



PENGEMBANGAN SISTEM PENILAIAN SANTRI TERPADU DI AL-IZZAH LEADERSHIP SCHOOL BERBASIS WEB

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh :

Muhammad Zaharuddin

NIM: 155150200111261



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

MALANG

2020



PENGESAHAN

PENGEMBANGAN SISTEM PENILAIAN SANTRI TERPADU DI AL-IZZAH LEADERSHIP SCHOOL BERBASIS WEB

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Zaharuddin
NIM: 155150200111261

Skrripsi ini telah diuji dan dinyatakan lulus pada
26 Juni 2020

Telah diperiksa dan disetujui oleh:

Pembimbing I

Fajar Fadana, S.T., M.Eng.
NIP: 19871121 201504 1 004

Pembimbing II

Komang Candra Brata, S.Kom., M.T., M.Sc.
NIK: 201607 890711 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri A. Ito Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001



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 Juni 2020



METERAI
TEMPEL
NR. 4/DBAN/603/05493
6000
ETAMZIBURUPIAH

Muhammad Zaharuddin

NIM: 155150200111263

ABSTRAK

Muhammad Zaharuddin, Pengembangan Sistem Penilaian Santri Terpadu Di Al-Izzah Leadership School Berbasis Web

Pembimbing: Fajar Pradana, S.ST., M.Eng., dan Komang Candra Brata, S.Kom., M.T., M.Sc.

Pondok pesantren merupakan salah satu lembaga islam yang menjadi subkultur rakyat indonesia. Pendidikan akademik maupun religi menjadi perhatian penting bagi pondok pesantren dalam membentuk insan yang berpengetahuan luas. Al-Izzah *Leadership School* menjadi salah satu lembaga pendidikan yang memperhatikan santrinya dalam bidang akademik dan juga hafalan Qur'an mereka. Proses pembelajaran setiap semester menghasilkan nilai-nilai yang menggambarkan kesuksesan proses pembelajaran selama satu semester. Akan tetapi, pada proses pembelajaran pada ditemukan permasalahan diantaranya adalah kesulitan yang dialami wali santri dalam mengamati perkembangan nilai akademik dan hafalan Qur'an, sehingga orang tua tidak dapat menegur langsung ketika anak mendapat nilai yang kurang baik. Berdasarkan permasalahan tersebut maka dibuatlah sebuah sistem untuk mempermudah proses penilaian pada pondok pesantren Al-Izzah. Proses pengembangan sistem meliputi tahapan analisis kebutuhan, implementasi, dan pengujian. Tahapan analisis kebutuhan mendapatkan 34 kebutuhan fungsional. Tahapan implementasi menggunakan bahasa pemrograman PHP pada kerangka kerja Laravel dan MySQL pada bagian basis data. Tahapan pengujian meliputi pengujian fungsional dan non fungsional. Pengujian fungsional meliputi pengujian unit yang berhasil melakukan jalur pengujian berdasarkan 3 kasus uji, pengujian integrasi pada method menghasilkan status valid dan pengujian validasi menghasilkan 100% valid dari 52 kasus uji. Pengujian non fungsional menggunakan usability dengan hasil sistem dapat diterima oleh pengguna dengan skor 72,5.

Kata kunci: Pondok Pesantren, Sekolah Kepemimpinan Al-Izzah, pengembangan sistem, pengujian, Laravel



ABSTRACT

Muhammad Zaharuddin, Development of Web-Based Integrated Santri Assessment System in Al-Izzah Leadership School

Supervisors: Fajar Pradana, S.ST., M.Eng., and Komang Candra Brata, S.Kom., M.T., M.Sc.

Islamic boarding school is one of the Islamic institutions which is a subculture of the Indonesian people. Both academic and religious education is an important concern for Islamic boarding schools to produce knowledgeable students. Al-Izzah Leadership School became one of the educational institutions that paid attention to their students in the academic field and also memorizing their Qur'an. The learning process of each semester produces values that illustrate the success of the learning process for whole one semester. However, the learning process has problems, the problems are the difficulties experienced by the guardians of students in observing their children in case of development of academic values and memorization of the Qur'an, so that parents cannot reprimand directly when the child gets a bad grade. Based on these problems, this system made to facilitate the assessment process at Al-Izzah Islamic boarding school. The system development process includes the requirements analysis, implementation, and testing. Requirement analysis consists of 34 functional requirement. Implementation use the PHP programming language with Laravel framework and MySQL for database section. Testing include functional and non-functional testing. Functional testing includes unit testing that successfully conducted the test path based on 3 test cases, integration testing on the method resulted in valid status and validation testing resulted in 100% valid from 52 test cases. Non-functional testing using usability testing gets result that can be accepted by users with a score of 72.5.

Keywords: *Islamic Boarding School, Al-Izzah Leadership School, system development, testing, Laravel.*



DAFTAR ISI

PENGEMBANGAN SISTEM PENILAIAN SANTRI TERPADU DI AL-IZZAH LEADERSHIP SCHOOL BERBASIS WEB.....	i
PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Kajian Pustaka.....	5
2.2 Rekayasa Perangkat Lunak.....	6
2.3 Pengembangan Perangkat Lunak.....	6
2.3.1 <i>Software Development Life Cycle (SDLC)</i>	6
2.3.2 Tahapan Metode <i>Waterfall</i>	6
2.3.3 Pemodelan Berorientasi Objek.....	8
2.3.4 <i>Unified Modelling Language (UML)</i>	8
2.3.4.1 Use Case Diagram.....	8
2.3.4.2 Sequence Diagram.....	9
2.3.4.3 Class Diagram.....	10
2.4 Pengujian.....	10
2.4.1 <i>Unit Testing</i>	10



2.4.2 <i>Integration Testing</i>	11
2.4.3 <i>Validation Testing</i>	11
2.4.4 <i>Usability Testing</i>	11
2.5 Pondok Pesantren.....	11
2.6 Laravel.....	12
2.7 XAMPP.....	13
BAB 3 METODOLOGI	15
3.1 Studi Literatur.....	15
3.2 Rekayasa Kebutuhan.....	15
3.3 Perancangan Aplikasi.....	16
3.4 Implementasi.....	17
3.5 Pengujian dan Analisis.....	17
3.6 Kesimpulan dan Saran.....	17
BAB 4 ANALISIS KEBUTUHAN	18
4.1 Gambaran Umum Sistem.....	18
4.2 Identifikas Aktor.....	18
4.3 Daftar Kebutuhan Sistem.....	19
4.3.1 Teori Kode Kebutuhan.....	19
4.3.2 Daftar Kebutuhan Fungsional.....	19
4.3.3 Daftar Kebutuhan Non Fungsional.....	22
4.4 Pemodelan Kebutuhan.....	22
4.4.1 Pemodelan <i>Use Case Diagram</i>	22
4.5 <i>Use Case Scenario</i>	23
4.5.1 <i>Login</i>	23
4.5.2 <i>Logout</i>	23
4.5.3 Melihat Data Santri.....	24
4.5.4 Mengubah Data Santri.....	24
4.5.5 Menambah Data Santri.....	25
4.5.6 Menghapus Data Santri.....	25
4.5.7 Melihat Data Pengguna.....	26
4.5.8 Mengubah Data Pengguna.....	26
4.5.9 Menambah Data Pengguna.....	27



4.5.10 Menghapus Data Pengguna.....	23
4.5.11 Melihat Data Mata Pelajaran.....	28
4.5.12 Mengubah Data Mata Pelajaran.....	29
4.5.13 Menambah Data Mata Pelajaran.....	29
4.5.14 Menghapus Data Mata Pelajaran.....	30
4.5.15 Melihat Data Kepemimpinan.....	30
4.5.16 Mengubah Data Kepemimpinan.....	31
4.5.17 Menambah Data Kepemimpinan.....	31
4.5.18 Menghapus Data Kepemimpinan.....	32
4.5.19 Melihat Data Hafalan.....	32
4.5.20 Mengubah Data Hafalan.....	33
4.5.21 Menambah Data Hafalan.....	33
4.5.22 Menghapus Data Hafalan.....	34
4.5.23 Melihat Pelanggaran Santri.....	34
4.5.24 Menambah Pelanggaran Santri.....	35
4.5.25 Melihat Nilai Mata Pelajaran.....	35
4.5.26 Menambah Nilai Mata Pelajaran.....	36
4.5.27 Mengubah Nilai Mata Pelajaran.....	36
4.5.28 Melihat Nilai Kepemimpinan.....	37
4.5.29 Mengubah Nilai Kepemimpinan.....	37
4.5.30 Menambah Nilai Kepemimpinan.....	38
4.5.31 Melihat Nilai Hafalan.....	38
4.5.32 Mengubah Nilai Hafalan.....	39
4.5.33 Menambah Nilai Hafalan.....	39
4.5.34 Mencetak Rapor.....	40
BAB 5 PERANCANGAN DAN IMPLEMENTASI.....	41
5.1 Perancangan.....	41
5.1.1 Perancangan Arsitektur.....	41
5.1.1.1 Sequence Diagram Menambah Nilai Hafalan.....	41
5.1.1.2 Sequence Diagram Mengubah Data Mata Pelajaran.....	42
5.1.1.3 Sequence Diagram Melihat Data Pengguna.....	43
5.1.1.4 Class Diagram Sistem Penilaian Santri.....	43



5.1.2 Perancangan Data	47
5.1.3 Perancangan Komponen	48
5.1.3.1 Perancangan Komponen Method <i>updateprogres</i> dan <i>savenilai</i>	48
5.1.3.2 Perancangan Komponen Method <i>setupUpdateOperation</i>	49
5.1.3.3 Perancangan Komponen Method <i>setupListOperation</i> ...	50
5.1.4 Perancangan Antarmuka	50
5.1.4.1 Perancangan Antarmuka Halaman Nilai Hafalan	50
5.1.4.2 Perancangan Antarmuka Halaman Data Mata Pelajaran	52
5.1.4.3 Perancangan Antarmuka Halaman Data Pengguna	53
5.2 Implementasi Sistem	55
5.2.1 Spesifikasi Sistem	55
5.2.2 Implementasi Basis Data	56
5.2.3 Implementasi Kode Program	56
5.2.3.1 Implementasi Kode Program Method <i>updateprogres()</i> dan <i>savenilai()</i> pada NilaiHafalan	57
5.2.3.2 Implementasi Kode Program Method <i>setupUpdateOperation()</i> pada <i>MataPelajaranCrudController</i> ...	58
5.2.3.3 Implementasi Kode Program Method <i>setupListOperation()</i> pada <i>PenggunaController</i>	59
5.2.4 Implementasi Antarmuka	59
5.2.4.1 Implementasi Antarmuka Halaman Nilai Hafalan	59
5.2.4.2 Implementasi Antarmuka Halaman Data Mata Pelajaran	60
5.2.4.3 Implementasi Antarmuka Melihat Data Pengguna	61
BAB 6 PENGUJIAN	62
6.1 Pengujian Unit	62
6.1.1 Pengujian Unit NilaiHafalan	62
6.1.2 Pengujian Unit <i>Function edit</i> dan <i>update</i>	65
6.1.3 Pengujian Unit <i>Method setupListOperation</i>	68
6.2 Pengujian Integrasi	70
6.3 Pengujian Validasi	73



6.3.1 Pengujian Validasi <i>Login</i>	73
6.3.2 Pengujian Validasi <i>Logout</i>	75
6.3.3 Pengujian Validasi Melihat Data Santri	75
6.3.4 Pengujian Validasi Mengubah Data Santri	76
6.3.5 Pengujian Validasi Menambah Data Santri	76
6.3.6 Pengujian Validasi Menghapus Data Santri	77
6.3.7 Pengujian Validasi Melihat Data Pengguna	78
6.3.8 Pengujian Validasi Mengubah Data Pengguna	78
6.3.9 Pengujian Validasi Menambah Data Pengguna	79
6.3.10 Pengujian Validasi Menghapus Data Pengguna	79
6.3.11 Pengujian Validasi Melihat Data Mata Pelajaran	80
6.3.12 Pengujian Validasi Mengubah Data Mata Pelajaran	80
6.3.13 Pengujian Validasi Menambah Data Mata Pelajaran.....	81
6.3.14 Pengujian Validasi Menghapus Data Mata Pelajaran	81
6.3.15 Pengujian Validasi Melihat Data Kepemimpinan	82
6.3.16 Pengujian Validasi Mengubah Data Kepemimpinan	82
6.3.17 Pengujian Validasi Menambah Data Kepemimpinan.....	83
6.3.18 Pengujian Validasi Menghapus Data Kepemimpinan	84
6.3.19 Pengujian Validasi Melihat Data Hafalan	84
6.3.20 Pengujian Validasi Mengubah Data Hafalan	84
6.3.21 Pengujian Validasi Menambah Data Hafalan.....	85
6.3.22 Pengujian Validasi Menghapus Data Hafalan	86
6.3.23 Pengujian Validasi Daftar Melihat Pelanggaran Santri	86
6.3.24 Pengujian Validasi Menambah Pelanggaran Santri	86
6.3.25 Pengujian Validasi Melihat Nilai Mata Pelajaran	87
6.3.26 Pengujian Validasi Menambah Nilai Mata Pelajaran	87
6.3.27 Pengujian Validasi Mengubah Nilai Mata Pelajaran	89
6.3.28 Pengujian Validasi Melihat Nilai Kepemimpinan	89
6.3.29 Pengujian Validasi Mengubah Nilai Kepemimpinan	89
6.3.30 Pengujian Validasi Menambah Nilai Kepemimpinan	90
6.3.31 Pengujian Validasi Melihat Nilai Hafalan	90
6.3.32 Pengujian Validasi Mengubah Nilai Hafalan	91



6.3.33 Pengujian Validasi Menambah Nilai Hafalan	91
6.3.34 Pengujian Validasi Mencetak Rapor	91
6.4 Pengujian Usabilitas	92
6.4.1 Prosedur Pengujian <i>Usability</i>	92
6.4.2 Analisis dan Hasil Pengujian <i>Usability</i>	92
BAB 7 PENUTUP	94
7.1 Kesimpulan	94
7.2 Saran	94
DAFTAR REFERENSI	95
LAMPIRAN A WAWANCARA	97
LAMPIRAN B RAPOR	100

**DAFTAR TABEL**

Tabel 4.1 Kebutuhan Fungsional Sistem.....	20
Tabel 4.2 Kebutuhan Non Fungsional Sistem.....	22
Tabel 4.3 <i>Use Case Scenario</i> Login.....	23
Tabel 4.4 <i>Use Case Scenario</i> Logout.....	23
Tabel 4.5 <i>Use Case Scenario</i> Melihat Data Santri.....	24
Tabel 4.6 <i>Use Case Scenario</i> Mengubah Data Santri.....	24
Tabel 4.7 <i>Use Case Scenario</i> Menambah Data Santri.....	25
Tabel 4.8 <i>Use Case Scenario</i> Menghapus Data Santri.....	26
Tabel 4.9 <i>Use Case Scenario</i> Melihat Data Pengguna.....	26
Tabel 4.10 <i>Use Case Scenario</i> Mengubah Data Pengguna.....	27
Tabel 4.11 <i>Use Case Scenario</i> Menambah Data Pengguna.....	27
Tabel 4.12 <i>Use Case Scenario</i> Menghapus Data Pengguna.....	28
Tabel 4.13 <i>Use Case Scenario</i> Melihat Data Mata Pelajaran.....	28
Tabel 4.14 <i>Use Case Scenario</i> Mengubah Data Mata Pelajaran.....	29
Tabel 4.15 <i>Use Case Scenario</i> Menambah Data Mata Pelajaran.....	29
Tabel 4.16 <i>Use Case Scenario</i> Menghapus Data Mata Pelajaran.....	30
Tabel 4.17 <i>Use Case Scenario</i> Melihat Data Kepemimpinan.....	30
Tabel 4.18 <i>Use Case Scenario</i> Mengubah Data Kepemimpinan.....	31
Tabel 4.19 <i>Use Case Scenario</i> Menambah Data Kepemimpinan.....	31
Tabel 4.20 <i>Use Case Scenario</i> Menghapus Data Kepemimpinan.....	32
Tabel 4.21 <i>Use Case Scenario</i> Melihat Data Hafalan.....	32
Tabel 4.22 <i>Use Case Scenario</i> Mengubah Data Hafalan.....	33
Tabel 4.23 <i>Use Case Scenario</i> Menambah Data Hafalan.....	33
Tabel 4.24 <i>Use Case Scenario</i> Menghapus Data Hafalan.....	34
Tabel 4.25 <i>Use Case Scenario</i> Melihat Pelanggaran Santri.....	34
Tabel 4.26 <i>Use Case Scenario</i> Menambah Pelanggaran Santri.....	35
Tabel 4.27 <i>Use Case Scenario</i> Melihat Nilai Mata Pelajaran.....	35
Tabel 4.28 <i>Use Case Scenario</i> Menambah Nilai Mata Pelajaran.....	36
Tabel 4.29 <i>Use Case Scenario</i> Mengubah Nilai Mata Pelajaran.....	37
Tabel 4.30 <i>Use Case Scenario</i> Melihat Nilai Kepemimpinan.....	37
Tabel 4.31 <i>Use Case Scenario</i> Mengubah Nilai Kepemimpinan.....	38



Tabel 4.32 <i>Use Case Scenario</i> Menambah Nilai Kepemimpinan	33
Tabel 4.33 <i>Use Case Scenario</i> Mengubah Nilai Hafalan.....	39
Tabel 4.34 <i>Use Case Scenario</i> Menambah Nilai Hafalan.....	39
Tabel 4.35 <i>Use Case Scenario</i> Mencetak Rapor	40
Tabel 5.1 <i>Pseudocode Method Update</i> progres dan <i>Savenilai</i>	49
Tabel 5.2 <i>Pseudocode Method Setup</i> UpdateOperation	49
Tabel 5.3 <i>Pseudocode Method Setup</i> ListOperation	50
Tabel 5.4 Penjelasan Perancangan Antarmuka Halaman Nilai Hafalan.....	51
Tabel 5.5 Penjelasan Perancangan Antarmuka Halaman Data Mata Pelajaran .	52
Tabel 5.6 Penjelasan Perancangan Antarmuka Halaman Data Pengguna.....	54
Tabel 5.7 Spesifikasi Perangkat Keras	55
Tabel 5.8 Spesifikasi Perangkat Lunak	55
Tabel 5.9 <i>Source Code Method Update</i> progres() Pada NilaiHafalan	57
Tabel 5.10 <i>Source Code Method Savenilai</i> () Pada NilaiHafalan.....	57
Tabel 5.11 <i>Source Code Method Setup</i> UpdateOperation() Pada <i>MataPelajaranCrudController</i>	58
Tabel 5.12 <i>Source Code Method Edit</i> () Pada <i>UpdateOperation</i>	58
Tabel 5.13 <i>Source Code Method Update</i> () Pada <i>UpdateOperation</i>	58
Tabel 5.14 <i>Source Code Method Setup</i> ListOperation() Pada <i>PenggunaController</i> 59	
Tabel 6.1 <i>Pseudocode</i> NilaiHafalan.....	62
Tabel 6.2 Hasil Pengujian Unit <i>Method update</i> progres dan <i>savenilai</i>	64
Tabel 6.3 <i>Pseudocode edit</i> dan <i>update</i>	65
Tabel 6.4 Hasil Pengujian Unit <i>method edit</i> dan <i>update</i>	67
Tabel 6.5 <i>Pseudocode setup</i> ListOperation	68
Tabel 6.6 Hasil Pengujian Unit <i>Method setup</i> ListOperation.....	69
Tabel 6.7 <i>Pseudocode</i> Algoritme <i>update</i> pada <i>DraftController</i>	70
Tabel 6.8 Hasil Pengujian Unit <i>Method update</i>	72
Tabel 6.9 Kasus Uji Berhasil Melakukan <i>Login</i>	73
Tabel 6.10 kasus Uji Tidak berhasil Melakukan <i>Login</i> dengan Mengosongkan Kolom <i>Password</i>	74
Tabel 6.11 Kasus Uji Tidak berhasil Melakukan <i>Login</i> dengan Tidak Melakukan Pengisian Kolom <i>E-mail</i>	74



Tabel 6.12 Kasus Uji Tidak berhasil Melakukan <i>Login</i> dengan Masukan yang Tidak Sesuai Pada Kolom <i>E-mail</i>	74
Tabel 6.13 Kasus Uji Tidak berhasil Melakukan <i>Login</i> dengan Masukan yang Tidak Sesuai Pada Kolom <i>Password</i>	75
Tabel 6.14 Kasus Uji Berhasil <i>Logout</i>	75
Tabel 6.15 Kasus Uji Berhasil Melihat Data Santri	75
Tabel 6.16 Kasus Uji Berhasil Mengubah Data Santri.....	76
Tabel 6.17 Kasus Uji Tidak berhasil Mengubah Data Santri.....	76
Tabel 6.18 Kasus Uji Berhasil Menambah Data Santri	76
Tabel 6.19 Kasus Uji Tidak berhasil Menambah Data Santri	77
Tabel 6.20 Kasus Uji Berhasil Menghapus Data Santri.....	77
Tabel 6.21 Kasus Uji Berhasil Melihat Data Pengguna.....	78
Tabel 6.22 Kasus Uji Berhasil Mengubah Data Pengguna.....	78
Tabel 6.23 Kasus Uji Tidak berhasil Mengubah Data Pengguna.....	78
Tabel 6.24 Kasus Uji Berhasil Menambah Data Pengguna.....	79
Tabel 6.25 Kasus Uji Tidak berhasil Menambah Data Pengguna.....	79
Tabel 6.26 Kasus Uji Berhasil Menghapus Data Pengguna.....	79
Tabel 6.27 Kasus Uji Berhasil Melihat Data Mata Pelajaran	80
Tabel 6.28 Kasus Uji Berhasil Mengubah Data Mata Pelajaran.....	80
Tabel 6.29 Kasus Uji Tidak berhasil Mengubah Data Mata Pelajaran.....	80
Tabel 6.30 Kasus Uji Berhasil Menambah Data Mata Pelajaran.....	81
Tabel 6.31 Kasus Uji Tidak berhasil Menambah Data Mata Pelajaran.....	81
Tabel 6.32 Kasus Uji Berhasil Menghapus Data Mata Pelajaran.....	81
Tabel 6.33 Kasus Uji Berhasil Melihat Data Kepemimpinan	82
Tabel 6.34 Kasus Uji Berhasil Mengubah Data Kepemimpinan.....	82
Tabel 6.35 Kasus Uji Tidak berhasil Mengubah Data Kepemimpinan.....	83
Tabel 6.36 Kasus Uji Berhasil Menambah Data Kepemimpinan.....	83
Tabel 6.37 Kasus Uji Tidak berhasil Menambah Data Kepemimpinan	83
Tabel 6.38 Kasus Uji Berhasil Menghapus Data Kepemimpinan.....	84
Tabel 6.39 Kasus Uji Berhasil Melihat Data Hafalan.....	84
Tabel 6.40 Kasus Uji Berhasil Mengubah Data Hafalan.....	84
Tabel 6.41 Kasus Uji Tidak berhasil mengubah data hafalan.....	85
Tabel 6.42 Kasus Uji Berhasil Menambah Data Hafalan.....	85



