* 3rd penyunt. s.l.:Addision Wesley.

Booch, G., Rumbaugh, J. & Jacobson, I., 2005. *The Unified Modeling Language User Guide.* 2nd penyunt. s.l.:Addison Wesley Professional.

Erinton, R., Negara, R. M. & Sanjoyo, D. D., 2017. ANALISIS PERFORMASI FRAMEWORK CODEIGNITER DAN LARAVEL. e-Proceeding of Engineering, Volume 4, pp. 3565-3572. [e-journal] Tersedia melalui: <https://openlibrary.telkomuniversity.ac.id> [Diakses 09 Februari 2019]

Inc, A., 2013. *Physical Data Model (PDM)s: An Agile Introduction.* [Online] Tersedia di: <http://agilemodeling.com/artifacts/physicalDataModel.htm> [Diakses 13 Februari 2019].

Istanto, S. H. R., Aknuranda, I. & Saputra, M. C., 2018. Pengembangan Sistem Informasi Penggajian Pegawai dan Dokumentasi. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* 2(11), pp. 4886-4894. [e-journal]. Tersedia melalui: <http://jtiik.ub.ac.id> [Diakses 09 Februari 2019]

Laudon, K. C. & Laudon, J. P., 2014. *Management Information Systems Managing the Digital Firm Thirteenth Edition.* s.l.:s.n. [e-book] Tersedia Melalui: <http://dinus.ac.id/repository/docs/ajar/Kenneth_C.Laudon,Jane_P_.Laudon_-_Management_Information_Sysrem_13th_Edition_.pdf> [Diakses 01 September 2018]

Mishra, A. & Dubey, D., 2013. A Comparative Study of Different Software Development Life Cycle Models in. *International Journal of Advance Research in,* 1(5), pp. 64-69. [online] Tersedia di: <https://www.researchgate.net/publication/289526047_A_Comparative_Study_of_Different_Software_Development_Life_Cycle_Models_in_Different_Scenarios> [Diakses 09 Februari 2019]

Monk, E. & Wagner, B., 2013. *Concepts in Enterprise Resource Planning.* 4th penyunt. s.l.:Course Technology, Cengage Learning. [e-book] Tersedia

Melalui:<http://abedinionline.com/Docs/Concepts_in_Enterprise Resource_Planning.pdf> [Diakses 09 Februari 2019]

Mulyadi, 2016. *Bimbingan Konseling di Sekolah dan Madrasah.* Jakarta: Prenadamedia Group.

Object Management Group, 2005. *Unified Modeling Language: Superstructure.* 2nd penyunt. s.l.:s.n.

Object Management Group, 2013. *Business Process Model and Notation Version 2.0.2.* s.l.:s.n.

O'Brien, J. A. & Marakas, G. M., 2011. *Management Information Systems.* 10th penyunt. New York: McGraw-Hill Irwin.

Pargiyani, A., Priyambadha, B. & Arwan, A., 2019. Pengembangan Sistem Informasi Akademik Pada Sekolah Dasar Islam. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* 3(2), pp. 2121-2127. [e-journal]. Tersedia melalui: <http://jtiik.ub.ac.id> [Diakses 09 Februari 2019]

PowerMapper, 2019. *SortSite - Browser Compatibility Tests.* [Online] Tersedia di:<https://www.powermapper.com/products/sortsite/checks/browser-compatibility/> [Diakses 26 Februari 2019].

Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach.* 7th penyunt. New York: McGraw-Hill.

Raharjo, B., 2018. *Belajar Otodidak Framework Codeigniter Teknik Pemrograman Web dengan PHP 7 dan Framewok Codeigniter.* Bandung: INFORMATIKA.

Sommerville, I., 2011. *Software Engineering.* 9th penyunt. United States: Addison-Wesley.

Weske, M., 2007. *Business Process Management.* s.l.:Springer. [e-book] Tersedia Melalui: <https://epdf.tips/download/business-process-modeling-concepts-methods-technology.html> [Diakses 09 Februari 2019]