Tabel 6.43 Kasus Uji Tidak berhasil Menambah Data Hafalan	85
Tabel 6.44 Kasus Uji Berhasil Menghapus Data Hafalan.....	86
Tabel 6.45 Kasus Uji Berhasil Melihat Pelanggaran Santri.....	86
Tabel 6.46 Kasus Uji Berhasil Menambah Pelanggaran Santri.....	86
Tabel 6.47 Kasus Uji Tidak berhasil Menambah Pelanggaran Santri	87
Tabel 6.48 Kasus Uji Berhasil Melihat Nilai Mata Pelajaran	87
Tabel 6.49 Kasus Uji Berhasil Menambah Nilai Mata Pelajaran	87
Tabel 6.50 Kasus Uji Berhasil Menambah Nilai Mata Pelajaran dengan <i>file excel</i>	88
Tabel 6.51 Kasus Uji Berhasil Reset Nilai Mata Pelajaran dengan <i>File Excel</i>	88
Tabel 6.52 Kasus Uji Berhasil Mengubah Nilai Mata Pelajaran	89
Tabel 6.53 Kasus Uji Berhasil Melihat Nilai Kepemimpinan	89
Tabel 6.54 Kasus Uji Berhasil Mengubah Nilai Kepemimpinan	89
Tabel 6.55 Kasus Uji Tidak berhasil Mengubah Nilai Kepemimpinan	90
Tabel 6.56 Kasus Uji Berhasil Menambah Nilai Kepemimpinan	90
Tabel 6.57 Kasus Uji Berhasil Melihat Nilai Hafalan.....	90
Tabel 6.58 Kasus Uji Berhasil Mengubah Nilai Hafalan.....	91
Tabel 6.59 Kasus Uji Berhasil Menambah Nilai Hafalan	91
Tabel 6.60 Kasus Uji Berhasil Mencetak Rapor.....	91
Tabel 6.61 Daftar Pertanyaan Pengujian <i>Usability</i> menggunakan Metode SUS	92
Tabel 6.62 Hasil Kuisiner Responden Terhadap Pertanyaan Metode SUS.....	93
Tabel 6.63 Hasil Perhitungan Skor Akhir Penilaian Kuisiner.....	93

DAFTAR GAMBAR

Gambar 2.1 Waterfall Model (Somerville, 2016).....	7
Gambar 2.2 Contoh <i>Use Case Diagram</i> (Booch, Rumbaugh and Jacobson, 2005)9	
Gambar 2.3 Contoh <i>Sequence Diagram</i> (Booch, Rumbaugh and Jacobson, 2005)9	
Gambar 2.4 Contoh <i>Class Diagram</i> (Booch, Rumbaugh and Jacobson, 2005) ...	10
Gambar 2.5 Ilustrasi MVC (Stauffer, 2017).....	13
Gambar 2.6 Laravel <i>Route</i> dengan <i>Controller</i> (Stauffer, 2017).....	13
Gambar 2.7 XAMPP <i>Control Panel</i>	14
Gambar 3.1 Alur Penelitian.....	15
Gambar 4.1 Gambaran Umum Sistem.....	18
Gambar 4.2 Aturan Penomoran.....	19
Gambar 4.3 <i>Use Case Diagram</i> Sistem Penilaian Santri Terpadu.....	22
Gambar 5.1 <i>Sequence Diagram</i> Menambah Nilai Hafalan.....	42
Gambar 5.2 <i>Sequence Diagram</i> Mengubah Data Mata Pelajaran.....	43
Gambar 5.3 <i>Sequence Diagram</i> Melihat Data Pengguna.....	43
Gambar 5.4 <i>Class Diagram</i>	45
Gambar 5.5 <i>Class Diagram Model</i>	46
Gambar 5.6 <i>Class Diagram Controller</i>	47
Gambar 5.7 Perancangan <i>Entity Relational Diagram</i>	48
Gambar 5.8 Perancangan Antarmuka Halaman Nilai Hafalan.....	51
Gambar 5.9 Perancangan Antarmuka Halaman Data Mata Pelajaran.....	52
Gambar 5.10 Perancangan Antarmuka Halaman Data Pengguna.....	54
Gambar 5.11 Perancangan <i>Physical Data Model</i>	56
Gambar 5.12 Implementasi Antarmuka Menambah Nilai Hafalan.....	60
Gambar 5.13 Implementasi Antarmuka Mengubah Data Mata Pelajaran.....	61
Gambar 5.14 Implementasi Antarmuka Melihat Data Pengguna.....	61
Gambar 6.1 <i>Flow Graph</i> pada NilaiHafalan.....	63
Gambar 6.2 <i>Flow Graph Function Edit dan Update</i>	67
Gambar 6.3 <i>Flow Graph Method setupListOperation</i>	69
Gambar 6.4 <i>Flow Graph Method update()</i> pada <i>DraftController</i>	71



DAFTAR LAMPIRAN

A.1 Wawancara.....	97
B.1 Rapor Nilai Hafalan.....	100
B.2 Rapor Nilai Kepemimpinan.....	101
B.3 Rapor Nilai Mata Pelajaran.....	102



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Lembaga Pendidikan Muslimah Indonesia (LPMI) Al-Izzah berdiri pada tanggal 17 Juli 2006 dan di Kota Batu Jawa Timur. LPMI Al-Izzah didirikan dengan bertujuan untuk menghapus kegelisahan orang tua tentang masa depan putra-putri mereka, terlebih untuk anak-anak putri. LPMI merupakan lembaga yang menggabungkan nilai-nilai pondasi Islam dengan memberikan pengetahuan yang menyeluruh mengenai hal-hal yang akan menjadi tantangan mereka pada zamannya. Berkembangnya krisis kepemimpinan di Indonesia, Al-Izzah International Boarding School mendirikan Al-Izzah Leadership School (ALS) khusus untuk santri-santri putra. ALS memiliki visi untuk menjadi lembaga pendidikan yang dapat menghasilkan kader-kader pemimpin yang berakhlak mulia, inteligen, dan berkredibilitas yang siap melaksanakan amanah Allah sebagai hamba dan khalifah-Nya. Tujuan dari berdirinya ALS untuk mempersiapkan generasi berbudi mulia yang siap untuk memimpin dan juga dipimpin untuk mempersatukan dan menciptakan kejayaan bagi umat Islam, mempersiapkan putra-putri yang akan meraih prestasi di dalam *ulumudin*, pengetahuan alam dan teknologi, serta mempersiapkan putra-putri yang pantang menyerah dan juga memiliki jiwa juang yang tinggi untuk mencapai cita-cita mereka masing-masing. ALS juga menawarkan program tahfidz Qur'an yang akan menjadi bekal mereka dalam melanjutkan studi di Timur Tengah dan juga di akhirat kelak (Al-Izzah, 2015).

Pengelolaan data siswa di sekolah seringkali tidak dianggap penting oleh para pelaksana pendidikan di Negara berkembang, dan masalah itu perlu ditangani jika ingin mencapai perencanaan yang efektif dan tujuan yang berkualitas ketika merealisasikan ke dalam dunia pendidikan (Ndiku, Oyco and Qwano, 2014). Ustaz sebagai guru pada pondok pesantren lebih memperhatikan proses dan perkembangan pendidikan santri daripada hasil yang mereka dapatkan yaitu rapor, padahal nilai atau rapor menjadi sebuah aspek yang sangat penting dalam proses pendidikan santri. Masalah kedisiplinan santri bagi para ustaz menjadi perhatian penting untuk keberhasilan mereka dalam mengajar dan untuk kemajuan sekolah dalam pembinaan santri. Sekolah yang mengutamakan kedisiplinan seperti pesantren akan selalu menciptakan proses belajar yang menyenangkan. Sebaliknya dalam sekolah yang tidak menerapkan kedisiplinan, kondisinya akan jauh berbeda. Pelanggaran yang akan terjadi akan dianggap hal biasa pada sekolah tersebut dan untuk memperbaiki kondisi tersebut tidaklah mudah (Kamaruddin and Patak, 2018).

Peran wali santri dalam mengetahui perkembangan anak dalam segi akademis, ekstrakurikuler, pelanggaran yang dilakukan santri, dan hafalan Qur'an mereka sangatlah penting. Dalam rangka memperoleh nilai atau prestasi yang baik dalam pendidikan mereka di sekolah, pemantauan orang tua kepada anak harus dilakukan secara terus menerus (Ceka and Murati, 2016). Perhatian orang tua



pada pendidikan anak diantara kesibukan pekerjaan mereka sangat penting dalam perkembangan pendidikan anak. Ketidakhadiran orang tua dalam pendidikan anak berdampak besar pada perkembangan nilai mereka. Putra putri yang mendapat perhatian lebih sedikit dari orang tua mereka, mempunyai presentasi lebih besar untuk mendapatkan nilai yang rendah (nilai yang lebih rendah dari 50 atau 75) dibandingkan dengan anak dengan perhatian besar dari orang tua mereka (Fu et al., 2017).

Oleh karena itu, pada saat observasi di pondok pesantren Al-Izzah dan ditemukan beberapa permasalahan dalam sistem penilaian pada pondok pesantren tersebut. Permasalahan yang ditemukan adalah kesulitan dalam pengamatan dan pendataan perkembangan nilai para santri, penilaian akhlak santri dan pelanggaran yang mereka lakukan, serta hafalan yang sudah mereka setorkan kepada pembimbing mereka. Pada saat ini wali santri diharuskan untuk datang ke pondok dan bertanya secara pribadi kepada pembimbing asrama dan ustaz untuk mengetahui secara lengkap perkembangan anak mereka. Adapun rapor nilai yang didapatkan pada bulan pertama semester, bulan ketiga semester, dan akhir semester sangat banyak. Setiap pembagian rapor nilai, setiap santri mendapatkan 4 sampai 5 lembar rapor untuk setiap penilaian yang ada.

Melihat persoalan yang telah dijelaskan, maka pengembangan teknologi dalam bidang sistem informasi yang dimaksud adalah "Pengembangan Sistem Penilaian Santri Terpadu di Al-Izzah Leadership School Berbasis Web". Tujuan dari pengembangan sistem ini adalah untuk memberi kemudahan kepada wali santri untuk mengamati perkembangan santri dalam kegiatan akademik, non akademik dan juga kemajuan hafalan mereka. Dengan itu para wali santri tetap akan mengetahui perkembangan pendidikan anak mereka di tengah kesibukan mereka. Maka dalam sistem yang akan dikembangkan ini, pengembangan perangkat lunak untuk pondok ini menggunakan metode *waterfall*, karena dalam proses elisitasi data di pondok tersebut semua kebutuhan yang penelitian ini butuhkan sudah diperoleh pada saat melakukan elisitasi kebutuhan. Pada penelitian ini terdapat fitur untuk menampilkan nilai para santri setiap awal semester, mid semester, dan akhir semester. Kemudian fitur untuk menampilkan sejauh mana hafalan yang sudah di setorkan ke pembimbing hafalan dan juga nilai akhlak dan kepemimpinan dari setiap santri. Sistem ini nantinya akan dikembangkan dengan platform *website* yang bertujuan agar wali santri, ustaz, dan *murobbi* mendapatkan kemudahan dalam mengakses sistem dimanapun dan kapanpun mereka berada. Sistem penilaian santri terpadu ini diharapkan dapat memberi kemudahan kepada santri, orang tua maupun ustaz dalam hal pengoperasian. Sistem ini diciptakan agar mempermudah semua elemen di dalam pondok dalam mengamati kualitas belajar para santri dan juga aktifitas santri di dalam sekolah.

1.2 Rumusan Masalah

Rumusan masalah merupakan permasalahan yang bersumber pada segala sesuatu yang dibahas di latar belakang, rumusan masalah yang didapat yaitu:



1. Bagaimanakah hasil analisis kebutuhan dalam sistem penilaian santri terpadu di Al-Izzah *Leadership School*?
2. Bagaimanakah hasil dalam tahap perancangan pada sistem penilaian santri terpadu di Al-Izzah *Leadership School*?
3. Bagaimanakah hasil dalam tahap implementasi pada sistem penilaian santri terpadu di Al-Izzah *Leadership School*?
4. Bagaimanakah hasil yang didapat dari pengujian sistem penilaian santri terpadu di Al-Izzah *Leadership School* mampu memastikan sistem berjalan dengan baik?

1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Melakukan analisis kebutuhan terhadap sistem penilaian santri terpadu di Al-Izzah *Leadership School* berbasis web.
2. Merancang sistem penilaian santri terpadu di Al-Izzah *Leadership School* berbasis web.
3. Mengimplementasikan sistem penilaian santri terpadu di Al-Izzah *Leadership School* berbasis web.
4. Melakukan pengujian terhadap sistem penilaian santri terpadu di Al-Izzah *Leadership School* berbasis web.

1.4 Manfaat

Penelitian ini memperoleh banyak manfaat yaitu dapat memberikan kemudahan bagi pemangku kepentingan dengan adanya pengembangan sistem penilaian santri terpadu berbasis web tersebut, untuk para santri dan wali santri lebih mudah dalam pengontrolan nilai anak mereka secara berkala, dan juga memudahkan ustaz dalam dokumentasi untuk penilaian santri.

1.5 Batasan Masalah

Berlandaskan rumusan masalah diatas, permasalahan diberikan batasan sebagai berikut.

1. Sistem ini dikembangkan dalam ruang lingkup Al-Izzah *Leadership School*.
2. Metode yang penelitian ini gunakan adalah metode *Waterfall*.
3. PHP dan MySQL merupakan bahasa pemrograman yang digunakan dalam penelitian ini.
4. *Web* adalah *platform* yang dipakai, maka yang diperlukan oleh pengguna adalah koneksi internet dan *browser*.

1.6 Sistematika Pembahasan

Sistematika penulisan menjelaskan uraian dan detail dari pembuatan susunan dari skripsi dengan gambaran umum, antara lain:



BAB I PENDAHULUAN

Bab ini memberikan penjelasan tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan pada penelitian.

BAB II LANDASAN KEPUSTAKAAN

Bab ini mencakup pelaksanaan implementasi dengan penjelasan teori dasar dan referensi pendukung yang digunakan.

BAB III METODOLOGI PENELITIAN

Bab ini membeberikan penjelasan mengenai tahapan dari metode penelitian yang dipakai dalam penelitian ini beserta penjelasan dari setiap tahapannya.

BAB IV ANALISIS DAN PERANCANGAN

Bab ini memberikan penjelasan mengenai tahapan dalam melakukan analisis kebutuhan dan juga perancangan yang dilakukan berdasar pada analisis kebutuhan yang ditemukan.

BAB V IMPLEMENTASI

Bab ini memberikan penjelasan mengenai proses imlementasi dari hasil perancangan yang dilaksanakan pada tahap perancangan.

BAB VI PENGUJIAN

Bab ini memberikan penjelasan mengenai pengujian yang dilakukan untuk memeriksa kecakapan sistem yang telah diimplementasikan.

BAB VII PENUTUP

Bab ini memaparkan kesimpulan yang diperoleh setelah melakukan penelitian ini serta saran yang ditujukan kepada peneliti selanjutnya mengenai pengembangan yang dapat dilakukan akan kekurangan yang ditemukan.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian terkait adalah penelitian dengan judul *“Education Management Information Systems in the Primary Schools of Sindh a case study of Hyderabad Division”*. Hasil dari penelitian ini mengindikasikan dampak yang sangat positif dalam penggunaan teknologi komunikasi dan informasi di lingkungan pendidikan. Perkembangan teknologi informasi di abad ke-21 memainkan peran penting dalam pendidikan Pakistan dan Teknologi Informasi Komunikasi (TIK) memiliki peran yang lebih efektif pada lembaga pendidikan seperti sekolah, perguruan tinggi dan universitas. Manajemen tertinggi dari berbagai lembaga telah menggunakan manajemen sistem informasi untuk meningkatkan efisiensi pekerjaan administratif mereka dengan bantuan sistem manajemen informasi. Tujuan dari penelitian ini adalah untuk mengeksplorasi pandangan dari kepala sekolah tentang SIM dan cara-cara di mana sekolah menggunakan sistem manajemen informasi. Responden dalam penelitian ini adalah 98 kepala sekolah dasar dan perguruan tinggi dari divisi Hyderabad. Pengumpulan data menggunakan kuesioner dengan lima bagian. Bagian pertama mengumpulkan informasi demografis responden. Bagian kuesioner yang tersisa memiliki pernyataan tentang sistem informasi manajemen sekolah. Data dianalisis untuk mendapatkan frekuensi, persentase, rata-rata dan standar deviasi. Hasilnya menunjukkan bahwa meskipun kurangnya infrastruktur teknis di sekolah dasar, otoritas yang lebih tinggi memberikan kontribusi penting untuk mendukung manajemen sekolah dalam penerapan sistem manajemen informasi (Nagar et al., 2019). Pesantren Al-Izzah *Leadership School* mempunyai kebutuhan spesifik yaitu manajemen penilaian akhlak dari para santri dan hafalan Qur'an mereka. Sistem ini tidak hanya mengembangkan kemampuan para guru atau ustaz, sistem ini akan mengembangkan pengetahuan orang tua akan teknologi agar mereka dapat mengetahui perkembangan pendidikan anak mereka kapanpun dan dimanapun.

Penelitian kedua berjudul *“Developing a Web-based Assessment System for Evaluating Examinee’s Understanding of the Procedure of Scientific Experiments”* yang dilakukan oleh Tzu-Hua Wang. Sistem penilaian daring atau yang biasa disebut ujian daring, telah banyak dipakai di lingkungan pendidikan, dan hal tersebut menjadi kebutuhan dalam pengajaran dan penelitian. Kelebihan utama dari ujian daring terletak pada pengurangan penggunaan kertas dan mempercepat pengumpulan nilai serta analisis data tersebut. Keutamaan tersebut menunjukkan betapa bergunanya hal tersebut untuk para guru dan juga para peneliti (Wang, 2018). Dengan adanya sistem penilaian terpadu ini akan mengurangi penggunaan kertas ketika adanya pembagian nilai rapor, dimana setiap pembagian rapor membutuhkan 5 lembar setiap santrinya.

Penelitian ketiga berjudul *“School Management Information Systems : Challenges To Educational Decisionmaking In The Big Data Era”* yang dilakukan



oleh Vivienne V. Forrester. Penelitian tersebut mengemukakan bahwa perkembangan dalam teknologi informasi dan faktor-faktor yang terhubung dengannya seperti pertukaran informasi membuat harapan dari masyarakat meningkat, pengelolaan sistem informasi pada sekolah telah membuat dampak yang signifikan terhadap instansi pendidikan dalam praktik pendidikan mereka. Penggunaan data dalam menginformasikan keputusan dalam bidang pendidikan telah mengundang perhatian, terutama pada pelaporan data siswa di dalam sebuah Negara atau daerah (V. Forrester, 2019). Banyak pendapat bahwa sekolah menggunakan banyak basis data yang terpisah, satu basis data digunakan untuk absensi kehadiran, basis data lain digunakan untuk penyimpanan nilai tes ujian, basis data lain untuk menyimpan nilai siswa, dan basis data tersebut tidak semua bisa dihubungkan antara satu basis data dengan basis data yang lain (Hinze-Pifer and Ramsey, 2011). Untuk sistem penilaian terpadu, basis data dari seluruh aspek dalam pondok Al-Izzah *Leadership School* terintegrasi antara satu dengan yang lain, sehingga ketika pengumpulan data siswa bisa diketahui semua perkembangan mereka pada sistem tersebut.

2.2 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan penerapan teknik yang berhubungan dengan bagian-bagian dalam proses pembuatan *software*, dimulai dengan melakukan spesifikasi sistem sampai pemeliharannya ketika sistem tersebut telah digunakan oleh pengguna. Rekayasa sistem berhubungan dengan semua bagian dalam pembangunan dan pertumbuhan sistem yang elusif. Rekayasa sistem berhubungan dengan pembangunan perangkat keras, kebijaksanaan dan proses dalam membuat desain, penerbitan sistem dan rekayasa perangkat lunak. Insinyur sistem mempunyai peran dalam menentukan sistem yang akan dibuat, arsitektur dalam sistem tersebut, kemudian mengintegrasikan seluruh bagian-bagian tersebut untuk membuat sistem yang sempurna (Sommerville, 2016).

2.3 Pengembangan Perangkat Lunak

2.3.1 *Software Development Life Cycle (SDLC)*

Siklus ini merupakan gambaran yang dipermudah dari proses pengembangan perangkat lunak. Setiap model proses mewakili suatu proses dari sudut pandang yang spesifik dan mempersiapkan informasi yang terpisah pada proses tersebut. Contoh, pada suatu proses menunjukkan urutan aksi dari sebuah sistem tetapi tidak menunjukkan peran orang yang terlibat dalam aksi tersebut (Sommerville, 2016). Dalam sistem ini SDLC menentukan proses dari pengembangan sistem yang akan dibangun.

2.3.2 Tahapan Metode *Waterfall*

Waterfall model pada Gambar 2.1 adalah contoh dari model yang terencana. Pada prinsipnya perencanaan dan penjadwalan semua aktifitas sebelum memulai



pengembangan perangkat lunak. Tahapan dalam model *waterfall* secara langsung mencerminkan dasar dari pengembangan perangkat lunak (Sommerville, 2016):

1. *Requirements analysis and definition*. Layanan, hambatan, dan maksud dari sistem ditentukan melalui diskusi dengan *user* lalu dideskripsikan secara menyeluruh dan digunakan untuk menjadi spesifikasi sistem.
2. Desain sistem dan perangkat lunak. Dalam mendesain sistem diperlukan kebutuhan dari *software* dan *hardware*. Hal ini membangun arsitektur sistem secara menyeluruh. Perangkat lunak di desain dengan mengenali dan menampilkan konsep dasar dari sistem perangkat lunak dan kaitan dari konsep tersebut.
3. Pengujian unit dan implementasi. Tahap ini mewujudkan desain dari perangkat lunak sebagai satu unit program. Pengujian dari unit program tersebut membutuhkan verifikasi setiap unit bahwa unit tersebut memenuhi spesifikasi yang dibutuhkan.
4. Pengujian integrasi dan sistem. Masing masing unit program digabungkan dan dilakukan percobaan sebagai sistem yang lengkap. Hal tersebut dilakukan dengan tujuan untuk menentukan kelengkapan dari sebuah kebutuhan. Setelah melakukan pengujian, pengguna mencoba sistem perangkat lunak yang sudah diuji.
5. Perbaikan dan operasi. Fase siklus ini biasanya adalah siklus hidup terpanjang. Sistem digunakan oleh pengguna. Pada tahap pemeliharaan melibatkan perbaikan dari kekeliruan yang sebelumnya tidak dijumpai pada awal siklus. Pengembangan dan peningkatan sistem dilakukan ketika menemukan kebutuhan dari perangkat lunak terkini.



Gambar 2.1 Waterfall Model (Sommerville, 2016)



2.3.3 Pemodelan Berorientasi Objek

Pemodelan berorientasi objek merupakan pemodelan yang terdiri dari objek-objek yang berinteraksi, objek tersebut mempertahankan keadaan mereka dan menyediakan operasi pada keadaan tersebut. Representasi dari keadaan tersebut bersifat pribadi dan tidak dapat diakses langsung dari luar objek. Desain berorientasi objek melibatkan perancangan kelas-kelas dari objek dan hubungan antara kelas-kelas tersebut. Kelas-kelas ini mendefinisikan objek dalam sistem dan interaksinya. Ketika desain diimplementasikan sebagai program, objek tersebut dibuat secara dinami dari definisi kelas yang telah dirancang (Sommerville, 2016).

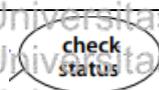
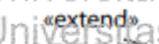
2.3.4 Unified Modelling Language (UML)

Unified Modeling Language (UML) merupakan bahasa ilustratif guna melukiskan, memastikan, menciptakan, serta menyimpan artefak dari perangkat lunak. UML memberikan cara termudah ketika menulis bagan dari sistem, meliputi hal-hal yang konseptual seperti proses bisnis dan fungsi dari sistem (Booch, Rumbaugh and Jacobson, 2005). Penelitian ini mencakup 3 diagram UML.

2.3.4.1 Use Case Diagram

Use case diagram memperlihatkan hubungan beberapa *use case* dengan aktor di dalam sistem. Diagram ini mengatasi bentuk *use case* statis dari sistem. Diagram ini sangat penting dalam mengorganisir dan membuat model perilaku dari sistem. Notasi dan penjelasannya dari diagram ini dijelaskan pada Tabel 2.1 dan contoh dari diagram ini terdapat pada Gambar 2.2.

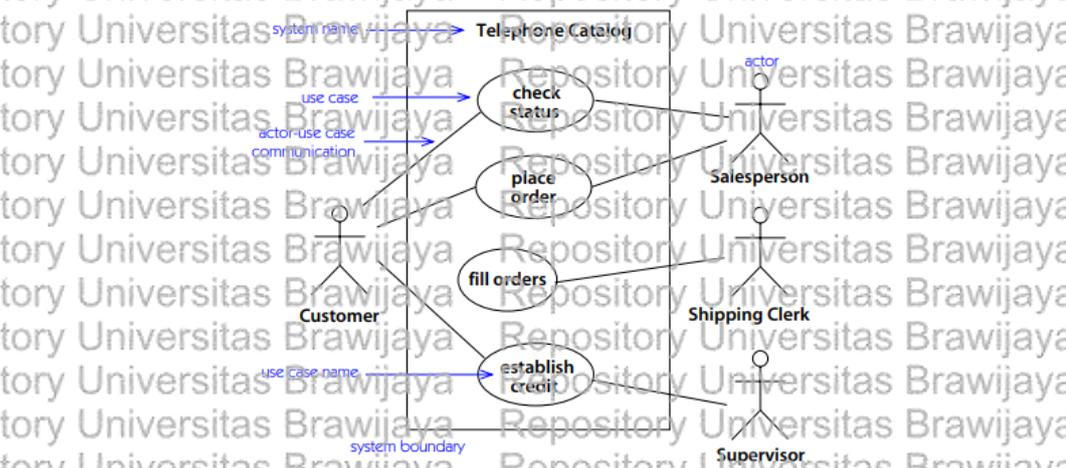
Tabel 2.1 Notasi Use Case Diagram

Notasi	Deskripsi
	Notasi Aktor adalah penggambaran dari pengguna sistem yang berinteraksi di dalam sistem.
	Notasi <i>use case</i> adalah fungsionalitas yang dapat dijalankan oleh aktor
	Notasi asosiasi merupakan jalur antara aktor dengan <i>use case</i> yang bersangkutan untuk melakukan komunikasi
	Notasi <i>extend</i> adalah relasi untuk menambah perilaku <i>base use case</i> yang tidak mengetahui penambahan tersebut
	Notasi <i>use case generalization</i> adalah relasi antara <i>general use case</i> dengan <i>use case</i> yang lebih spesifik untuk mewariskan dan menambahkan fitur



Tabel 2.1 Notasi Use Case Diagram (Lanjutan)

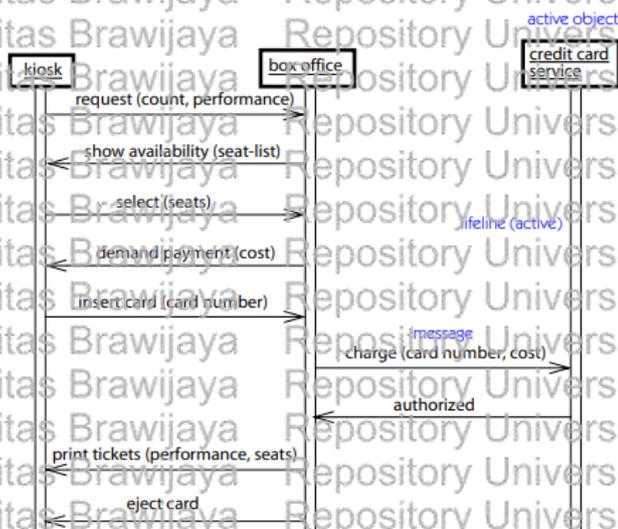
Notasi *include* adalah relasi untuk menambah perilaku *base use case* dan secara eksplisit mendeskripsikan penambahan tersebut



Gambar 2.2 Contoh Use Case Diagram (Booch, Rumbaugh and Jacobson, 2005)

2.3.4.2 Sequence Diagram

Sequence diagram merupakan diagram yang memberikan gambaran dari pengurutan waktu untuk setiap pesan. *Sequence diagram* menunjukkan kumpulan dari peran, pengiriman pesan dan penerimaan pesan dari dan ke suatu objek. *Sequence diagram* digunakan untuk mengilustrasikan tampilan dinamis dari sistem. Pada Gambar 2.3 terdapat contoh *sequence diagram*.

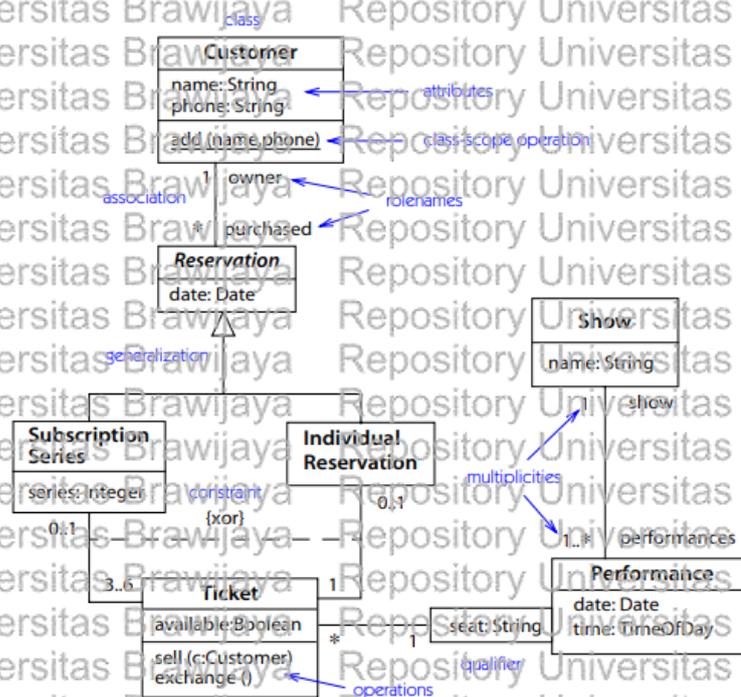


Gambar 2.3 Contoh Sequence Diagram (Booch, Rumbaugh and Jacobson, 2005)



2.3.4.3 Class Diagram

Class diagram menampilkan kumpulan kelas, antarmuka, kumpulan, dan ikatan dari setiap *class*. *Class diagram* merupakan *class* pemodelan yang paling sering dijumpai pada sistem berorientasi objek. *Class diagram* membahas tentang bentuk desain yang statis pada suatu sistem. *Class diagram* yang mempunyai kelas-kelas yang aktif mengarahkan kepada proses yang statis pada suatu sistem. Contoh *class diagram* terdapat pada Gambar 2.4.



Gambar 2.4 Contoh *Class Diagram* (Booch, Rumbaugh and Jacobson, 2005)

2.4 Pengujian

Pengujian merupakan tahap untuk menunjukkan bahwa program yang dibuat sesuai dengan tujuannya, dan untuk menemukan kekurangan pada program yang telah dibuat sebelum digunakan. Pada tahap pengujian perangkat lunak, proses dalam menjalankan program dengan data buatan. Kemudian, memeriksa hasil dari uji coba program tersebut untuk menemukan kesalahan, ketidaknormalan dan informasi dari atribut program yang tidak berfungsi dengan baik (Sommerville, 2016).

2.4.1 Unit Testing

Unit testing dilakukan pada level terendah dari pengujian. *Unit testing* menguji unit dasar dari perangkat lunak, unit tersebut adalah bagian terkecil dari perangkat lunak. *Unit testing* juga disebut pengujian modul atau komponen. Hal ini merujuk kepada tes yang memverifikasi fungsionalitas dari bagian kode



tertentu, biasanya terdapat pada *class*, dan *unit testing* termasuk dengan pembangun dan penghancur perangkat lunak (Ghuman, 2014).

2.4.2 Integration Testing

Pengujian ini dilakukan ketika ada dua atau lebih unit yang diuji digabungkan ke dalam struktur yang lebih besar. Pengujian ini sering dilakukan pada kedua antarmuka antara komponen dan struktur yang lebih besar ketika sedang dibangun, sedangkan kualitasnya tidak dapat diukur dari komponennya. *Integration testing* adalah jenis pengujian perangkat lunak yang berupaya untuk memverifikasi antarmuka antara komponen dan desain perangkat lunak. Komponen perangkat lunak dapat diintegrasikan secara iteratif satu per satu atau semua komponen secara bersamaan. *Integration testing* berfungsi untuk mengekspos kerusakan pada antarmuka dan interaksi antara komponen yang terintegrasi. Beberapa komponen perangkat lunak yang telah diuji menyesuaikan dengan elemen dari desain arsitektur, diintegrasikan dan diuji hingga perangkat lunak bekerja sebagai suatu sistem (Ghuman, 2014).

2.4.3 Validation Testing

Validation testing adalah proses yang paling umum. Tujuan dari *validation testing* pada perangkat lunak adalah untuk menandakan bahwa perangkat lunak yang dibuat melengkapi harapan dari pengguna. *Validation testing* lebih dari pemeriksaan kesesuaian dengan spesifikasi untuk menunjukkan bahwa perangkat lunak melakukan apa yang diharapkan pengguna untuk dilakukan. Validasi sangat penting karena pernyataan dari kebutuhan pengguna tidak selalu mencerminkan keinginan atau kebutuhan nyata dari pelanggan dan pengguna sistem (Sommerville, 2016).

2.4.4 Usability Testing

Usability testing adalah metode dimana pengguna dari suatu perangkat lunak diminta untuk melakukan tugas-tugas tertentu dalam upaya untuk mengukur kemudahan penggunaan perangkat lunak, waktu, dan persepsi pengguna tentang pengalaman dalam menggunakan perangkat lunak. *Usability testing* dapat dilakukan secara resmi, di laboratorium dengan kamera untuk pengamatan pengguna, atau secara informal dengan kertas mockup aplikasi atau situs web. Banyak perubahan dilakukan pada suatu perangkat lunak berdasarkan penemuan dari *usability testing*. Partisipasi dari *usability testing* didorong untuk berpikir keras dalam menyuatkan setiap penilaian mereka (M et al., 2014).

2.5 Pondok Pesantren

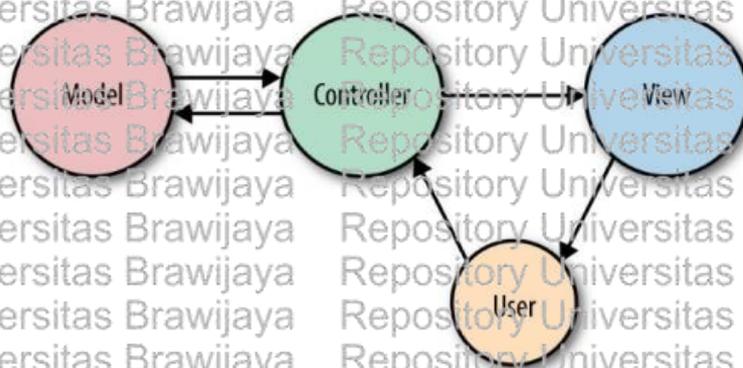
Pesantren adalah salah satu lembaga islam yang menjadi subkultur rakyat indonesia. Pesantren merupakan institusi yang unik, mempunyai ciri khas yang kuat dan melekat. Sejak dulu, pesantren berupaya untuk mencerdaskan bangsa tanpa henti. Pesantren sudah memberikan pendidikan sejak masa perjuangan



melawan penjajahan dan menjadi tempat belajar yang masih bertahan hingga saat ini. Pendidikan dalam pesantren tidak sama sekali bertujuan untuk mengejar kepentingan duniawi seperti uang dan kekuasaan, tetapi pesantren menanamkan kepada benak para santri bahwa belajar adalah kewajiban dan untuk mengabdikan kepada Tuhan. Oleh karena itu, pesantren memiliki tanggung jawab yang besar dalam membentuk karakter dari setiap santri (Zuhriy, 2011). Pondok tempat penelitian ini dilaksanakan Al-izzah *Leadership School* adalah salah satu pondok yang mengaplikasikan pendidikan karakter untuk membentuk para santri menjadi pemimpin di masa depan, selain itu mereka juga mengadakan program tahfidz Qur'an dimana pondok ini tidak hanya membimbing mereka mencapai kehidupan duniawi melainkan juga mencapai kehidupan akhirat. Pondok pesantren ini tak hanya mengajarkan pelajaran islam saja, mata pelajaran ilmu pengetahuan alam dan ilmu pengetahuan sosial sebagaimana pada kurikulum di sekolah menengah pertama dan sekolah menengah atas juga diajarkan kepada para santri guna membekali para santri dikala mereka ingin mendaftarkan diri ke perguruan tinggi, maka dari itu penelitian ini memfokuskan kepada proses pengolahan data nilai-nilai santri dalam hal akademik, nilai kepemimpinan dan juga nilai dan perkembangan hafalan Qur'an para santri. Pelanggaran santri juga kami kelola dalam sistem kami karena pelanggaran adalah satu hal yang sulit diketahui oleh para wali santri jika santri tersebut tidak memberitahukan kepada orang tuanya, maka kami mengaplikasikan notifikasi pelanggaran santri kepada setiap wali santri agar para wali santri dapat mengetahui dan menegur anaknya jika melakukan pelanggaran.

2.6 Laravel

Laravel adalah salah satu *framework* untuk pengembangan aplikasi web, laravel memiliki sintaks yang dapat digambarkan secara tepat dan juga secara elegan. Laravel juga memberikan solusi yang baik dalam pengembangan dengan memfasilitasi tugas-tugas yang umum dalam pengerjaan proyek dengan skala yang besar. Laravel mempunyai fungsi-fungsi yang sangat memudahkan dalam pengerjaan seperti *Blade Templating*, *Eloquent ORM*, *Application Logic*, *Reverse Routing*, *Class Auto Loading* dll. Implementasi *framework* laravel pada penelitian ini mencakup konsep *model view controller (MVC)*, *Eloquent Object Relation Mapping (ORM)*, *Blade Templating*, dan juga pemasangan *package* yang membantu fungsi dari system (Parker et al., 2016). Konsep MVC dapat dilihat pada Gambar 2.5.



Gambar 2.5 Ilustrasi MVC (Stauffer, 2017)

Controller pada dasarnya adalah kelas-kelas yang mengatur logika satu atau lebih rute bersama di satu tempat. *Controller* cenderung mengelompokkan rute yang sama pada satu tempat, terutama jika aplikasi Anda terstruktur dengan format CRUD. Tindakan paling sederhana yang dapat dilakukan dalam aplikasi Laravel adalah menentukan rute dan mengembalikan hasil kapan saja seseorang mengunjungi rute itu (Stauffer, 2017). Menambahkan *controller* pada rute sederhana dapat memudahkan ketika pemanggilan rute dengan *method* tertentu pada suatu kelas *controller*. Contoh dari rute sederhana dengan *controller* dapat dilihat pada Gambar 2.6.

```

// File: routes/web.php
<?php

Route::get('/', 'WelcomeController@index');
// File: app/Http/Controllers/welcomeController.php
<?php
namespace app\Http\Controllers;

class WelcomeController
{
    public function index()
    {
        return 'Hello, World!';
    }
}
  
```

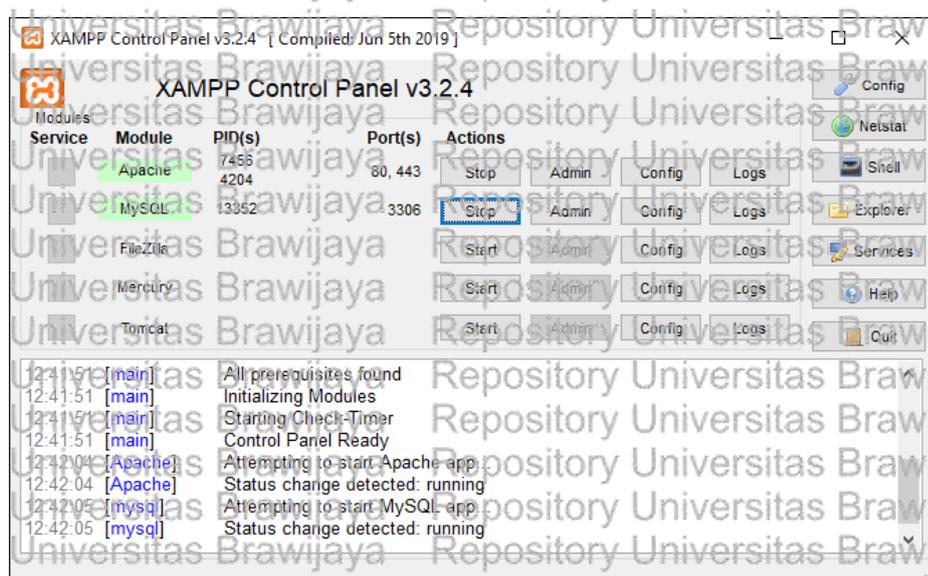
Gambar 2.6 Laravel Route dengan Controller (Stauffer, 2017)

2.7 XAMPP

Menginstall server web apache tidak mudah dilakukan dan semakin susah jika hendak memasukkan MariaDB, PHP dan Perl. Tujuan XAMPP adalah untuk membangun distribusi instalasi yang mempermudah para pengembang untuk masuk ke dunia Apache. XAMPP telah dikonfigurasi dengan semua fitur yang dapat digunakan, dengan tujuan agar para pengembang lebih nyaman dalam penggunaan XAMPP. Untuk penggunaan secara komersial XAMPP dapat diakses secara gratis, untuk saat ini sudah ada distribusi untuk Windows, Linux, dan OS X (About apachefriends.org). XAMPP digunakan untuk mengakses web server untuk menampilkan sistem yang kami buat di sebuah browser. Untuk mengaktifkan



server apache dan MySQL kita dapat mengaktifkannya pada aplikasi XAMPP Control Panel yang dapat dilihat pada Gambar 2.7.



Gambar 2.7 XAMPP Control Panel



BAB 3 METODOLOGI

Gambar 3.1 menjelaskan mengenai serangkaian langkah-langkah pengembangan perangkat lunak yang diadaptasi dari metode waterfall. Tahap-tahap ini dilakukan untuk menyelesaikan permasalahan Pengembangan Sistem Penilaian Terpadu di Al-Izzah Leadership School Berbasis Web. Serangkaian langkah tersebut yaitu:



Gambar 3.1 Alur Penelitian

3.1 Studi Literatur

Tahap ini menjelaskan dasar penelitian untuk membantu kelancaran perancangan dari sistem yang hendak dibangun. Dasar penelitian ini diperoleh dari penelitian yang sudah ada sebelumnya seperti jurnal, buku dan literatur yang diperoleh dari internet yang mana sebagai penunjang dalam penulisan penelitian ini. Dasar penelitian tersebut telah dijelaskan dalam bab sebelumnya yaitu Landasan Kepustakaan.

3.2 Rekayasa Kebutuhan

Tahapan ini diperlukan guna memperoleh semua kebutuhan yang diperlukan pada tahap perancangan sistem. Dari tahap ini nantinya akan memperoleh suatu representasi yang nantinya dapat mempermudah ketika melakukan perancangan.



1. Elisitasi dan Analisis Kebutuhan

Pada saat elisitasi dan analisis kebutuhan pertama kali dilakukan didapatkan kebutuhan fungsional untuk mengolah nilai mata pelajaran, mengolah nilai kepemimpinan, mengolah nilai hafalan, dan mengolah data santri. Dari kebutuhan fungsional tersebut menghasilkan kebutuhan fungsional untuk mengolah data nilai mata pelajaran, mengolah data nilai kepemimpinan, dan mengolah data nilai hafalan. Untuk setiap fungsional diatas, dibutuhkan hak akses masing-masing dari setiap pengguna sistem dan menghasilkan kebutuhan fungsional untuk mengolah data pengguna. Kebutuhan non-fungsional didapatkan dari tingkat kenyamanan dan kemudahan dari pengguna ketika menggunakan sistem ini, dikarenakan pengguna dari sistem ini adalah para guru-guru dari pesantren tersebut dan juga wali santri yang telah menginjak umur 30 sampai 40 tahun ke atas, maka dari itu kami mengambil *usability* sebagai kebutuhan non-fungsional.

2. Spesifikasi Kebutuhan

Pada tahap ini hasil dari elisitasi dan analisis kebutuhan yang telah dilakukan sebelum ini dijabarkan secara rinci, jelas, tidak ambigu, mudah dipahami, dan konsisten. Kebutuhan fungsional menjelaskan hal-hal yang dapat dilakukan sistem yang akan dibangun sedangkan kebutuhan non-fungsional menjelaskan kebutuhan sistem dalam segi kualitas yang akan dibangun.

3. Manajemen Kebutuhan

Tahapan ini dilakukan untuk memudahkan dalam mengidentifikasi, mengontrol dan melacak kebutuhan. Maka pada tahapan ini dilakukan kodefikasi pada kebutuhan-kebutuhan yang ada. Kode SPST-F-1 untuk menandakan kebutuhan fungsional dan SPST-NF-1 untuk menandakan kebutuhan non-fungsional.

4. Pemodelan Kebutuhan

Tahap terakhir adalah pemodelan kebutuhan. Untuk menggambarkan fungsionalitas dan batasan sistem secara visual, penelitian ini menggunakan alat bantu *use case diagram*, sedangkan untuk memberikan penjelasan mengenai skenario interaksi antar aktor dengan sistem secara mendetail, penelitian ini menggunakan alat bantu *use case scenario*. Penggunaan kedua diagram tersebut dalam tahapan ini dapat memodelkan kebutuhan sistem baik secara visual maupun secara tekstual dengan detail.

3.3 Perancangan Aplikasi

1. Perancangan Arsitektur. Pemodelan pada perancangan sistem yang hendak dibangun nantinya mengacu pada pemodelan berorientasi objek yang akan dibuat dalam bentuk *use case diagram*, *sequence diagram* dan *class diagram*.
2. Perancangan Data. Pada tahap perancangan data nantinya berisikan penjelasan mengenai rancangan tabel basis data. Perancangan data ini sebagai acuan untuk menjadi dasar dari implementasi basis data di dalam sistem.



3. Perancangan Komponen. Pada tahap ini nantinya digambarkan beberapa contoh yang berisi algoritme utama pada sistem. Algoritme ini didapatkan dari kelas *controller*. Algoritme tersebut akan dituliskan dengan bentuk *pseudocode*.

4. Perancangan *interface*. Pada tahap perancangan *interface* pada sistem yang dirancang nantinya tahap awal akan membuat sebuah *mock up (low fidelity)* yang memvisualisasikan gambaran tata letak dan beberapa komponen fitur yang dibangun menurut kebutuhan sistem yang telah diperoleh pada tahap rekayasa kebutuhan.

3.4 Implementasi

Tahap implementasi menjelaskan tahapan dalam pembangunan sistem yang menggunakan hasil dari pada perancangan pada tahap sebelumnya dan implementasi hal-hal yang telah didapatkan dari proses yang telah dilalui. Fase-fase dalam melakukan implementasi adalah implementasi dengan memakai *framework Bootstrap* dengan memakai *framework laravel* sebagai bahasa pemrograman PHP dan penggunaan *database* dalam sistem ini memakai MySQL-*phpMyAdmin*.

3.5 Pengujian dan Analisis

Tahap pengujian dan analisis dilakukan guna menemukan kesalahan sistem setelah selesai dibangun. Pada penelitian ini dilakukan beberapa pengujian. Pengujian bertujuan untuk melakukan pengujian dari tiap unit yaitu *class*, komponen, ataupun *object* yang berasal dari perangkat lunak yang sebelumnya telah ditentukan pada tahap perancangan. Pengujian juga melakukan beberapa pengecekan mengenai fungsi-fungsi pada perangkat lunak yang ada apakah dalam proses pembangunan sistem sudah benar atau belum. Pengujian ini termasuk dengan pengujian "*black box testing*" yang mana cara menjalankannya atau mengeksekusinya dengan menggunakan model atau unit lain dilakukan pengamatan pada hasilnya, apakah hasil yang didapat sesuai apa yang diinginkan.

3.6 Kesimpulan dan Saran

Tahapan terakhir pada penelitian ini dilakukan jika semua tahapan telah selesai dilakukan. Hasil dari kesimpulan dan saran nantinya dapat digunakan sebagai acuan bagi peneliti selanjutnya untuk memperbaiki kesalahan dan dapat dilakukan pengembangan ke tahapan selanjutnya.



BAB 4 ANALISIS KEBUTUHAN

4.1 Gambaran Umum Sistem

Pengembangan sistem penilaian terpadu *online* ini akan membantu pesantren Al-Izzah dalam kebutuhan administrasi pesantren. Sistem ini memberikan fitur untuk memudahkan ustaz untuk mendaftarkan nilai-nilai santri setiap ujian selesai dilaksanakan dan juga memberikan kemudahan dalam pembuatan rapor untuk setiap santri setiap penerbitan rapor pada semester pertama, pertengahan semester, dan akhir semester. Sebelum adanya sistem ini, bagian administrasi pesantren diharuskan mendaftarkan nilai santri dari setiap aspek penilaian dan memasukkannya ke setiap rapor santri. Dengan adanya sistem ini memudahkan para ustaz untuk memudahkan penerbitan rapor dengan langkah dan waktu yang lebih efisien. Selanjutnya fitur yang ada pada penelitian ini notifikasi kepada wali santri tentang pelanggaran yang dilakukan oleh anak mereka. SPST mewakili singkatan dari sistem ini yaitu Sistem Penilaian Santri Terpadu, gambaran umum dari sistem ini dapat dilihat pada Gambar 4.1.



Gambar 4.1 Gambaran Umum Sistem

4.2 Identifikasi Aktor

Identifikasi aktor menunjukkan status aktor yang menjadi pengguna sistem. Penentuan aktor pada sistem diperoleh setelah melakukan observasi pada pesantren Al-Izzah. Kolom deskripsi menjelaskan hak akses yang dimiliki setiap aktor pada sistem dideskripsi dijelaskan secara lengkap pada Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Admin	Admin merupakan staff administrasi pesantren yang dapat mengakses seluruh fungsi yang ada pada sistem.
Ustaz	Ustaz merupakan guru pesantren yang mengajar pelajaran umum maupun pelajaran <i>diniyah</i> . Guru dapat mengakses nilai mata pelajaran, dan rapor mental. Tetapi, ustaz tidak dapat mengakses fitur mencetak dari ketiga fitur tersebut.



Tabel 4.2 Identifikasi Aktor (Lanjutan)

Aktor	Deskripsi
<i>Murobbi</i>	<i>Murobbi</i> adalah pembimbing santri yang memantau hafalan quran dan mendidik para santri di asrama. <i>Murobbi</i> dapat mengakses nilai hafalan, dan nilai kepemimpinan. <i>Murobbi</i> tidak dapat mengakses fitur mencetak dari kedua fitur tersebut.
Wali Santri	Wali santri adalah orang tua dari santri. Wali santri hanya dapat melihat nilai-nilai anak mereka dan juga dapat mencetaknya.

4.3 Daftar Kebutuhan Sistem

4.3.1 Teori Kode Kebutuhan

Teori kode kebutuhan digunakan untuk memberikan standarisasi aturan penomoran pada kebutuhan fungsional maupun kebutuhan non fungsional. Aturan penomoran ini berfungsi untuk mempermudah dalam memahami spesifikasi dari daftar kebutuhan. Sebagaimana dijelaskan pada Gambar 4.2, SPST mewakili singkatan dari sistem ini yaitu Sistem Penilaian Santri Terpadu, F/NF membedakan antara kebutuhan fungsional dan non fungsional, dan nomor di belakang untuk mengetahui urutan dari kebutuhan tersebut.

Kode : SPST-F/NF-1



Gambar 4.2 Aturan Penomoran

4.3.2 Daftar Kebutuhan Fungsional

Kebutuhan fungsional adalah fitur yang telah sistem siapkan beserta prosesnya, fitur tersebut dapat digunakan oleh pengguna. Kebutuhan fungsional diperoleh dari hasil elisitasi kebutuhan, dan ditemukan 34 kebutuhan dari hasil tersebut. Kebutuhan fungsional sistem dijelaskan pada Tabel 4.2.



Tabel 4.1 Kebutuhan Fungsional Sistem

No.	Use Case	Spesifikasi	Kode
1	Login	Sistem mampu mengidentifikasi pengguna sebagai admin, wali santri, ustaz, atau <i>murobbi</i> .	SPST_F_1
2	Logout	Sistem harus memiliki fungsi untuk membuat pengguna keluar dari sistem.	SPST_F_2
3	Melihat Data Santri	Sistem mampu menampilkan detail dari data santri yang terdaftar di pesantren Al- <i>izzah</i> .	SPST_F_3
4	Mengubah Data Santri	Sistem mampu melakukan perubahan pada data santri.	SPST_F_4
5	Menambah Data Santri	Sistem harus dapat menambah data santri	SPST_F_5
6	Menghapus Data Santri	Sistem mampu melakukan penghapusan pada data santri	SPST_F_6
7	Melihat Data Pengguna	Sistem mampu memperlihatkan detail dari data pengguna	SPST_F_7
8	Mengubah Data Pengguna	Sistem mampu mengubah data pengguna	SPST_F_8
9	Menambah Data Pengguna	Sistem mampu melakukan penambahan pada data pengguna	SPST_F_9
10	Menghapus Data Pengguna	Sistem mampu melakukan penghapusan pada data pengguna	SPST_F_10
11	Melihat Data Mata Pelajaran	Sistem harus mampu menampilkan data mata pelajaran	SPST_F_11
12	Mengubah Data Mata Pelajaran	Sistem harus mampu mengubah data mata pelajaran	SPST_F_12
13	Menambah Data Mata Pelajaran	Sistem harus mampu menambah data mata pelajaran	SPST_F_13
14	Menghapus Data Mata Pelajaran	Sistem harus mampu menghapus data mata pelajaran	SPST_F_14
15	Melihat Data Kepemimpinan	Sistem harus dapat menampilkan data kepemimpinan santri	SPST_F_15
16	Mengubah Data Kepemimpinan	Sistem harus dapat mengubah data kepemimpinan santri	SPST_F_16
17	Menambah Data Kepemimpinan	Sistem harus dapat menambah data kepemimpinan santri	SPST_F_17
18	Menghapus Data Kepemimpinan	Sistem harus dapat menghapus data kepemimpinan santri	SPST_F_18



Tabel 4.2 Kebutuhan Fungsional Sistem (Lanjutan)

No.	Use Case	Spesifikasi	Kode
19	Melihat Data Hafalan	Sistem mampu memperlihatkan data hafalan santri	SPST_F_19
20	Mengubah Data Hafalan	Sistem harus dapat mengubah data hafalan santri	SPST_F_20
21	Menambah Data Hafalan	Sistem harus dapat menambah data hafalan santri	SPST_F_21
22	Menghapus Data Hafalan	Sistem harus dapat menghapus data hafalan santri	SPST_F_22
23	Melihat Pelanggaran Santri	Sistem harus dapat menampilkan data pelanggaran santri	SPST_F_23
24	Menambah Pelanggaran Santri	Sistem harus dapat menambah data pelanggaran santri	SPST_F_24
25	Melihat Nilai Mata Pelajaran	Sistem harus dapat menampilkan nilai mata pelajaran santri	SPST_F_25
26	Menambah Nilai Mata Pelajaran	Sistem harus dapat menambah nilai mata pelajaran santri	SPST_F_26
27	Mengubah Nilai Mata Pelajaran	Sistem harus dapat mengubah nilai mata pelajaran santri	SPST_F_27
28	Melihat Nilai Kepemimpinan	Sistem harus dapat menampilkan nilai kepemimpinan santri	SPST_F_28
29	Mengubah Nilai Kepemimpinan	Sistem harus dapat mengubah nilai kepemimpinan santri	SPST_F_29
30	Menambah Nilai Kepemimpinan	Sistem harus dapat menambah nilai kepemimpinan santri	SPST_F_30
31	Melihat Nilai Hafalan	Sistem harus dapat menampilkan nilai hafalan santri	SPST_F_31
32	Mengubah Nilai Hafalan	Sistem harus dapat mengubah nilai hafalan santri	SPST_F_32
33	Menambah Nilai Hafalan	Sistem harus dapat menambah nilai hafalan santri	SPST_F_33
34	Mencetak Rapor	Sistem harus dapat mencetak rapor dari seluruh nilai santri	SPST_F_34



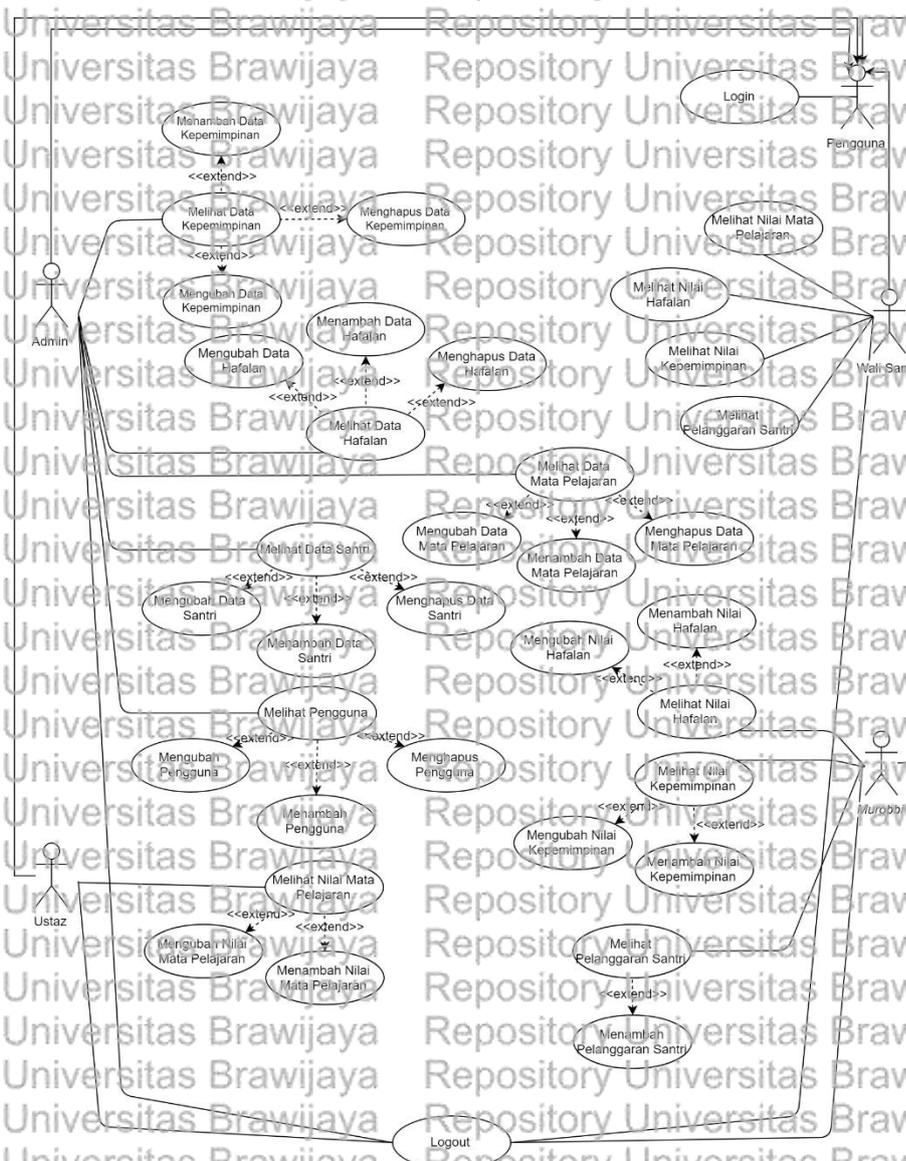
4.3.3 Daftar Kebutuhan Non Fungsional

Tabel 4.2 Kebutuhan Non Fungsional Sistem

No.	Parameter	Deskripsi	Kode
1	Usability	Sistem perlu mempunyai <i>interface</i> yang mudah dipakai oleh Admin, Ustadz, Murobbi, dan Wali Santri	SPST_NF_1

4.4 Pemodelan Kebutuhan

4.4.1 Pemodelan Use Case Diagram



Gambar 4.3 Use Case Diagram Sistem Penilaian Santri Terpadu



4.5 Use Case Scenario

Use case scenario menjelaskan alur tiap *use case diagram* yang dijelaskan pada Gambar 4.3. *Use case scenario* berisikan *actor*, objektif, kondisi awal, aliran utama, aliran alternatif, kondisi akhir, yang ditulis didalam tabel. Jumlah *use case scenario* yang dituliskan berjumlah 34 *case* seperti yang telah dijelaskan pada *use case diagram*.

4.5.1 Login

Pada Tabel 4.3 menjelaskan *use case scenario login* dan semua pengguna memiliki akses kedalam fungsi tersebut. Pengguna tersebut adalah admin, ustadz, *murabbi* dan wali santri. Fungsi *use case login* adalah untuk menyediakan izin pada admin, ustadz, *murabbi* dan wali santri mengakses sistem menurut hak istimewanya.

Tabel 4.3 Use Case Scenario Login

Aktor	Pengguna
Objektif	Memberi izin aktor untuk mengakses sistem menurut hak istimewa tiap pengguna .
Kondisi Awal	Aktor sudah membuka <i>form Login</i> .
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor melakukan pengisian <i>e-mail</i> dan <i>password</i> lalu menekan tombol <i>Login</i>. 2. Sistem memperlihatkan <i>dashboard</i> menurut hak istimewa tiap pengguna.
Aliran Alternatif	<ol style="list-style-type: none"> 1. Jika tombol <i>Login</i> ditekan oleh aktor namun <i>e-mail</i> dan <i>password</i> tidak terisi, sistem memperlihatkan pesan "<i>please fill out this field</i>". 2. Jika ditemukan kesalahan dalam pengisian <i>e-mail</i> atau <i>password</i> maka akan memperlihatkan pesan "<i>These Credentials do not match our records</i>".
Kondisi Akhir	Aktor mampu mendapat hak akses sistem menurut hak istimewa tiap pengguna.

4.5.2 Logout

Tabel 4.4 menjelaskan *use case scenario logout*. Semua pengguna mempunyai akses untuk fitur ini. *Use case* ini memiliki fungsi untuk membuat para aktor keluar dari sistem.

Tabel 4.4 Use Case Scenario Logout

Aktor	Admin, Ustadz, <i>Murabbi</i> dan Wali Santri
Objektif	Memberi izin aktor untuk keluar dari sistem.



Tabel 4.5 Use Case Scenario Logout (Lanjutan)

Kondisi Awal	Aktor masih ada di dalam sistem menurut hak istimewa tiap pengguna.
Aliran Utama	1. Aktor tekan menu <i>logout</i> . 2. Sistem memproses <i>logout</i> .
Aliran Alternatif	
Kondisi Akhir	Sistem beralih ke halaman login dan aktor tidak dalam keadaan login

4.5.3 Melihat Data Santri

Pada Tabel 4.5 menjelaskan *use case scenario* melihat data santri. Hanya admin yang mampu mengakses fitur ini. *Use case* ini mempunyai fungsi untuk memperlihatkan semua data santri beserta detail data santri.

Tabel 4.5 Use Case Scenario Melihat Data Santri

Aktor	Admin
Objektif	Memberi izin kepada aktor agar bisa melihat data santri.
Kondisi Awal	Sistem memperlihatkan halaman data santri.
Aliran Utama	1. Aktor menekan menu data santri. 2. Sistem memperlihatkan detail data santri yaitu, nomor induk santri dan nama lengkap.
Aliran Alternatif	-
Kondisi Akhir	Aktor berhasil melihat detail data santri.

4.5.4 Mengubah Data Santri

Dalam Tabel 4.6 dijelaskan *use case scenario* Mengubah Data Santri. Admin adalah aktor yang mempunyai akses terhadap fitur ini. *Use case* ini memiliki fungsi untuk membarui data santri.

Tabel 4.6 Use Case Scenario Mengubah Data Santri

Aktor	Admin
Objektif	Memberi izin aktor untuk melakukan perubahan pada data santri.
Kondisi Awal	Aktor telah memilih data santri yang hendak diubah datanya.
Aliran Utama	1. Sistem menampilkan formulir ubah data santri. 2. Aktor melakukan pengisian formulir ubah data santri berisikan nomor induk santri dan nama. 3. Aktor menekan tombol " <i>Save and back</i> ".



Tabel 4.7 Use Case Scenario Mengubah Data Santri (Lanjutan)

Aliran Utama	4. Sistem menyimpan data santri dan memperlihatkan “ <i>The Item has been modified successfully</i> ”, lalu memperlihatkan data santri yang telah diubah datanya.
Aliran Alternatif	Jika Aktor menekan tombol “ <i>Save and back</i> ” namun beberapa data santri belum dilakukan pengisian maka akan memperlihatkan peringatan “Harap diisi” pada kolom yang belum diisi.
Kondisi Akhir	Aktor berhasil mengubah data santri.

4.5.5 Menambah Data Santri

Tabel 4.7 memberikan penjelasan mengenai *use case scenario* Menambah Data Santri. Admin adalah aktor yang mempunyai akses terhadap fitur ini. *Use case* ini memiliki fungsi untuk membuat data santri baru ketika pondok melakukan penerimaan santri baru.

Tabel 4.7 Use Case Scenario Menambah Data Santri

Aktor	Admin
Objektif	Memberi izin aktor untuk melakukan penambahan pada data santri.
Kondisi Awal	Sistem memperlihatkan halaman data santri dan aktor telah menekan tombol buat santri.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor tekan tombol buat santri. 2. Sistem memperlihatkan halaman formulir penambahan data santri berisikan nomor induk santri dan nama lengkap. 3. Aktor melakukan pengisian formulir data santri dan memilih tombol “<i>Save and back</i>”. 4. Sistem melakukan penyimpanan data santri dan menampilkan “<i>The item has been added successfully</i>”, lalu menampilkan data santri yang baru ditambahkan.
Aliran Alternatif	Jika beberapa kolom data santri belum dilakukan pengisian namun sudah menekan tombol “ <i>Save and back</i> ” maka akan memunculkan pesan “Harap diisi” pada kolom yang belum diisi.
Kondisi Akhir	Aktor berhasil menambah data santri.

4.5.6 Menghapus Data Santri

Tabel 4.8 memberikan penjelasan tentang *use case scenario* menghapus data santri. Admin merupakan aktor yang mempunyai hak akses untuk menggunakan fitur ini. *Use case* ini memiliki fungsi membuat data santri yang dikehendaki terhapus.



Tabel 4.8 Use Case Scenario Menghapus Data Santri

Aktor	Admin
Objektif	Memberi izin aktor guna menghapus data santri.
Kondisi Awal	Aktor sudah berada pada halaman data santri.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menentukan data santri yang hendak dihilangkan. 2. Aktor melakukan penekanan pada tombol “Delete” 3. Sistem memperlihatkan peringatan untuk melanjutkan atau membatalkan proses penghapusan. 4. Aktor melakukan penekanan pada tombol “Delete” 5. Sistem melakukan penghapusan data santri dari basis data kemudian mengalihkan kembali ke halaman data santri.
Aliran Alternatif	Jika aktor memilih tombol “Cancel” saat konfirmasi penghapusan, maka data santri yang telah dipilih akan dibatalkan proses penghapusannya.
Kondisi Akhir	Aktor berhasil menghapus data santri.

4.5.7. Melihat Data Pengguna

Tabel 4.9 memberikan penjelasan perihal *use case scenario* melihat data Pengguna. Hanya admin yang dapat mengakses fitur ini. *Use case* ini mempunyai fungsi untuk memperlihatkan data pengguna. Pengguna dalam sistem ini termasuk *ustaz*, *murabbi*, dan wali santri.

Tabel 4.9 Use Case Scenario Melihat Data Pengguna

Aktor	Admin
Objektif	Memberi izin aktor melihat detail data pengguna.
Kondisi Awal	Sistem memperlihatkan halaman pengguna.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor tekan menu pengguna sistem. 2. Aktor tekan menu pengguna 3. Sistem memperlihatkan data pengguna yang terdiri dari nama, <i>e-mail</i>, dan jabatan.
Aliran Alternatif	-
Kondisi Akhir	Aktor berhasil melihat detail data santri.

4.5.8 Mengubah Data Pengguna

Tabel 4.10 memberikan penjelasan mengenai *use case scenario* mengubah data pengguna. Admin adalah aktor yang mempunyai akses pada fitur ini. *Use case* ini mempunyai fungsi untuk membuat pembaruan pada data pengguna.



Tabel 4.10 Use Case Scenario Mengubah Data Pengguna

Aktor	Admin
Objektif	Memberi izin aktor untuk melakukan perubahan pada data pengguna.
Kondisi Awal	Aktor telah memilih data pengguna yang hendak diubah datanya.
Aliran Utama	<ol style="list-style-type: none"> 1. Sistem menampilkan formulir ubah data pengguna. 2. Aktor mengisi formulir ubah data pengguna berupa nama, e-mail, jabatan 3. Aktor tekan tombol "Save and back". 4. Sistem membuat data pengguna tersimpan lalu memunculkan peringatan "The item has been modified successfully" kemudian mengalihkan kembali ke halaman pengguna.
Aliran Alternatif	Jika tombol "Save and back" telah ditekan namun beberapa data pengguna belum dilakukan pengisian maka akan memperlihatkan peringatan "Please fill out this field"
Kondisi Akhir	Perubahan data pengguna berhasil dilakukan dan sistem mengalihkan ke halaman pengguna dengan data terbaru.

4.5.9 Menambah Data Pengguna

Tabel 4.11 memberikan penjelasan mengenai *use case scenario* menambah data pengguna. Admin adalah aktor yang mempunyai akses ke fitur ini. *Use case* ini memiliki fungsi membuat data pengguna yang baru terdaftar.

Tabel 4.11 Use Case Scenario Menambah Data Pengguna

Aktor	Admin
Objektif	Memberi izin aktor untuk membuat data pengguna baru.
Kondisi Awal	Sistem sudah memperlihatkan halaman data pengguna.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor tekan tombol "buat pengguna" 2. Sistem memperlihatkan halaman formulir penambahan data pengguna yang terdiri dari nama, e-mail, password dan jabatan. 3. Aktor melakukan pengisian formulir dan memilih tombol "Daftarkan Akun" 4. Sistem melakukan penyimpanan data pengguna dan menampilkan "The item has been added successfully" dan beralih ke halaman pengguna
Aliran Alternatif	Jika tombol "Daftarkan Akun" telah ditekan namun data pengguna belum dilakukan pengisian maka sistem memunculkan peringatan "This field is required" pada kolom yang belum diisi.
Kondisi Akhir	Aktor berhasil menambah data pengguna.



4.5.10 Menghapus Data Pengguna

Tabel 4.12 memberikan penjelasan mengenai *use case scenario* menghapus data pengguna. Admin adalah aktor yang mempunyai hak akses ke dalam fitur ini. *Use case* ini mempunyai fungsi untuk membuat data pengguna yang ditentukan terhapus.

Tabel 4.12 Use Case Scenario Menghapus Data Pengguna

Aktor	Admin
Objektif	Memberi izin aktor untuk melakukan penghapusan data pengguna.
Kondisi Awal	Sistem sudah memperlihatkan halaman pengguna.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menentukan data pengguna yang hendak dihilangkan. 2. Aktor tekan tombol "Delete". 3. Sistem memberikan peringatan untuk melanjutkan proses penghapusan. 4. Aktor tekan tombol "Delete". 5. Sistem melakukan penghapusan pada data yang dipilih. 6. Sistem memberi peringatan "Item Deleted". 7. Sistem mengalihkan kembali ke halaman pengguna dengan data terbaru.
Aliran Alternatif	Jika aktor memilih tombol "Cancel" saat konfirmasi penghapusan, maka proses penghapusan data yang dipilih batal.
Kondisi Akhir	Sistem telah menghapus data yang terpilih.

4.5.11 Melihat Data Mata Pelajaran

Tabel 4.13 menjelaskan *use case scenario* melihat data mata pelajaran. Admin merupakan aktor yang mempunyai akses pada fitur ini. *Use case* ini memiliki fungsi guna memperlihatkan data pelajaran yang terdaftar pada kurikulum di Al-Izzah.

Tabel 4.13 Use Case Scenario Melihat Data Mata Pelajaran

Aktor	Admin
Objektif	Memberi izin aktor untuk melihat data mata pelajaran.
Kondisi Awal	Sistem menampilkan halaman mata pelajaran.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menekan menu mata pelajaran. 2. Sistem memperlihatkan data mata pelajaran.
Aliran Alternatif	
Kondisi Akhir	Sistem menampilkan data mata pelajaran.



4.5.12 Mengubah Data Mata Pelajaran

Tabel 4.14 memberikan penjelasan mengenai *use case scenario* mengubah data mata pelajaran. Admin adalah aktor yang mempunyai akses ke fitur ini. *Use case* ini memiliki fungsi guna membuat pembaruan pada data mata pelajaran yang terdaftar.

Tabel 4.14 Use Case Scenario Mengubah Data Mata Pelajaran

Aktor	Admin
Objektif	Memberi izin aktor mengubah data mata pelajaran.
Kondisi Awal	Aktor telah membuka halaman mata pelajaran.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor memilih data mata pelajaran yang hendak diubah. 2. Sistem memperlihatkan formulir ubah data mata pelajaran. 3. Aktor mengisi formulir ubah data mata pelajaran lalu menekan tombol "Save and back". 4. Sistem melakukan penyimpanan pada data yang telah diubah kemudian menampilkan "The Item has been modified Successfully" lalu mengalihkan ke halaman mata pelajaran.
Aliran Alternatif	Jika tombol "Save and back" telah ditekan namun ada kolom yang belum dilakukan pengisian maka sistem memunculkan pesan "Please fill out this field".
Kondisi Akhir	Aktor berhasil mengubah data mata pelajaran.

4.5.13 Menambah Data Mata Pelajaran

Tabel 4.15 memberikan penjelasan mengenai *use case scenario* menambah data mata pelajaran. Admin adalah aktor yang mempunyai hak akses ke fitur ini. *Use case* ini memiliki fungsi untuk menambah mata pelajaran terbaru dari pondok.

Tabel 4.15 Use Case Scenario Menambah Data Mata Pelajaran

Aktor	Admin
Objektif	Memberi izin aktor melakukan penambahan data mata pelajaran.
Kondisi Awal	Aktor telah membuka halaman mata pelajaran.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menekan tombol "Buat Mata Pelajaran". 2. Sistem memperlihatkan formulir mata pelajaran baru. 3. Aktor mengisi formulir tambah data mata pelajaran kemudian memilih tombol "Save and back". 4. Sistem menyimpan data mata pelajaran baru dan menampilkan halaman mata pelajaran dengan data yang baru disimpan.
Aliran Alternatif	Jika tombol "Tambah" telah ditekan namun ada data mata pelajaran yang belum dilakukan pengisian maka akan memperlihatkan pesan "Please fill out this field".

Tabel 4.15 *Use Case Scenario* Menambah Data Mata Pelajaran (Lanjutan)

Kondisi Akhir	Aktor berhasil menambah data mata pelajaran.
----------------------	--

4.5.14 Menghapus Data Mata Pelajaran

Tabel 4.16 memberikan penjelasan mengenai *use case scenario* menghapus data mata pelajaran. Admin adalah aktor yang mempunyai akses ke dalam fitur ini. *Use case* ini memiliki fungsi membuat data mata pelajaran yang dikehendaki terhapus.

Tabel 4.16 *Use Case Scenario* Menghapus Data Mata Pelajaran

Aktor	Admin
Objektif	Memberi izin aktor guna menghapus data mata pelajaran.
Kondisi Awal	Aktor telah membuka halaman mata pelajaran.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor memilih mata pelajaran yang hendak dihilangkan. 2. Aktor menekan tombol "Delete". 3. Sistem memberikan peringatan untuk melanjutkan atau membatalkan proses penghapusan. 4. Aktor menekan tombol "Delete". 5. Sistem menghapus data mata pelajaran dari basis data dan memperlihatkan data mata pelajaran yang telah diubah.
Aliran Alternatif	Jika tombol "Cancel" dipilih proses penghapusan akan dibatalkan.
Kondisi Akhir	Sistem menghapus mata pelajaran yang dipilih.

4.5.15 Melihat Data Kepemimpinan

Tabel 4.17 memberikan penjelasan perihal *use case scenario* melihat data kepemimpinan. Hanya admin yang memiliki akses ke fitur ini. *Use case* ini mempunyai fungsi guna memperlihatkan data penilaian kepemimpinan.

Tabel 4.17 *Use Case Scenario* Melihat Data Kepemimpinan

Aktor	Admin
Objektif	Memberi izin aktor untuk melihat data penilaian kepemimpinan.
Kondisi Awal	Aktor telah membukan halaman data kepemimpinan.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor tekan menu administrasi. 2. Aktor tekan menu kepemimpinan. 3. Sistem memperlihatkan data nilai kepemimpinan.
Aliran Alternatif	
Kondisi Akhir	Sistem memoerlihatkan data kepemimpinan.



4.5.16 Mengubah Data Kepemimpinan

Tabel 4.18 memberikan penjelasan mengenai *use case scenario* mengubah data kepemimpinan. Admin adalah aktor yang memiliki akses pada fitur ini. *Use case* ini memiliki fungsi guna membuat pembaruan pada data kepemimpinan.

Tabel 4.18 Use Case Scenario Mengubah Data Kepemimpinan

Aktor	Admin
Objektif	Memberi izin aktor untuk melakukan perubahan pada data kepemimpinan.
Kondisi Awal	Aktor berada di halaman data kepemimpinan.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menentukan data mata pelajaran yang hendak diubah. 2. Sistem memperlihatkan formulir ubah data kepemimpinan. 3. Aktor mengisi formulir ubah data kepemimpinan lalu menekan tombol "Ubah" 4. Sistem melakukan penyimpanan pada data yang diubah dan memperlihatkan pesan "<i>The item has been modified successfully</i>", lalu mengalihkan kembali ke halaman data kepemimpinan dengan data yang telah diubah.
Aliran Alternatif	Jika Aktor memilih tombol "Ubah" tetapi beberapa kolom data kepemimpinan belum dilakukan pengisian maka akan memperlihatkan pesan "Harap diisi" pada kolom yang belum diisi.
Kondisi Akhir	Aktor berhasil mengubah data kepemimpinan.

4.5.17 Menambah Data Kepemimpinan

Tabel 4.19 memberikan penjelasan mengenai *use case scenario* menambah data kepemimpinan. Admin adalah aktor yang memiliki akses pada fitur ini. *Use case* ini mempunyai fungsi guna membuat data kepemimpinan yang baru.

Tabel 4.19 Use Case Scenario Menambah Data Kepemimpinan

Aktor	Admin
Objektif	Memberi izin aktor untuk menambah data kepemimpinan.
Kondisi Awal	Aktor berada di halaman data kepemimpinan.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor tekan tombol "Buat Kepemimpinan". 2. Sistem memperlihatkan formulir data kepemimpinan. 3. Aktor melakukan pengisian pada formulir data kepemimpinan lalu menekan tombol "<i>Save and back</i>". 4. Sistem membuat data kepemimpinan baru tersimpan lalu mengalihkan ke halaman kepemimpinan dengan data terbaru.

Tabel 4.19 *Use Case Scenario* Menambah Data Kepemimpinan (Lanjutan)

Aliran Alternatif	Jika tombol “Save and back” ditekan namun ada data kepemimpinan yang belum dilakukan pengisian, sistem akan memunculkan pesan “Please fill out this field”.
Kondisi Akhir	Aktor berhasil menambah data kepemimpinan.

4.5.18 Menghapus Data Kepemimpinan

Tabel 4.20 memberikan penjelasan mengenai *use case scenario* menghapus data kepemimpinan. Hanya admin yang memiliki akses pada fitur ini. *Use case* ini mempunyai fungsi membuat data kepemimpinan yang dikehendaki terhapus.

Tabel 4.20 *Use Case Scenario* Menghapus Data Kepemimpinan

Aktor	Admin
Objektif	Memberi izin aktor guna menghapus data kepemimpinan.
Kondisi Awal	Aktor sudah berada di halaman data kepemimpinan.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menentukan data kepemimpinan yang hendak dihapus. 2. Aktor tekan tombol “Hapus”. 3. Sistem memperlihatkan peringatan persetujuan penghapusan. 4. Aktor tekan tombol “Hapus”. 5. Sistem melakukan penghapusan data kepemimpinan yang dipilih lalu mengalihkan kembali ke halaman kepemimpinan.
Aliran Alternatif	Jika tombol “batal” ditekan maka proses penghapusan data kepemimpinan dibatalkan.
Kondisi Akhir	Aktor berhasil menghapus data kepemimpinan.

4.5.19 Melihat Data Hafalan

Tabel 4.21 memberikan penjelasan perihal *use case scenario* melihat data hafalan. Admin adalah aktor yang memiliki akses pada fitur ini. *Use case* ini mempunyai fungsi untuk memperlihatkan juz berapa yang harus dihafal santri.

Tabel 4.21 *Use Case Scenario* Melihat Data Hafalan

Aktor	Admin
Objektif	Memberi izin aktor untuk melihat data hafalan.
Kondisi Awal	Aktor sudah berada di halaman data hafalan.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor tekan menu administrasi 2. Aktor tekan menu hafalan 3. Sistem memperlihatkan data hafalan.
Aliran Alternatif	Brawijaya
Kondisi Akhir	Aktor berhasil melihat data hafalan.



4.5.20 Mengubah Data Hafalan

Tabel 4.22 memberikan penjelasan mengenai *use case scenario* mengubah data hafalan. Admin adalah aktor yang memiliki akses pada fitur ini. *Use case* ini memiliki fungsi untuk untuk mengubah juz pada data hafalan.

Tabel 4.22 Use Case Scenario Mengubah Data Hafalan

Aktor	Admin
Objektif	Memberi izin aktor untuk membarui juz pada data hafalan.
Kondisi Awal	Aktor sudah berada di halaman data hafalan.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menentukan data mata pelajaran yang hendak diubah. 2. Sistem memperlihatkan formulir ubah data hafalan. 3. Aktor melakukan pengisian formulir ubah data hafalan lalu memilih tombol "Save and back". 4. Sistem melakukan penyimpanan pada data yang dipilih dan memunculkan pesan "The item has been modified successfully", lalu mengalihkan kembali ke halaman data hafalan.
Aliran Alternatif	Jika tombol "Save and back" ditekan namun ada data hafalan yang belum dilakukan pengisian, sistem memunculkan pesan "Please fill out this field".
Kondisi Akhir	Aktor berhasil mengubah data hafalan.

4.5.21 Menambah Data Hafalan

Tabel 4.23 memberikan penjelasan perihal *use case scenario* menambah data hafalan. Admin adalah aktor yang memiliki akses pada fitur ini. *Use case* ini memiliki fungsi membuat juz baru pada data hafalan.

Tabel 4.23 Use Case Scenario Menambah Data Hafalan

Aktor	Admin
Objektif	Memberi izin aktor untuk membuat juz baru pada data hafalan.
Kondisi Awal	Aktor sudah berada di halaman data hafalan.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menekan tombol "tambah data". 2. Sistem memperlihatkan halaman tambah data hafalan. 3. Aktor melakukan pengisian pada formulir data hafalan kemudian memilih tombol "Save and back". 4. Sistem melakukan penyimpanan pada data hafalan baru dan mengalihkan kembali ke halaman data hafalan dengan data yang baru disimpan.



Tabel 4.23 Use Case Scenario Menambah Data Hafalan (Lanjutan)

Aliran Alternatif	Jika tombol "Save and back" ditekan namun ada data hafalan yang belum dilakukan pengisian sistem memperlihatkan pesan "Please fill out this field".
Kondisi Akhir	Sistem telah melakukan penambahan pada data hafalan.

4.5.22 Menghapus Data Hafalan

Tabel 4.24 memberikan penjelasan mengenai *use case scenario* menghapus data hafalan. Admin adalah aktor yang memiliki akses pada fitur ini. *Use case* ini memiliki fungsi untuk menghapus juz yang tidak lagi dimasukkan dalam syarat hafalan.

Tabel 4.24 Use Case Scenario Menghapus Data Hafalan

Aktor	Admin
Objektif	Memberi izin aktor guna menghapus data hafalan
Kondisi Awal	Aktor sudah berada di halaman data hafalan
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menentukan data hafalan yang hendak dihapus. 2. Aktor menekan tombol "Delete". 3. Sistem memberikan peringatan untuk melanjutkan atau membatalkan proses penghapusan. 4. Aktor tekan tombol "Delete". 5. Sistem melakukan penghapusan data hafalan dan mengalihkan kembali ke halaman hafalan dengan data terbaru.
Aliran Alternatif	Jika tombol "batal" ditekan, sistem membatalkan proses penghapusan data hafalan.
Kondisi Akhir	Sistem telah menghapus data hafalan yang dipilih

4.5.23 Melihat Pelanggaran Santri

Pada Tabel 4.25 memberikan penjelasan mengenai *use case scenario* melihat pelanggaran santri. *Murobbi* adalah aktor yang mempunyai akses ke fitur ini. *Use case* ini mempunyai fungsi untuk memperlihatkan pelanggaran yang telah dilakukan oleh santri.

Tabel 4.25 Use Case Scenario Melihat Pelanggaran Santri

Aktor	<i>Murobbi</i>
Objektif	Memberi izin aktor untuk melihat pelanggaran santri
Kondisi Awal	Aktor telah mengakses halaman pelanggaran santri
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor tekan menu pelanggaran santri. 2. Sistem memperlihatkan data pelanggaran santri.



Tabel 4.25 Use Case Scenario Melihat Pelanggaran Santri (Lanjutan)

Aliran Alternatif	-
Kondisi Akhir	Aktor berhasil melihat data pelanggaran santri.

4.5.24 Menambah Pelanggaran Santri

Tabel 4.26 memberikan penjelasan mengenai *use case scenario* menambah pelanggaran santri. *Murobbi* adalah aktor yang mempunyai akses ke fitur ini. *Use case* ini memiliki fungsi untuk menambahkan pelanggaran yang dilakukan santri.

Tabel 4.26 Use Case Scenario Menambah Pelanggaran Santri

Aktor	Ustaz, <i>Murobbi</i>
Objektif	Memberi izin aktor untuk menambah pelanggaran santri
Kondisi Awal	Aktor telah mengakses halaman pelanggaran santri
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor memilih santri yang melakukan pelanggaran. 2. Sistem memperlihatkan jenis pelanggaran. 3. Aktor memilih pelanggaran yang dilakukan santri 4. Aktor tekan tombol "Kirim". 5. Sistem membuat data pelanggaran santri tersimpan lalu sistem mengalihkan kembali ke halaman pelanggaran. 6. Sistem mengirimkan notifikasi pelanggaran santri ke Wali Santri dari santri yang melakukan pelanggaran.
Aliran Alternatif	Jika tombol "Kirim" tetapi aktor belum memilih pelanggaran yang dilakukan santri maka sistem memunculkan pesan " <i>The field is required</i> ".
Kondisi Akhir	Aktor berhasil menambah pelanggaran santri.

4.5.25 Melihat Nilai Mata Pelajaran

Tabel 4.27 memberikan penjelasan mengenai *use case scenario* melihat nilai mata pelajaran. Aktor yang mampu mengakses fitur ini adalah ustaz dan wali santri. *Use case* ini memiliki fungsi memperlihatkan nilai mata pelajaran.

Tabel 4.27 Use Case Scenario Melihat Nilai Mata Pelajaran

Aktor	Ustaz, Wali Santri
Objektif	Memberi izin aktor untuk melihat nilai mata pelajaran
Kondisi Awal	Aktor sudah berada di halaman mata pelajaran
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor melakukan pemilihan pada menu nilai penilaian 2. Aktor memilih pilihan nilai yaitu nilai PR, Nilai ulangan, Nilai UAS/UTS



Tabel 4.27 Use Case Scenario Melihat Nilai Mata Pelajaran (Lanjutan)

Aliran Utama	3. Aktor memilih kelas dan mata pelajaran yang di ambil. 4. Sistem menampilkan nilai mata pelajaran
Aliran Alternatif	
Kondisi Akhir	Aktor berhasil melihat nilai mata pelajaran

4.5.26 Menambah Nilai Mata Pelajaran

Tabel 4.28 memberikan penjelasan mengenai *use case scenario* menambah nilai mata pelajaran. Aktor yang mampu mengakses fitur ini adalah Ustaz. Use case ini memiliki fungsi untuk menambah nilai mata pelajaran.

Tabel 4.28 Use Case Scenario Menambah Nilai Mata Pelajaran

Aktor	Ustaz
Objektif	Memberi izin aktor menambah nilai mata pelajaran.
Kondisi Awal	Aktor sudah berada di halaman mata pelajaran
Aliran Utama	1. Aktor memilih antara nilai PR, nilai ulangan, dan nilai UTS/UAS 2. Sistem menampilkan kelas dan mata pelajaran dari ustaz yang sedang <i>login</i> 3. Aktor memilih kelas dan mata pelajaran yang hendak diisikan nilai 4. Sistem menampilkan draft nilai mata pelajaran tersebut 5. Aktor tekan tombol "buat draft baru" 6. Sistem menampilkan data santri di kelas dan mata pelajaran tersebut 7. Aktor menambah nilai setiap santri 8. Aktor menekan tombol "Simpan".
Aliran Alternatif	1. Jika tombol "Simpan" ditekan namun nilai dari santri ada yang kosong, sistem memunculkan nama santri dengan nilai kosong dan menampilkan konfirmasi untuk tetap menyimpan draft nilai tersebut. 2. Jika aktor memilih tombol "Reset" maka nilai dari setiap santri akan diisi kembali dengan nilai kosong.
Kondisi Akhir	Aktor berhasil menambah nilai mata pelajaran.

4.5.27 Mengubah Nilai Mata Pelajaran

Tabel 4.29 memberikan penjelasan mengenai *use case scenario* mengubah nilai mata pelajaran. Aktor yang mampu mengakses fitur ini adalah Ustaz. Use case ini memiliki fungsi untuk mengubah nilai mata pelajaran.



Tabel 4.29 Use Case Scenario Mengubah Nilai Mata Pelajaran

Aktor	Ustaz
Objektif	Memberi izin aktor untuk mengubah nilai mata pelajaran.
Kondisi Awal	Aktor sudah berada di halaman penilaian.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menentukan kelas dan mata pelajaran yang hendak diisikan nilai 2. Sistem menampilkan draft nilai mata pelajaran tersebut 3. Aktor menekan tombol "edit" 4. Sistem memperlihatkan nilai santri di kelas dan mata pelajaran tersebut 5. Aktor mengubah nilai setiap santri 6. Aktor menekan tombol "Simpan"
Aliran Alternatif	<ol style="list-style-type: none"> 1. Jika tombol "Simpan" ditekan namun nilai dari santri ada yang kosong, sistem menampilkan nama santri dengan nilai kosong dan menampilkan konfirmasi untuk tetap menyimpan draft nilai tersebut. 2. Jika aktor memilih tombol "Reset" maka nilai dari setiap santri akan diisi kembali dengan nilai kosong.
Kondisi Akhir	Aktor berhasil mengubah nilai mata pelajaran.

4.5.28 Melihat Nilai Kepemimpinan

Tabel 4.30 memberikan penjelasan mengenai *use case scenario* melihat nilai kepemimpinan. *Murobbi* dan wali santri adalah aktor yang mempunyai akses ke fitur ini. Use case ini mempunyai fungsi memperlihatkan nilai kepemimpinan.

Tabel 4.30 Use Case Scenario Melihat Nilai Kepemimpinan

Aktor	<i>Murobbi</i> , Wali Santri
Objektif	Memberi izin aktor melihat nilai kepemimpinan
Kondisi Awal	Aktor telah mengakses halaman nilai kepemimpinan
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menekan menu nilai kepemimpinan 2. Sistem memperlihatkan nilai kepemimpinan
Aliran Alternatif	-
Kondisi Akhir	Sistem menampilkan semua nilai kepemimpinan

4.5.29 Mengubah Nilai Kepemimpinan

Tabel 4.31 memberikan penjelasan mengenai *use case scenario* mengubah nilai kepemimpinan. *Murobbi* adalah aktor yang mempunyai akses ke fitur ini. Use case ini mempunyai fungsi membuat pembaruan pada nilai kepemimpinan.



Tabel 4.31 Use Case Scenario Mengubah Nilai Kepemimpinan

Aktor	<i>Murobbi</i>
Objektif	Memberi izin aktor untuk mengubah nilai kepemimpinan.
Kondisi Awal	Aktor telah membuka halaman nilai kepemimpinan.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor mengubah nilai dengan menambahkan <i>file excel</i> atau mengubah nilai di halaman nilai kepemimpinan untuk setiap mata pelajaran. 2. Aktor memilih tombol “Ubah”.
Aliran Alternatif	Jika tombol “Ubah” ditekan tetapi ada nilai yang belum diisi kembali maka akan memunculkan pesan “ <i>The field is required</i> ”.
Kondisi Akhir	Aktor berhasil mengubah nilai kepemimpinan.

4.5.30 Menambah Nilai Kepemimpinan

Tabel 4.32 memberikan penjelasan mengenai *use case scenario* menambah nilai kepemimpinan. *Murobbi* adalah aktor yang mempunyai akses ke fitur ini. Use case ini mempunyai fungsi membuat nilai kepemimpinan baru.

Tabel 4.32 Use Case Scenario Menambah Nilai Kepemimpinan

Aktor	<i>Ustaz, Murobbi</i>
Objektif	Memberi izin aktor menambah nilai mata pelajaran kepemimpinan.
Kondisi Awal	Aktor telah membuka halaman nilai kepemimpinan.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menambah nilai dengan menambahkan <i>file excel</i> atau menambah nilai di halaman nilai kepemimpinan untuk setiap mata pelajaran. 2. Aktor menekan tombol “Simpan”.
Aliran Alternatif	Jika tombol “Simpan” ditekan namun ada nilai yang belum terisi maka memunculkan pesan “ <i>The field is required</i> ”.
Kondisi Akhir	Aktor berhasil menambah nilai kepemimpinan.

4.5.31 Melihat Nilai Hafalan

Tabel 4.33 memberikan penjelasan perihal *use case scenario* melihat nilai hafalan. Aktor yang dapat mengakses fitur ini adalah *Murobbi*. Use case ini mempunyai fungsi untuk memperlihatkan nilai dari hafalan yang telah dilakukan para santri.



Tabel 4.33 Use Case Scenario Melihat Nilai Hafalan

Aktor	<i>Murobbi</i>
Objektif	Memberi izin aktor melihat nilai hafalan
Kondisi Awal	Aktor telah mengakses halaman nilai hafalan.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menekan menu nilai hafalan 2. Sistem memperlihatkan nilai hafalan
Aliran Alternatif	
Kondisi Akhir	Aktor berhasil melihat semua nilai hafalan

4.5.32 Mengubah Nilai Hafalan

Tabel 4.34 memberikan penjelasan mengenai *use case scenario* mengubah nilai hafalan. Aktor yang dapat mengakses fitur ini adalah *Murobbi*. Use case ini mempunyai fungsi untuk membuat pembaruan pada nilai hafalan santri.

Tabel 4.33 Use Case Scenario Mengubah Nilai Hafalan

Aktor	<i>Murobbi</i>
Objektif	Memberi izin aktor untuk mengubah nilai hafalan.
Kondisi Awal	Aktor telah membuka halaman nilai hafalan.
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor mengubah nilai dengan menambahkan <i>file excel</i> atau mengubah nilai di halaman nilai hafalan untuk setiap mata pelajaran. 2. Aktor memilih tombol "Ubah".
Aliran Alternatif	Jika tombol "Ubah" ditekan namun ada nilai yang belum terisi maka akan memunculkan pesan " <i>The field is required</i> ".
Kondisi Akhir	Aktor berhasil mengubah nilai hafalan

4.5.33 Menambah Nilai Hafalan

Tabel 4.35 memberikan penjelasan mengenai *use case scenario* menambah nilai hafalan. Aktor yang dapat mengakses fitur ini adalah *Murobbi*. Use case ini mempunyai fungsi membuat nilai hafalan santri baru.

Tabel 4.34 Use Case Scenario Menambah Nilai Hafalan

Aktor	<i>Murobbi</i>
Objektif	Memberi izin aktor menambah nilai mata pelajaran hafalan.
Kondisi Awal	Aktor telah membuka halaman nilai hafalan.



Tabel 4.35 Use Case Scenario Menambah Nilai Hafalan (Lanjutan)

Aliran Utama	<ol style="list-style-type: none"> 1. Aktor menambah nilai dengan menambahkan <i>file excel</i> atau menambah nilai di halaman nilai hafalan untuk setiap mata pelajaran. 2. Aktor memilih tombol "Simpan".
Aliran Alternatif	Jika tombol "Simpan" ditekan namun ada nilai yang belum terisi maka akan memunculkan pesan " <i>The field is required</i> ".
Kondisi Akhir	Aktor berhasil menambah nilai hafalan.

4.5.34 Mencetak Rapor

Tabel 4.36 memberikan penjelasan mengenai *use case scenario* mencetak rapor. Admin adalah aktor yang memiliki akses pada fitur ini, Wali Santri. Use case ini memiliki fungsi untuk mencetak rapor semua penilaian dari santri.

Tabel 4.35 Use Case Scenario Mencetak Rapor

Aktor	Admin, Wali Santri
Objektif	Memberi izin aktor untuk mencetak rapor dari seluruh penilaian
Kondisi Awal	Aktor berada di <i>dashboard</i>
Aliran Utama	<ol style="list-style-type: none"> 1. Aktor tekan tombol "Cetak Rapor". 2. Sistem memperlihatkan pilihan "Bulan ini" dan "Semester". 3. Aktor memilih pilihan antara "Bulan ini" dan "Semester". 4. Sistem memperlihatkan <i>file pdf</i> untuk di unduh oleh Aktor 5. Aktor mengunduh <i>file pdf</i>.
Aliran Alternatif	<ol style="list-style-type: none"> 1. Jika Aktor memilih pilihan "Bulan ini" tetapi waktu pembelajaran belum memasuki bulan yang dipilih maka sistem menampilkan "Nilai Belum Ada". 2. Jika Aktor memilih pilihan "Semester" tetapi waktu pembelajaran belum mencapai akhir semester maka sistem menampilkan "Nilai Belum Ada".
Kondisi Akhir	Aktor berhasil mengunduh rapor.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

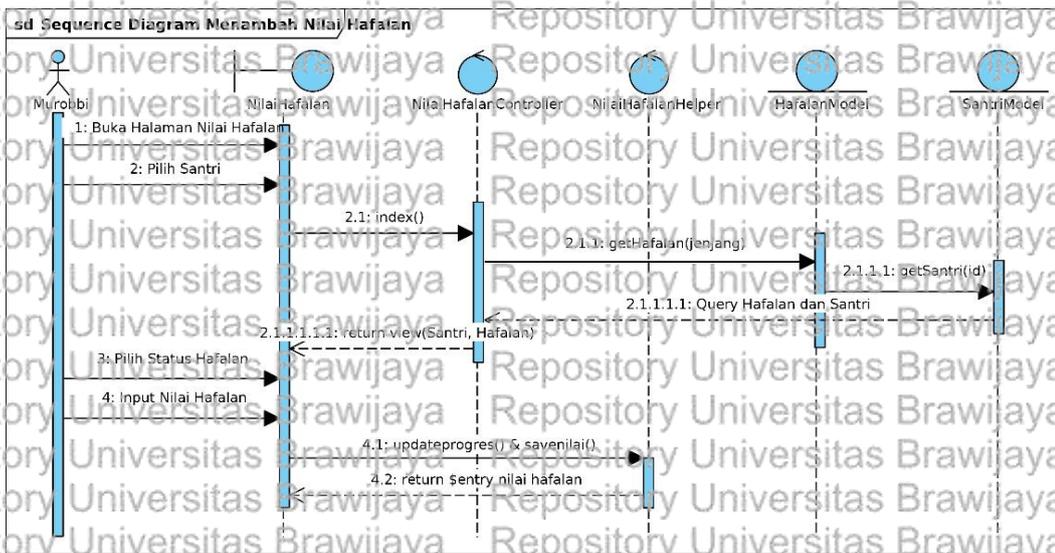
Tahap ini menjelaskan beberapa tahapan dalam proses perancangan. Tahapan tersebut merangkum perancangan arsitektur, perancangan data, perancangan algoritme dan perancangan antarmuka. Pada tahap perancangan arsitektur akan ditampilkan *sequence diagram* dan *class diagram*.

5.1.1 Perancangan Arsitektur

Tahap ini memberikan penjelasan secara rinci mengenai *sequence diagram* dan *class diagram*. *Sequence diagram* memberikan penjelasan mengenai alur pesan antar objek dalam sistem. Selanjutnya, beberapa *sequence diagram* yang akan diberikan penjelasannya akan mewakili proses utama dari sistem penilaian santri terpadu. Ketiga proses tersebut yakni menambahkan nilai hafalan, menambahkan data santri, dan menambah pelanggaran santri. Selain itu, akan ada penjelasan mengenai *class diagram* yang menggambarkan *class-class* beserta hubungannya sebagai penyusunnya.

5.1.1.1 Sequence Diagram Menambah Nilai Hafalan

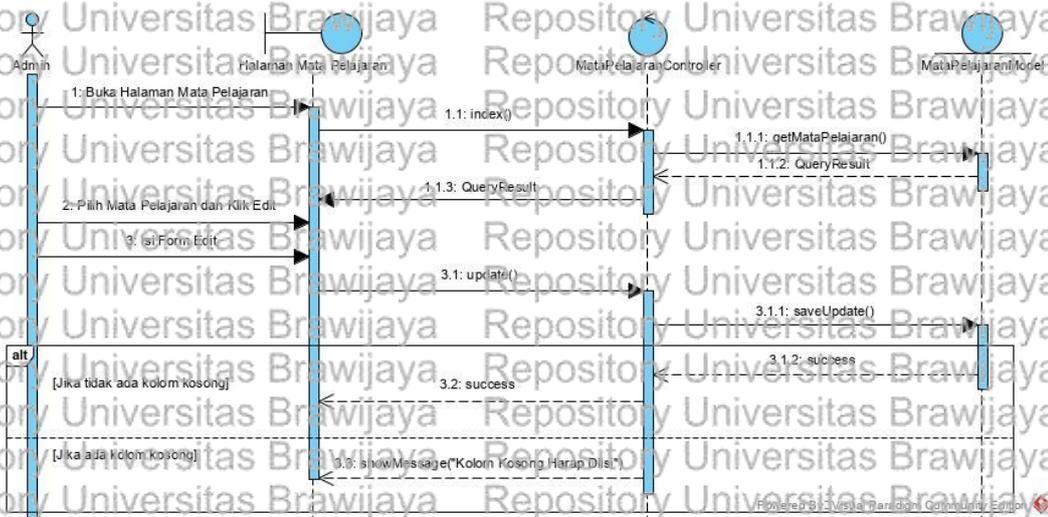
Pada Gambar 5.1 ditampilkan *Sequence diagram* untuk menambah nilai hafalan. Pada *sequence diagram* menambah nilai hafalan dijelaskan interaksi antara objek yang berada di dalam sistem untuk menambahkan nilai dari juz yang dihafalkan oleh santri. *Sequence diagram* ini terdiri dari 5 objek yaitu, 1 *boundary* NilaiHafalan, 2 *controller* NilaiHafalanController dan NilaiHafalanHelper, dan 2 model *HaflanModel*, *SantriModel*. *Murobbi* melakukan *login* untuk mengakses halaman penilaian hafalan, setelah masuk ke halaman penilaian hafalan *murobbi* memilih santri yang telah melakukan setoran hafalan pada juz tertentu, batasan juz untuk setiap santri berbeda tergantung jenjang mereka. Setelah memilih santri yang akan dilakukan penilaian hafalannya, *murobbi* memilih status hafalan yang sudah dilakukan oleh santri tersebut, status dari penambahan nilai hafalan terbagi menjadi 2, *progress* hafalan dan khatam. *Progress* hafalan menandakan bahwa santri belum menyelesaikan hafalannya pada juz tersebut, dan *murobbi* mengisikan halaman terakhir santri menghafal pada juz tersebut. Khatam menandakan bahwa santri sudah menyelesaikan hafalannya pada juz tersebut. *murobbi* mengisikan nilai kesempurnaan hafalan santri tersebut yang tergolong menjadi 4 predikat, *Mumtaz*, *Jayyid Jiddan*, *Jayyid*, dan *Rosib*. Setelah pengisian nilai selesai dilakukan *murobbi* dapat akses untuk menginformasikan ke wali santri bahwa santri dari wali santri tersebut telah melakukan setoran hafalan dengan nilai tersebut.



Gambar 5.1 Sequence Diagram Menambah Nilai Hafalan

5.1.1.2 Sequence Diagram Mengubah Data Mata Pelajaran

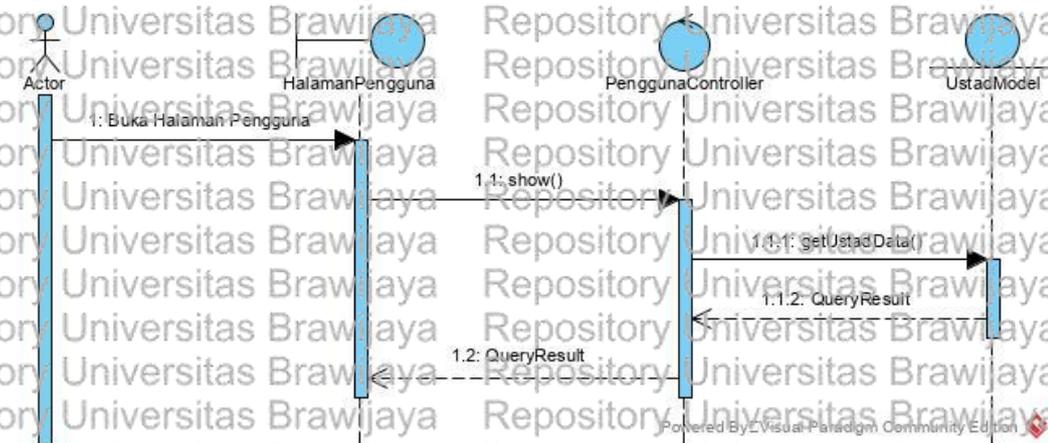
Sequence diagram melihat data pengguna dapat dilihat pada Gambar 5.2. Sequence diagram ini memperlihatkan interaksi objek di dalam sistem untuk mendapatkan data dari mata pelajaran yang ada pada kurikulum Al-izzah. Terandung 3 objek dari sequence diagram ini, objek pertama yaitu *boundary* MataPelajaran, objek kedua yaitu *controller* MataPelajaranController, dan objek terakhir yaitu *model* MataPelajaranModel. Untuk mendapatkan data dari mata pelajaran interaksi pertama dari aktor mengakses halaman mata pelajaran dan masuk ke *boundary* MataPelajaran dengan memanggil *method* *index* pada *controller* MataPelajaranController. Selanjutnya, dilakukan proses *query* data mata pelajaran dengan memanggil *method* *getMataPelajaran* diteruskan dengan pengembalian *query* ke MataPelajaranController untuk ditampilkan kembali di dalam *boundary* MataPelajaran. Kemudian, aktor memilih mata pelajaran yang akan diubah datanya dan mengisi *form edit* dari mata pelajaran yang telah dipilih. Setelah melakukan penyimpanan pada data mata pelajaran yang telah diubah, sistem melakukan validasi dengan melakukan pemeriksaan kolom yang masih kosong. Jika semua kolom dalam *form edit* telah terisi, maka sistem menyimpan data mata pelajaran tersebut, jika ada kolom dari *form edit* yang belum terisi maka *controller* MataPelajaranController akan mengembalikan peringatan untuk mengisi kolom yang belum diisi oleh aktor.



Gambar 5.2 Sequence Diagram Mengubah Data Mata Pelajaran

5.1.1.3 Sequence Diagram Melihat Data Pengguna

Sequence diagram melihat data pengguna ditampilkan pada Gambar 5.3. Sequence diagram ini menampilkan interaksi antar objek pada sistem untuk mendapatkan data pengguna. Dalam sequence diagram ini terdapat tiga objek yang terdiri dari satu boundary yaitu HalamanPengguna, satu controller yaitu PenggunaController, dan satu model yaitu PenggunaModel. Pengguna yang termasuk dalam sequence diagram ini adalah aktor ustaz, murobbi, dan wali santri. Untuk dapat menampilkan pengguna dari sistem ini dimulai dari boundary HalamanPengguna yang ketika dibuka memanggil method index di dalam controller PenggunaController. Method tersebut berguna untuk mendapatkan query pengguna dari PenggunaModel, dan menampilkannya pada boundary HalamanPengguna.



Gambar 5.3 Sequence Diagram Melihat Data Pengguna

5.1.1.4 Class Diagram Sistem Penilaian Santri

Perancangan class diagram merepresentasikan class demi class yang terdapat pada sistem. perancangan ini bertujuan untuk melukiskan pembentukan objek dan hubungan yang dihasilkan. Perancangan class diagram terdiri dari dua



perancangan, perancangan detail dan perancangan umum. Perancangan umum memberikan gambaran dari hubungan antar *class*. Perancangan detail, memberikan gambaran dari karakter dan operasi yang ada pada *class*.

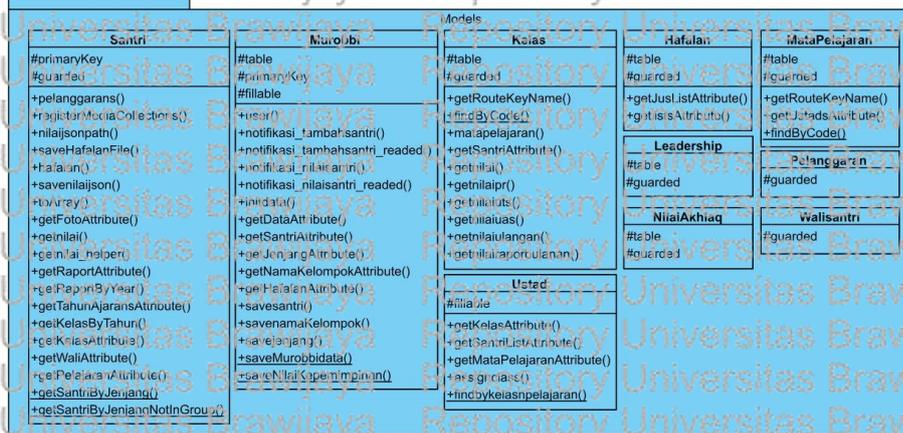
1. Perancangan Umum

Perancangan umum bertujuan untuk menguraikan deskripsi dari objek-objek yang terbentuk beserta hubungannya. Pada perancangan umum, di deskripsikan hubungan antar *class* yang melibatkan bagian dari *controller*. Pada admin terdapat 12 *controller* yaitu, *LeadershipCrudController*, *HafalanCrudController*, *PenggunaControlller*, *RaportSemesterTrait*, *MurobbiController*, *SantriCrudController*, *MataPelajaranCrudController*, *DetailNilaiKelasController*, *RaportBulananTrait*, *KelasCrudController*, dan *AdminController*. Pada aktor ustaz terdapat 6 *controller* yaitu, *NilaiUASSantriController*, *NilaiPRSantriController*, *DraftController*, *NilaiUTSSantriController*, *NilaiULANGANSantriController*, *DataKelasController*. Pada aktor *murobbi* terdapat 4 *controller* yaitu, *DataMurobbiController*, *LeadershipController*, *NilaiHafalanController*, *PelanggaranController*. Pada proses autentikasi aktor pada sistem terdapat 6 *controller* yaitu, *ConfirmPasswordController*, *ForgotPasswordController*, *LoginController*, *RegisterController*, *ResetPasswordController*, *VerificationController*. Pada aktor wali santri terdapat 2 *controller* yaitu, *WaliNilaiSantriController*, *RaportSantriController*. Pada bagian santri terdapat 3 *controller* yaitu, *RaportController*, *NilaiController*, dan *DataController*. Pada bagian *model*, dilakukan generalisasi kepada *EdasWaffleModel*. Pemodelan *class diagram* pada perancangan umum digambarkan melalui Gambar 5.4.



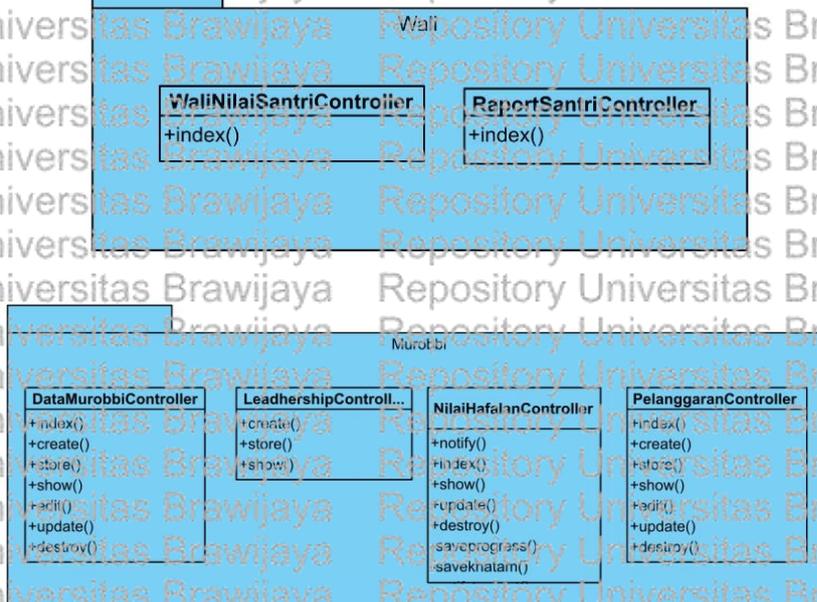
2. Perancangan Detail

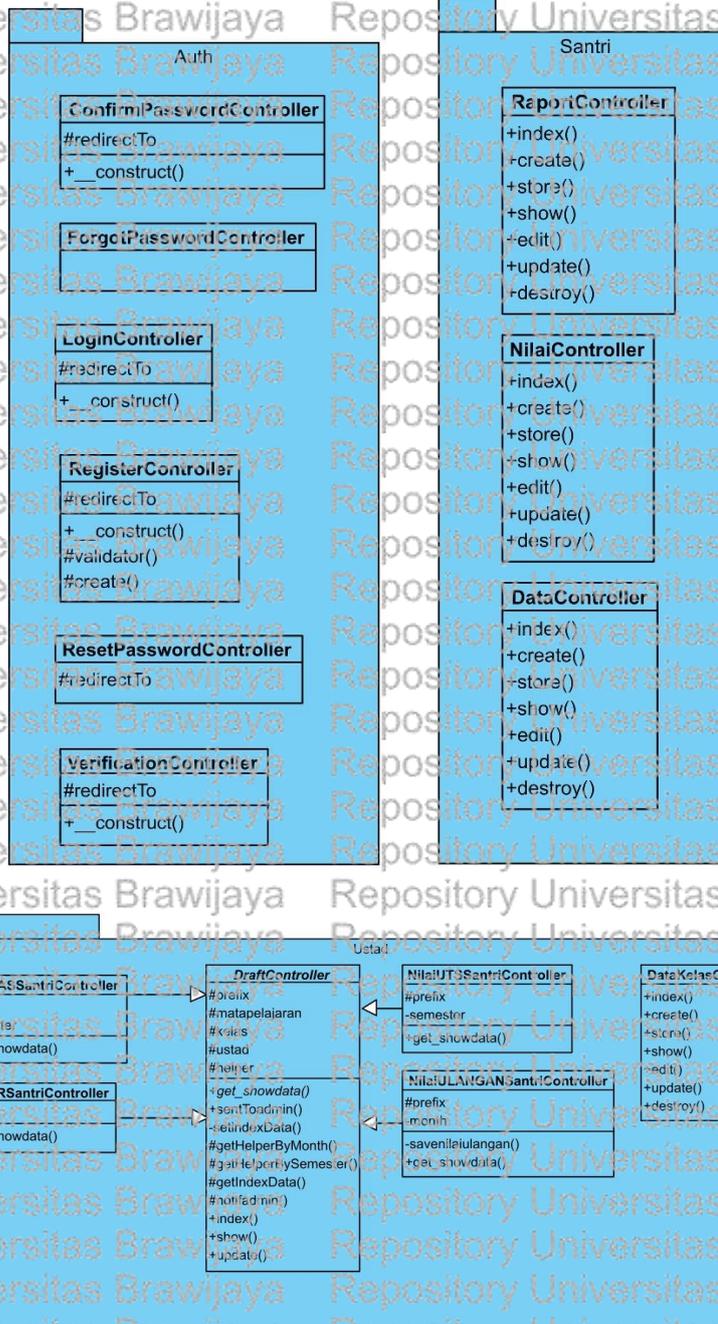
Proses definisi *attribute* dan *method* yang ada di tiap *class* dilakukan pada tahap perancangan detail. Perancangan ini ditujukan pada bagian *model* dan *controller*. Pada bagian model terdapat sepuluh *class*. Perancangan detail pada *class* model digambarkan pada Gambar 5.5.



Gambar 5.5 Class Diagram Model.

Perancangan detail yang melibatkan atribut dan operasi pada *class-class* dalam *controller* digambarkan dalam Gambar 5.6. Perancangan ini mendeskripsikan *method* dari setiap atribut.





Gambar 5.6 Class Diagram Controller.

5.1.2 Perancangan Data

Perancangan data menggunakan Entity Relation Diagram (ERD) untuk membuat model. Relasi antar entitas data sistem ditampilkan di dalam ERD yang dilakukan pada tahap perancangan data. Gambar 5.7 adalah ERD sistem penilaian santri.



Gambar 5.7 Perancangan Entity Relational Diagram

Dalam Gambar 5.7 ditampilkan 7 entitas yang ada pada sistem ini yaitu pengguna, ustad, *murobbi*, santri, kelas, pelanggaran, walisantri. Entitas pengguna memiliki 8 atribut yaitu: id, name, email, password, email_verified_at, created_at, updated_at, remember_token. Entitas ustad memiliki 5 atribut yaitu; id, nama, data_ustad, created_at, updated_at. Entitas *murobbi* memiliki 5 atribut yaitu; id, nama, data_murobbi, created_at, updated_at. Entitas santri memiliki 5 atribut yaitu; id, nis, nama, created_at, updated_at. Entitas kelas memiliki 7 atribut yaitu; id, jenjang, jurusan, nama, kode, created_at, updated_at. Entitas pelanggaran memiliki 5 atribut yaitu; id, pelanggaran, hukuman, created_at, updated_at. Entitas walisantri memiliki 3 atribut yaitu; id, created_at, updated_at.

5.1.3 Perancangan Komponen

Sub sistem yang rinci dari setiap bagian *software* akan dipaparkan pada tahap perancangan komponen. Pada perancangan komponen akan memaparkan alur algoritme. Pada penelitian ini, akan memaparkan tiga algoritme *method* dalam sistem yang dikembangkan. Ketiga *method* tersebut terlibat dalam perancangan tiga fitur utama pada sistem.

5.1.3.1 Perancangan Komponen *Method updateprogres dan savenilai*

Pada Tabel 5.1 memaparkan algoritme *method updateprogres* dan *savenilai* yang digunakan untuk memperbarui nilai hafalan ketika melakukan setoran hafalan kepada *murobbi*. Algoritme ini melakukan *looping* data hafalan untuk mengambil semua data hafalan. Pada *update* progres hafalan hanya memasukan inputan berupa *integer* yang menandakan halaman terakhir hafalan disetorkan. Pada proses pemberian nilai hafalan dilakukan perulangan untuk pengecekan nilai mengklasifikasikan masing-masing nilai masuk ke kategori nilai mumtaz, jayyid jiddan, jayyid, atau rosib. Nama *class* NilaiHafalan dan nama *method* *updateprogres* dan *savenilai*.



Tabel 5.1 Pseudocode Method Updateprogres dan Savenilai

No.	Pseudocode Method Updateprogres dan Savenilai
1	START
2	GET \$hafalans = hafalans.map(function (\$entry) use (\$id,\$progres)
3	IF (\$entry[id] !== \$id)
4	RETURN \$entry
5	SET \$entry[halaman] = \$progres
6	RETURN \$entry
7	GET \$hafalan = \$hafalan.map(function (\$entry) use (\$id,\$progres)
8	IF (\$entry[id] !== \$id)
9	RETURN \$entry
10	SET \$entry[nilai][nilai] = \$nilai
11	SET \$predikat = Rosib
12	IF (\$nilai > 80)
13	SET \$predikat = Muntaz
14	IF (\$nilai >= 70)
15	SET \$predikat = Jayyid Jiddan
16	IF (\$nilai >= 60)
17	SET \$predikat = Jayyid
18	SET \$entry[nilai][predikat] = \$predikat
19	RETURN \$entry
20	SET \$change = \$hafalan.first(function (\$item) use (\$id)
21	RETURN \$item[id] === \$id
22	RETURN \$change

5.1.3.2 Perancangan Komponen Method setupUpdateOperation

Pada Tabel 5.2 memaparkan algoritme *method setupUpdateOperation* yang digunakan untuk menambahkan transaksi pada basis data tabel *OrderDetails*. Dari algoritme ini, dilakukan inisialisasi variabel yang relevan. Kemudian dilakukan perulangan untuk mengambil nilai kode transaksi terakhir. Saat proses penyusunan nilai kode transaksi selesai, dilakukan perulangan untuk memasukkan seluruh *item* pada transaksi pada tabel *OrderDetails*. Setelah sukses, pesan transaksi berhasil dikembalikan dalam representasi *Json*. Nama *class MataPelajaranCrudController* dan nama *method setupUpdateOperation*.

Tabel 5.2 Pseudocode Method SetupUpdateOperation

No.	Pseudocode Method SetupUpdateOperation
1	START
2	CALL setupUpdateOperation()
3	CALL setupCreateOperation()
4	CALL function edit (\$id)
5	GET hasAccessorFail (update)
6	SET \$id = getCurrentEntryId() ?? \$id
7	GET setOperationSetting(fields, getUpdateFields())
8	SET \$data[entry] = getEntry(\$id)
9	SET \$data[crud] = \$crud
10	SET \$data[saveAction] = getSaveAction()
11	SET \$data[title] = getTitle() ?? trans(crud.edit).entityname
12	SET \$data[id] = \$id
13	RETURN view(getEditView() (\$data))
14	CALL function update()
15	GET hasAccessorFail (update)
16	SET \$request = validateRequest()
17	SET \$item = update(\$request.get(getKeyName()))
18	SET \$data[entry] = \$entry + \$item
19	ALERT success
20	GET setSaveAction()
21	RETURN performSaveAction(\$item,getKey())
22	END



5.1.3.3 Perancangan Komponen *Method setupListOperation*

Pada Tabel 5.4 memaparkan algoritme *method setupListOperation* yang digunakan untuk menambahkan transaksi pada basis data tabel *OrderDetails*. Dan algoritme ini, dilakukan inialisasi variabel yang relevan. Kemudian dilakukan perulangan untuk mengambil nilai kode transaksi terakhir. Saat proses penyusunan nilai kode transaksi selesai, dilakukan perulangan untuk memasukkan seluruh *item* pada transaksi pada tabel *OrderDetails*. Setelah sukses, pesan transaksi berhasil dikembalikan dalam representasi *Json*. Nama *class PenggunaController* dan nama *method setupListOperation*.

Tabel 5.3 Pseudocode Method SetupListOperation

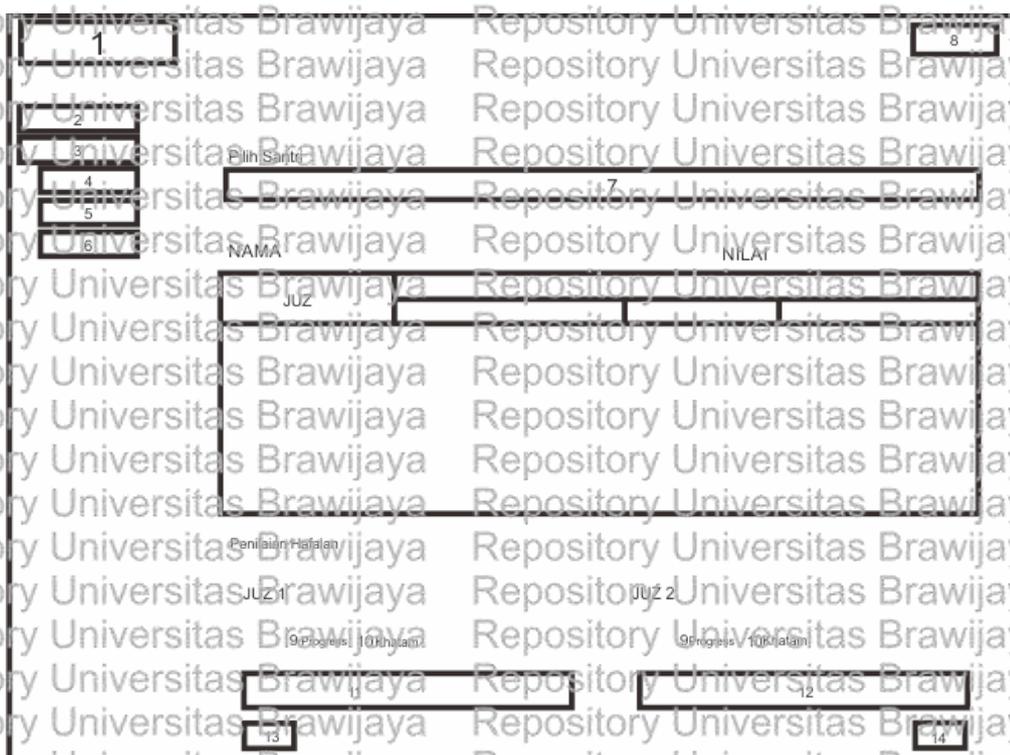
No.	Pseudocode Method SetupListOperation
1	START
2	CALL function setupListOperation()
3	PARENT setupListOperation()
4	GET \$allowAccess(show)
5	GET \$removeColumns([roles, permissions])
6	Get \$addColumn([
7	name, jabatan,
8	label, jabatan,
9	type.closure,
10	function function (Pengguna, \$entry)
11	RETURN \$entry.hasRole(murobbi) ? MUROBBI : USTAD
12	END

5.1.4 Perancangan Antarmuka

Ketika masuk dalam tahap perancangan antarmuka akan ditampilkan beberapa contoh antarmuka dari sistem yaitu, halaman nilai hafalan, halaman data mata pelajaran, dan halaman data pengguna. Pada tahap ini semua hasil yang didapat akan menjadi rujukan ketika melakukan implementasi sistem.

5.1.4.1 Perancangan Antarmuka Halaman Nilai Hafalan

Pada Gambar 5.8 ditampilkan perancangan antarmuka halaman nilai hafalan, dan Tabel 5.4 menjelaskan objek yang ada pada antarmuka halaman nilai hafalan.



Gambar 5.8 Perancangan Antarmuka Halaman Nilai Hafalan

Tabel 5.4 Penjelasan Perancangan Antarmuka Halaman Nilai Hafalan

No.	Nama Objek	Tipe	Keterangan
1	SPS	Header	Header untuk menunjukkan sistem pengguna
2	Dashboard	Button	Button untuk halaman dashboard
3	Penilaian	Button	Button untuk halaman penilaian
4	Hafalan	Button	Button untuk mengalihkan ke halaman hafalan
5	Leadership	Label	Label yang menampilkan halaman leadership
6	Pelanggaran	Label	Label yang menampilkan halaman pelanggaran
7	Jumlah Santri	Dropdown	Dropdown untuk memilih data santri
8	Akun	Button	Button untuk memunculkan pengguna yang tengah mengakses sistem
9	Progress	Radio Button	Radio Button untuk memilih progres



Tabel 5.4 Penjelasan Perancangan Antarmuka Halaman Nilai Hafalan (Lanjutan)

No.	Nama Objek	Tipe	Keterangan
10	<i>Khatam</i>	<i>Radio Button</i>	<i>Radio Button</i> untuk memilih <i>khatam</i>
11	Halaman	<i>Textfield</i>	<i>Textfield</i> untuk memasukkan halaman progres
12	Nilai	<i>Textfield</i>	<i>Textfield</i> untuk memasukkan nilai hafalan
13	Simpan	<i>Button</i>	<i>Button</i> untuk menyimpan progres
14	Simpan	<i>Button</i>	<i>Button</i> untuk melakukan penyimpanan nilai hafalan

5.1.4.2 Perancangan Antarmuka Halaman Data Mata Pelajaran

Pada Gambar 5.9 ditampilkan perancangan antarmuka halaman nilai hafalan, dan Tabel 5.6 menjelaskan objek yang ada pada antarmuka halaman nilai hafalan.



Gambar 5.9 Perancangan Antarmuka Halaman Data Mata Pelajaran

Tabel 5.5 Penjelasan Perancangan Antarmuka Halaman Data Mata Pelajaran

No.	Nama Objek	Tipe	Keterangan
1	SPS	<i>Header</i>	<i>Header</i> untuk menunjukkan sistem pengguna
2	<i>Dashboard</i>	<i>Button</i>	<i>Button</i> untuk halaman <i>dashboard</i>
3	Administrasi	<i>Button</i>	<i>Button</i> untuk menampilkan menu administrasi



Tabel 5.6 Penjelasan Perancangan Antarmuka Halaman Data Mata Pelajaran (Lanjutan)

No.	Nama Objek	Tipe	Keterangan
1	SPS	<i>Header</i>	Header untuk menampilkan sistem pengguna
2	<i>Dashboard</i>	<i>Button</i>	<i>Button</i> untuk halaman <i>dashboard</i>
3	Administrasi	<i>Button</i>	<i>Button</i> untuk menampilkan menu administrasi
4	Mata Pelajaran	<i>Button</i>	<i>Button</i> untuk mengalihkan ke halaman data mata pelajaran
5	Buat Mata Pelajaran	<i>Button</i>	<i>Button</i> untuk membuat mata pelajaran baru
6	Akun	<i>Button</i>	<i>Button</i> untuk menampilkan pengguna yang sedang <i>login</i> di dalam sistem
7	Search	<i>Textfield</i>	<i>Textfield</i> untuk mencari data mata pelajaran
8	<i>Preview</i>	<i>Button</i>	<i>Button</i> untuk menampilkan detail mata pelajaran
9	<i>Edit</i>	<i>Button</i>	<i>Button</i> untuk memperbarui data mata pelajaran
10	Delete	<i>Button</i>	<i>Button</i> untuk melakukan penghapusan pada data mata pelajaran
11	Page	<i>Button</i>	<i>Button</i> untuk memilih halaman dari data santri

5.1.4.3 Perancangan Antarmuka Halaman Data Pengguna

Pada Gambar 5.10 ditampilkan perancangan antarmuka halaman nilai hafalan, dan Tabel 5.7 menjelaskan objek yang ada pada antarmuka halaman nilai hafalan.



Gambar 5.10 Perancangan Antarmuka Halaman Data Pengguna

Tabel 5.6 Penjelasan Perancangan Antarmuka Halaman Data Pengguna

No.	Nama Objek	Tipe	Keterangan
1	SPS	Header	Header untuk memunculkan sistem pengguna
2	Dashboard	Button	Button untuk halaman dashboard
3	Administrasi	Button	Button untuk menampilkan menu administrasi
4	Pengguna sistem	Button	Button untuk menampilkan jenis pengguna sistem
5	Pengguna	Button	Button untuk menampilkan halaman data pengguna
6	Buat Pengguna	Button	Button untuk membuat pengguna baru
7	Akun	Button	Button untuk menampilkan pengguna yang sedang login di dalam sistem
8	Search	Textfield	Textfield untuk mencari data mata pelajaran
9	Edit	Button	Button untuk memperbarui data mata pelajaran
10	Delete	Button	Button untuk melakukan penghapusan pada data mata pelajaran
11	Page	Button	Button untuk memilih halaman dari data santri



5.2 Implementasi Sistem

Setelah selesai melakukan perancangan maka masuklah pada tahap implementasi. Hasil dari perancangan sistem dan analisis kebutuhan sistem akan digunakan dalam proses implementasi. Implementasi sistem dengan kode program yang memiliki dasar dari perancangan algoritme, diikuti dengan implementasi data dan implementasi antarmuka yang sesuai dengan perancangan data serta perancangan antarmuka. Penjelasan mengenai implementasi sistem lebih lanjut dimulai dengan spesifikasi sistem, kemudian implementasi data, implementasi kode program, dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

Bagian ini memberikan penjelasan mengenai spesifikasi perangkat lunak dan perangkat keras yang menunjang keberlangsungan penelitian ini. Spesifikasi perangkat keras yang digunakan ditampilkan pada Tabel 5.7, dan spesifikasi perangkat lunak ditampilkan pada Tabel 5.8.

Tabel 5.7 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>Processor</i>	AMD E1-6010 with AMD Radeon R2 Graphics 1.35 GHz
<i>Hard Disk Drive</i>	465 GB
<i>Main Memory (RAM)</i>	4.00 GB

Tabel 5.8 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 10 64-bit
<i>Text Editor</i>	JetBrains PhpStorm 2019.3 x64
<i>Programming Language</i>	PHP 7.3.12
<i>Framework</i>	Laravel 6.5.2
<i>Editor Dokumentasi</i>	Draw.io
<i>Editor Perancangan</i>	Visual Paradigm Enterprise version 16.0 Microsoft Word 2010



5.2.2 Implementasi Basis Data

Perancangan data menggunakan *Physical Data Model* (PDM) untuk membuat model. relasi antar entitas data sistem ditampilkan di dalam PDM yang dilakukan pada tahap perancangan data. Gambar 5.11 adalah PDM sistem penilaian santri.



Gambar 5.11 Perancangan *Physical Data Model*

Dalam Gambar 5.11 ditampilkan 9 entitas yang ada pada sistem ini yaitu pengguna, ustad, *murobbi*, santri, kelas, santrikelas, pelanggaran, wali santri, *santrimurobbi*. Entitas-entitas tersebut dan diakhiri dengan huruf “s” karena database pada laravel mengidentifikasi semua entitas pada database sebagai hal yang jamak. Semua entitas memiliki relasi satu sama lain untuk memberikan kemudahan melakukan tukar menukar data secara dinamis.

5.2.3 Implementasi Kode Program

Bagian ini akan memaparkan mengenai implementasi kode program. Implementasi akan menggunakan algoritme yang ada pada perancangan komponen di tahap sebelumnya sebagai dasar kode program. Pada penelitian ini PHP dan MySQL digunakan untuk menjadi bahasa pemrograman



5.2.3.1 Implementasi Kode Program *Method updateprogres()* dan *savenilai()* pada NilaiHafalan

Method updateprogres() dan *savenilai()* pada NilaiHafalan berfungsi untuk inisiasi pembaruan nilai hafalan dari suatu juz dengan mengisi di bagian progress jika hafalan pada juz tersebut belum selesai dan *khatam* jika hafalan pada juz tersebut sudah selesai dihafal dan diberikan penilaian pada kualitas hafalan pada juz tersebut.

Tabel 5.9 Source Code Method Updateprogres() Pada NilaiHafalan

```
Source code method updateprogres() pada NilaiHafalan
1 public function updateprogres(string $id, int $progres){
```

Tabel 5.9 Source Code Method Updateprogres() Pada NilaiHafalan (Lanjutan)

```
Source code method updateprogres() pada NilaiHafalan
2     $this->hafalans = $this->hafalans->map(function ($entry) use
3     ($id, $progres) {
4         if ($entry['id'] !== $id) {
5             return $entry;
6         }
7         $entry['halaman'] = $progres;
8         return $entry;
9     });
```

Tabel 5.10 Source Code Method Savenilai() Pada NilaiHafalan

```
Source code method savenilai() pada NilaiHafalan
1 public function savenilai(string $id, int $nilai){
2     $this->hafalans = $this->hafalans->map(function ($entry) use
3     ($id, $nilai) {
4         if ($entry['id'] !== $id) {
5             return $entry;
6         }
7         $entry['nilai']['nilai'] = $nilai;
8         $predikat = 'Rosib';
9         if ($nilai > 80) {
10            $predikat = 'Mumtaz';
11        }
12        elseif ($nilai >= 70) {
13            $predikat = 'Jayid Jiddan';
14        }
15        elseif ($nilai >= 60) {
16            $predikat = 'Jayid';
17        }
18        $entry['nilai']['predikat'] = $predikat;
19        return $entry;
20    });
21    $change = $this->hafalans->first(function ($item) use ($id)
22    {
23        return $item['id'] === $id;
24    });
25    return $change;
26 }
```



5.2.3.2 Implementasi Kode Program Method `setupUpdateOperation()` pada `MataPelajaranCrudController`

Method `setupUpdateOperation()` pada `MataPelajaranCrudController` berfungsi untuk memanggil method `setupCreateOperation` yang memakai traits `UpdateOperation` untuk memanggil method `edit` untuk menampilkan form edit untuk input perubahan data pada mata pelajaran. Selanjutnya, method `update` dipanggil setelah selesai pengisian data pada form edit. Method `update` berfungsi untuk penyimpanan data pada database setelah pengisian data pada form edit.

Tabel 5.11 Source Code Method `SetupUpdateOperation()` Pada `MataPelajaranCrudController`

Source code	method <code>setupUpdateOperation()</code> pada <code>MataPelajaranCrudController</code>
1	<code>protected function setupUpdateOperation()</code>
2	<code>{</code> <code> \$this->setupCreateOperation();</code> <code>}</code>

Tabel 5.12 Source Code Method `Edit()` Pada `UpdateOperation`

Source code	method <code>edit()</code> pada <code>UpdateOperation</code>
1	<code>public function edit(\$id){</code>
2	<code> \$this->crud->hasAccessOrFail('update');</code>
3	<code> \$id = \$this->crud->getCurrentEntryId() ?? \$id;</code>
4	<code> \$this->crud->setOperationSetting('fields', \$this->crud->getUpdateFields());</code>
5	<code> \$this->data['entry'] = \$this->crud->getEntry(\$id);</code>
6	<code> \$this->data['crud'] = \$this->crud;</code>
7	<code> \$this->data['saveAction'] = \$this->crud->getSaveAction();</code>
8	<code> \$this->data['title'] = \$this->crud->getTitle() ?? trans('backpack::crud.edit', \$this->crud->entity_name,</code>
9	<code> \$this->data['id'] = \$id;</code>
10	<code> return view(\$this->crud->getEditView(), \$this->data);</code>
11	<code>}</code>

Tabel 5.13 Source Code Method `Update()` Pada `UpdateOperation`

Source code	method <code>update()</code> pada <code>UpdateOperation</code>
1	<code>public function update()</code>
2	<code>{</code>
3	<code> \$this->crud->hasAccessOrFail('update');</code>
4	<code> \$request = \$this->crud->validateRequest();</code> <code> \$item = \$this->crud->update(\$request->get(\$this->crud->getModel()->getKeyName()), \$this->crud->getStrippedSaveRequest());</code>
5	<code> \$this->data['entry'] = \$this->crud->entry = \$item;</code>
6	<code> Alert::success(trans('backpack::crud.update_success'))->flash();</code>
7	<code> \$this->crud->setSaveAction();</code>
8	<code> return \$this->crud->performSaveAction(\$item->getKey());</code>



5.2.3.3 Implementasi Kode Program *Method* *setupListOperation()* pada *PenggunaController*

Method *setupListOperation()* pada *PenggunaController* berfungsi untuk menginisiasi data yang akan ditampilkan pada halaman pengguna. Untuk pengambilan data pengguna kelas *PenggunaController* extend kelas *UserCrudController* untuk menggunakan *trait* *ListOperation* dan memanggil *method* *index* yang ada di *trait* tersebut. *Method* *index* berfungsi untuk memanggil data pengguna untuk dikirimkan datanya ke *view*.

Tabel 5.14 Source Code *Method* *SetupListOperation()* Pada *PenggunaController*

Source code <i>method</i> <i>setupListOperation()</i> pada <i>PenggunaController</i>	
1	<code>public function setupListOperation(){</code>
2	<code> parents::setupListOperation();</code>
3	<code> \$this->crud->allowAccess('show');</code>
4	<code> \$this->crud->removeColumns(['roles', 'permissions']);</code>
5	<code> \$this->crud->addColumn([</code>
6	<code> 'name' => 'jabatan',</code>
7	<code> 'label' => 'Jabatan',</code>
8	<code> 'type' => 'closure'</code>
9	<code> 'function' => function(Pengguna \$entry){</code>
10	<code> return \$entry->hasRole('murobbi') ? 'MUROBBI' :</code>
	<code> 'USTAD'; }]);</code>

5.2.4 Implementasi Antarmuka

Implementasi antarmuka berdasar pada perancangan antarmuka pada sub bab 5.1.4. tanap ini memaparkan tiga hasil implementasi antarmuka yang terdapat pada sub bab 5.2.4.1 hingga sub bab 5.2.4.3.

5.2.4.1 Implementasi Antarmuka Halaman Nilai Hafalan

Pada Gambar 5.11 digambarkan antarmuka halaman nilai hafalan yang telah diimplementasikan pada menu hafalan. Halaman ini dapat diisi dengan nilai hafalan yang didapat santri ketika proses pembelajaran berlangsung.



Gambar 5.12 Implementasi Antarmuka Menambah Nilai Hafalan

5.2.4.2 Implementasi Antarmuka Halaman Data Mata Pelajaran

Dalam Gambar 5.12 adalah antarmuka yang telah diimplementasikan dari halaman mata pelajaran. Halaman ini berguna untuk melakukan penambahan mata pelajaran baru yang ada pada kurikulum Al-Izzan.



Gambar 5.13 Implementasi Antarmuka Mengubah Data Mata Pelajaran

5.2.4.3 Implementasi Antarmuka Melihat Data Pengguna

Dalam Gambar 5.13 adalah antarmuka yang telah diimplementasikan dari halaman data pengguna. Halaman ini memperlihatkan pengguna yang mempunyai hak akses sistem selain admin.



Gambar 5.14 Implementasi Antarmuka Melihat Data Pengguna

BAB 6 PENGUJIAN

Setelah implementasi selesai dilakukan, masuklah pada tahapan selanjutnya yaitu pengujian. Tahap ini melakukan pengujian dari hasil yang didapat dari implementasi sistem dan memeriksa kesesuaian dengan perancangan dan analisis kebutuhan, apakah sudah memenuhi ekpektasi yang diharapkan. pengujian unit, integrasi, validasi, dan usability adalah pengujian yang akan dilakukan pada penelitian ini.

6.1 Pengujian Unit

6.1.1 Pengujian Unit NilaiHafalan

Tabel 6.1 Pseudocode NilaiHafalan

No.	Pseudocode	Method	Updateprogres dan Savenilai
1	START		1
2	GET \$hafalans = hafalans.map(function(\$entry) use (\$id,\$progres)		2
3	IF (\$entry[id] !== \$id)	}	3
4	RETURN \$entry		
5	SET \$entry[halaman] = \$progres		
6	RETURN \$entry		
7	GET \$hafalan = \$hafalan.map(function (\$entry) use (\$id,\$progres)		4
8	IF (\$entry[id] !== \$id)		5
9	RETURN \$entry		
10	SET \$entry[nilai][nilai] = \$nilai	}	6
11	SET \$predikat = Rosib		
12	IF (\$nilai > 80)	}	7
13	SET \$predikat = Muntaz		
14	IF (\$nilai >= 70)	}	8
15	SET \$predikat = Jayyid Uldan		
16	IF (\$nilai >= 60)	}	9
17	SET \$predikat = Jayyid		
18	SET \$entry[nilai][predikat] = \$predikat		10
19	RETURN \$entry		11
20	SET \$change = \$hafalan.first(function (\$item) use (\$id)		12
21	RETURN \$item[id] === \$id		
22	RETURN \$change		



Flow Graph



Gambar 6.1 Flow Graph pada Nilai Hafalan

Cyclomatic Complexity

Pada Gambar 6.1 akan didapatkan jumlah jalur independen dengan menggunakan kalkulasi cyclomatic complexity. Hasil dari kalkulasi cyclomatic complexity flow graph method update progres dan save nilai yang terdapat pada kelas Nilai Hafalan dipaparkan oleh perhitungan berikut ini.

1. $V(G) = \text{jumlah region} = 4$
2. $V(G) = 15 \text{ edge} - 13 \text{ node} + 2 = 4$
3. $V(G) = 3 \text{ predicate node} + 1 = 4$

Independent Path

1. Jalur 1 = 1 – 2 – 3 – 11 – 12 – 13
2. Jalur 2 = 1 – 2 – 3 – 4 – 5 – 10 – 11 – 12 – 13
3. Jalur 3 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 10 – 11 – 12 – 13
4. Jalur 4 = 1 – 2 – 3 – 4 – 5 – 6 – 8 – 9 – 10 – 11 – 12 – 13

Dari jalur independen yang didapatkan, kemudian akan dijadikan dasar dari pembuatan kasus uji. Hasil pengujian menggunakan kasus uji dipaparkan melalui Tabel 6.2.

Tabel 6.2 Hasil Pengujian Unit *Method updateprogres* dan *savenilai*

Jalur	Prosedur Uji	Hasil Pengujian yang diharapkan	Hasil Pengujian	Status
1	<i>Method updateprogres()</i> dijalankan, <i>looping</i> data hafalan dengan menggunakan variabel \$id dan \$progres, lalu melakukan <i>entry data progress</i> hafalan dan menyimpan di dalam file json.	Menjalankan <i>method updateprogres()</i> dan menyimpan data <i>progress</i> hafalan santri.	Menjalankan <i>method updateprogres()</i> dan menyimpan data <i>progress</i> hafalan santri.	Valid
2	<i>Method savenilai()</i> dijalankan dengan <i>looping</i> data Shafalan, kemudian mengambil data dan menyimpan nilai dengan inputan, selanjutnya melakukan penentuan predikat nilai dengan nilai default \$predikat = "Rosib", kemudian menyimpan ke dalam file json.	Menjalankan <i>method create()</i> dan melakukan verifikasi pengguna yang sedang <i>login</i> dan menampilkan pesan error.	Menjalankan <i>method create()</i> dan melakukan verifikasi pengguna yang sedang <i>login</i> dan menampilkan pesan error.	Valid
3	<i>Method savenilai()</i> dijalankan dengan <i>looping</i> data Shafalan, kemudian mengambil data dan menyimpan nilai dengan inputan, selanjutnya melakukan penentuan predikat nilai dengan nilai \$predikat = "Mumta'az" jika nilai diatas 80, jika nilai lebih sama dengan 70, maka nilai \$predikat =	Menjalankan <i>method savenilai()</i> dan menyimpan nilai hafalan santri dengan .	Menjalankan <i>method create()</i> dan menampilkan data santri pada <i>class view draft</i> nilai santri.	Valid



Tabel 6.2 Hasil Pengujian Unit Method updateprogres dan savenilai (Lanjutan)

	<p>“Jayyid” kemudian menyimpan ke dalam file json.</p>		
4	<p>Method <i>savenilai()</i> dijalankan dengan looping data Shafalan, kemudian mengambil data dan menyimpan nilai dengan inputan, selanjutnya melakukan penentuan predikat nilai dengan nilai \$predikat = “Mumtaz” jika nilai diatas 80, jika nilai lebih sama dengan 60, maka nilai \$predikat = “Jayyid” kemudian menyimpan ke dalam file json.</p>		

6.1.2 Pengujian Unit Function edit dan update

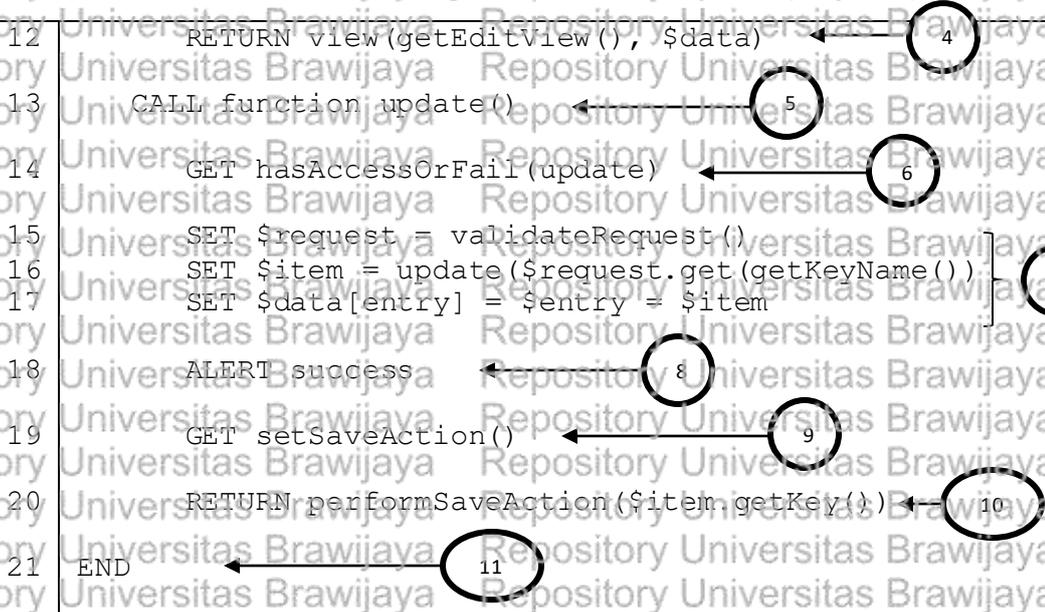
Pengujian unit *method setupUpdateOperation* yang terdapat pada kelas *MataPelajaranCrudController* akan dijelaskan dengan algorithme yang ditunjukkan pada Tabel 6.3.

Tabel 6.3 Pseudocode edit dan update

No.	Pseudocode Method SetupUpdateOperation
1	START
2	CALL function edit (\$sid)
3	GET hasAccessOrFail (update)
4	SET \$sid = getCurrentEntryId() ?? \$sid
5	GET setOperationSetting (fields,
6	getUpdateFields())
7	SET \$data[entry] = getEntry(\$sid)
8	SET \$data[crud] = \$crud
9	SET \$data[saveAction] = getSaveAction()
10	SET \$data[title] = getTitle()
11	trans (crud.edit).entityname SET \$data[id] = \$sid



Tabel 6.3 Pseudocode Algoritme edit dan update (lanjutan)





1. Basic Path Testing
Flow Graph



Gambar 6.2 Flow Graph Function Edit dan Update

Cyclomatic Complexity

Pada Gambar 6.2 akan didapatkan jumlah jalur independen dengan menggunakan kalkulasi cyclomatic complexity. Hasil dari kalkulasi cyclomatic complexity flow graph method edit dan update pada Mata Pelajaran Crud Controller dipaparkan oleh perhitungan berikut ini

1. $V(G) = \text{jumlah region} = 1$
2. $V(G) = 10 \text{ edge} - 11 \text{ node} + 2 = 1$
3. $V(G) = 0 \text{ predicate node} + 1 = 1$

Independent Path

- Jalur 1: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11

Tabel 6.4 Hasil Pengujian Unit method edit dan update

Jalur	Prosedur Uji	Hasil Pengujian yang diharapkan	Hasil Pengujian	Status
1	Memulai dengan memanggil function	Menjalankan function edit	Menjalankan function edit dan	Valid



Tabel 6.4 Hasil Pengujian Unit *method edit* dan *update* (Lanjutan)

Jalur	Prosedur Uji	Hasil Pengujian yang diharapkan	Hasil Pengujian	Status
	<p><i>edit</i> dengan parameter <i>id</i> kemudian menginisialisasi data dan mengembalikan data ke <i>view</i> dengan parameter <i>getEditView()</i> dan data yang telah diinisialisasi tadi, kemudian masuk ke <i>method update</i> untuk pembaruan data dengan meminta <i>request</i> data dan memanggil <i>method setSaveAction()</i> dan mengembalikan data ke <i>method performSaveAction()</i></p>	<p>dan <i>update</i> dan berhasil mengubah data mata pelajaran</p>	<p><i>update</i> dan berhasil mengubah data mata pelajaran</p>	

6.1.3 Pengujian Unit *Method setupListOperation*

Pseudocode

Tabel 6.5 *Pseudocode setupListOperation*

No.	Pseudocode Method SetupListOperation
1	START
2	CALL function <i>setupListOperation()</i>
3	PARENT <i>ListOperation()</i>
4	GET <i>\$allowAccess(show)</i>
5	GET <i>\$removeColumns([roles, permissions])</i>
6	Get <i>\$addColumn([</i>
7	<i>name.jabatan,</i>
8	<i>label.jabatan,</i>
9	<i>type.closure,</i>
10	<i>function.function(Pengguna \$entry)</i>
11	RETURN <i>\$entry.hasKole (murobbi) ? MUROBBI :</i>
12	USTAD
	END



Flow Graph



Gambar 6.3 Flow Graph Method setupListOperation

Cyclomatic Complexity

Pada Gambar 6.3 akan didapatkan jumlah jalur independen dengan menggunakan kalkulasi cyclomatic complexity. Hasil dari kalkulasi cyclomatic complexity flow graph method setupListOperation pada PenggunaController dipaparkan oleh perhitungan berikut ini.

- $V(G) = \text{jumlah region} = 1$
- $V(G) = 5 \text{ edge} - 6 \text{ node} + 2 = 1$
- $V(G) = 0 \text{ predicate node} + 1 = 1$

Independent Path

- Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6

Tabel 6.6 Hasil Pengujian Unit Method setupListOperation

Jalur	Prosedur Uji	Hasil Pengujian yang diharapkan	Hasil Pengujian	Status
1	Method setupListOperation () dijalankan dengan menggunakan fungsi dari parent class	Menjalankan method setupListOperation() dan menampilkan data pengguna dengan jabatan dari tiap pengguna.	Menjalankan method setupListOperation () dan menampilkan data pengguna dengan jabatan dari tiap	Valid

Tabel 6.6 Hasil Pengujian Unit *Method setupListOperation* (Lanjutan)

Jalur	Prosedur Uji	Hasil Pengujian yang diharapkan	Hasil Pengujian	Status
	<i>ListOperation()</i> memanggil semua data pengguna dengan menghilangkan kolom <i>roles</i> dan <i>permissions</i> , kemudian menambahkan kolom <i>jabatan</i> untuk <i>ustaz</i> atau <i>murobbi</i>		pengguna.	

6.2 Pengujian Integrasi

Pseudocode

Tabel 6.7 *Pseudocode* Algoritme *update* pada *DraftController*

Pseudocode method *update* pada *DraftController*

1	START	← 1
2	CALL function <i>update()</i>	← 2
3	GET <i>\$helper</i> = <i>\$getIndexData()</i> [<i>helper</i>]	← 3
4	SET <i>\$methodname</i> = <i>update_</i> <i>\$prefix</i>	
5	SET <i>\$helper</i> . <i>\$methodname</i>	
6	<i>\$id</i> , <i>collect(</i>	
7	<i>\request().input(data</i>	← 4
8	<i>)</i>	
9	RETURN <i>\response([message.updated])</i>	← 5
10	CALL function <i>getIndexData()</i>	← 6
11	CALL function <i>setIndexData()</i>	← 7
12	GET <i>\$ustad</i> = <i>auth().User().ustad</i>	← 8
13	IF (<i>request.has(kode_kelas, kode_pelajaran)</i>)	← 9
14	<i>\$matapelajaran</i> = <i>MataPelajaran::findByCode(</i>	
	<i>request.input(kode_pelajaran)</i>	
	<i>\$kelas</i> = <i>Kelas::findByCode(</i>	
	<i>Request.input(kode_kelas)</i>	← 10



Cyclomatic Complexity

Pada Gambar 6.3 akan didapatkan jumlah jalur independen dengan menggunakan kalkulasi *cyclomatic complexity*. Hasil dari kalkulasi *cyclomatic complexity flow graph method update* pada *DraftController* dipaparkan oleh perhitungan berikut ini.

- $V(G) = \text{jumlah region} = 4$
- $V(G) = 16 \text{ edge} - 14 \text{ node} + 2 = 4$
- $V(G) = 3 \text{ predicate node} + 1 = 4$

Independent Path

- Jalur 1 : 1 – 2 – 3 – 4 – 5 – 14
- Jalur 2 : 1 – 2 – 3 – 6 – 7 – 8 – 9 – 14
- Jalur 3 : 1 – 2 – 3 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14
- Jalur 4 : 1 – 2 – 3 – 6 – 7 – 8 – 9 – 10 – 11 – 13 – 14

Tabel 6.8 Hasil Pengujian Unit *Method update*

Jalur	Prosedur Uji	Hasil Pengujian yang diharapkan	Hasil Pengujian	Status
1	<i>Method update()</i> dijalankan dan memanggil <i>method update_prefix</i> yang diambil pada <i>method setIndexData()</i>	Berhasil memanggil <i>method update_\$prefix</i>	Berhasil memanggil <i>method update_\$prefix</i>	Valid
2	<i>Method update()</i> dijalankan kemudian memanggil <i>method setIndexData</i> , jika <i>request</i> tidak memiliki kode kelas dan kode pelajaran maka akan membatalkan proses update	Membatalkan proses update dikarenakan <i>request</i> tidak memiliki kode kelas dan kode pelajaran	Membatalkan proses update dikarenakan <i>request</i> tidak memiliki kode kelas dan kode pelajaran	Valid
3	<i>Method update()</i> dijalankan kemudian memanggil	Membuat objek pada <i>UstadDraftHelper</i> dengan parameter	Membuat objek pada <i>UstadDraftHelper</i> dengan parameter	Valid

Tabel 6.8 Hasil Pengujian Unit *Method update* (Lanjutan)

3	<i>method setIndexData</i> , jika <i>request</i> memiliki kode kelas dan kode pelajaran maka dilakukan perulangan untuk memeriksa apakah data yang dimasukkan adalah uas dan uts	ustad, kelas, mata pelajaran, prefix dan <i>current_semester()</i>	ustad, kelas, mata pelajaran, dan <i>current_semester()</i>
4	<i>Method update()</i> dijalankan kemudian memanggil <i>method setIndexData</i> , jika <i>request</i> memiliki kode kelas dan kode pelajaran maka dilakukan perulangan untuk memeriksa apakah data yang dimasukkan adalah selain uas dan uts	Membuat objek pada <i>UstadDraftHelper</i> dengan parameter ustad, kelas, mata pelajaran dan prefix	Membuat objek Valid pada <i>UstadDraftHelper</i> dengan parameter ustad, kelas, mata pelajaran dan prefix

6.3 Pengujian Validasi

Pengujian validasi adalah tahap ketiga dalam proses pengujian. Pengujian validasi termasuk pengujian *blackbox*, pengujian ini bertujuan untuk memeriksa kelengkapan sistem dalam memenuhi kebutuhan yang diperlukan oleh sistem dan memiliki tujuan dalam memeriksa tingkat kesesuaian dari fungsi yang sudah dibangun bersama fungsi yang diharapkan pada tahap perancangan. Pengujian *blackbox* mengamati hasil dari implementasi dan memperoleh kondisi dengan masukan yang diharapkan dari semua fungsional.

6.3.1 Pengujian Validasi Login

Tabel 6.9 Kasus Uji Berhasil Melakukan Login

Nama Kasus Uji	Kasus uji berhasil melakukan <i>login</i> .
Prosedur	1. Melakukan pengisian kolom <i>e-mail</i> . 2. Melakukan pengisian kolom <i>password</i> . 3. Melakukan penekanan pada tombol <i>login</i> .



Tabel 6.9 Kasus Uji Berhasil Melakukan Login (Lanjutan)

Hasil yang diharapkan	Berhasil masuk ke halaman <i>dashboard user</i> .
Hasil	Berhasil masuk ke halaman <i>dashboard user</i> .
Status	Valid

Tabel 6.10 kasus Uji Tidak berhasil Melakukan Login dengan Mengosongkan Kolom Password

Nama Kasus Uji	Kasus uji tidak berhasil melakukan <i>login</i> .
Prosedur	1. Melakukan pengisian kolom <i>e-mail</i> . 2. Mengosongkan kolom <i>password</i> . 3. Tekan tombol <i>login</i> .
Hasil yang diharapkan	memperlihatkan pesan " <i>please fill out this field</i> " pada kolom <i>password</i> .
Hasil	memperlihatkan pesan " <i>please fill out this field</i> " pada kolom <i>password</i> .
Status	Valid

Tabel 6.11 Kasus Uji Tidak berhasil Melakukan Login dengan Tidak Melakukan Pengisian Kolom E-mail

Nama Kasus Uji	Kasus uji tidak berhasil melakukan <i>login</i> .
Prosedur	1. tidak melakukan pengisian kolom <i>e-mail</i> . 2. Melakukan pengisian kolom <i>password</i> . 3. Tekan tombol <i>login</i> .
Hasil yang diharapkan	memperlihatkan pesan " <i>please fill out this field</i> " pada kolom <i>e-mail</i> .
Hasil	memperlihatkan pesan " <i>please fill out this field</i> " pada kolom <i>e-mail</i> .
Status	Valid.

Tabel 6.12 Kasus Uji Tidak berhasil Melakukan Login dengan Masukan yang Tidak Sesuai Pada Kolom E-mail

Nama Kasus Uji	Kasus uji tidak berhasil melakukan <i>login</i> .
Prosedur	1. Melakukan pengisian kolom <i>e-mail</i> dengan <i>e-mail</i> yang tidak sesuai. 2. Melakukan pengisian kolom <i>password</i> . 3. Tekan tombol <i>login</i> .



Tabel 6.12 Kasus Uji Tidak berhasil Melakukan Login dengan Masukan yang Tidak Sesuai Pada Kolom E-mail (Lanjutan)

Hasil yang diharapkan	memperlihatkan pesan " <i>These credentials do not match our record</i> ".
Hasil	Menampilkan pesan " <i>These credentials do not match our record</i> ".
Status	Valid

Tabel 6.13 Kasus Uji Tidak berhasil Melakukan Login dengan Masukan yang Tidak Sesuai Pada Kolom Password

Nama Kasus Uji	Kasus uji tidak berhasil melakukan login.
Prosedur	<ol style="list-style-type: none"> 1. Melakukan pengisian kolom <i>e-mail</i>. 2. Melakukan pengisian pada kolom <i>password</i> dengan isian yang salah. 3. Tekan tombol <i>login</i>.
Hasil yang diharapkan	Memperlihatkan pesan " <i>These credentials do not match our record</i> ".
Hasil	Memperlihatkan pesan " <i>These credentials do not match our record</i> ".
Status	Valid

6.3.2 Pengujian Validasi Logout

Tabel 6.14 Kasus Uji Berhasil Logout

Nama Kasus Uji	Kasus uji berhasil <i>logout</i> .
Prosedur	<ol style="list-style-type: none"> 1. Menekan tombol profil pada bagian kanan atas halaman. 2. Tekan tombol <i>logout</i>.
Hasil yang diharapkan	Memperlihatkan halaman utama.
Hasil	Memperlihatkan halaman utama.
Status	Valid.

6.3.3 Pengujian Validasi Melihat Data Santri

Tabel 6.15 Kasus Uji Berhasil Melihat Data Santri

Nama Kasus Uji	Kasus uji berhasil melihat data santri
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin. 2. Menekan menu data santri



Tabel 6.15 Kasus Uji Berhasil Melihat Data Santri (Lanjutan)

Hasil yang diharapkan	Memperlihatkan data santri.
Hasil	Memperlihatkan data santri.
Status	Valid.

6.3.4 Pengujian Validasi Mengubah Data Santri

Tabel 6.16 Kasus Uji Berhasil Mengubah Data Santri

Nama Kasus Uji	Kasus uji berhasil mengubah data santri
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Memilih menu data santri 3. Memilih tombol ubah pada salah satu data santri 4. Melakukan pengisian perubahan data santri 5. Tekan tombol simpan
Hasil yang diharapkan	Memperlihatkan pesan " <i>The item has been modified successfully</i> "
Hasil	Memperlihatkan pesan " <i>The item has been modified successfully</i> "
Status	Valid.

Tabel 6.17 Kasus Uji Tidak berhasil Mengubah Data Santri

Nama Kasus Uji	Kasus uji tidak berhasil mengubah data santri
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Memilih menu data santri 3. Memilih tombol ubah pada salah satu data santri 4. Tidak melakukan pengisian salah satu kolom 5. Menekan tombol simpan
Hasil yang diharapkan	Memperlihatkan pesan "harap isi kolom kosong"
Hasil	Memperlihatkan pesan "harap isi kolom kosong"
Status	Valid.

6.3.5 Pengujian Validasi Menambah Data Santri

Tabel 6.18 Kasus Uji Berhasil Menambah Data Santri

Nama Kasus Uji	Kasus uji berhasil menambah data santri.
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 1. Menekan menu data santri 2. Menekan tombol buat santri



Tabel 6.18 Kasus Uji Berhasil Menambah Data Santri (Lanjutan)

	<ol style="list-style-type: none"> 3. Melakukan pengisian seluruh data santri 4. Tekan tombol simpan
Hasil yang diharapkan	Memperlihatkan pesan <i>"The item has been added successfully"</i>
Hasil	Memperlihatkan pesan <i>"The item has been added successfully"</i>
Status	Valid.

Tabel 6.19 Kasus Uji Tidak berhasil Menambah Data Santri

Nama Kasus Uji	Kasus uji tidak berhasil menambah data santri
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu data santri 3. Menekan tombol buat santri 4. Tidak melakukan pengisian pada salah satu kolom 5. Menekan tombol simpan
Hasil yang diharapkan	Memperlihatkan pesan "harap isi kolom kosong"
Hasil	Memperlihatkan pesan "harap isi kolom kosong"
Status	Valid.

6.3.6 Pengujian Validasi Menghapus Data Santri

Tabel 6.20 Kasus Uji Berhasil Menghapus Data Santri

Nama Kasus Uji	Kasus uji berhasil menghapus data santri
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu data santri 3. Menekan tombol <i>delete</i> pada salah satu santri 4. Tekan tombol delete
Hasil yang diharapkan	Memperlihatkan pesan <i>"The item has been deleted successfully"</i>
Hasil	Memperlihatkan pesan <i>"The item has been deleted successfully"</i>
Status	Valid.



6.3.7 Pengujian Validasi Melihat Data Pengguna

Tabel 6.21 Kasus Uji Berhasil Melihat Data Pengguna

Nama Kasus Uji	Kasus uji berhasil melihat data pengguna
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu pengguna sistem 3. Menekan menu pengguna 4. Menekan tombol <i>preview</i>
Hasil yang diharapkan	Berhasil melihat data pengguna
Hasil	Berhasil melihat data pengguna.
Status	Valid.

6.3.8 Pengujian Validasi Mengubah Data Pengguna

Tabel 6.22 Kasus Uji Berhasil Mengubah Data Pengguna

Nama Kasus Uji	Kasus uji berhasil mengubah data pengguna
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu pengguna sistem 3. Menekan menu pengguna 4. Tekan tombol <i>edit</i> 5. Melakukan pengisian perubahan data santri 6. Tekan tombol "<i>Save and back</i>"
Hasil yang diharapkan	Berhasil mengubah data pengguna.
Hasil	Berhasil mengubah data pengguna.
Status	Valid.

Tabel 6.23 Kasus Uji Tidak berhasil Mengubah Data Pengguna

Nama Kasus Uji	Kasus uji berhasil mengubah data pengguna
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu pengguna sistem 3. Menekan menu pengguna 4. Tekan tombol <i>edit</i> 5. Tidak melakukan pengisian salah satu kolom 6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Memperlihatkan pesan "harap isi kolom kosong"
Hasil	Memperlihatkan pesan "harap isi kolom kosong"
Status	Valid.



6.3.9 Pengujian Validasi Menambah Data Pengguna

Tabel 6.24 Kasus Uji Berhasil Menambah Data Pengguna

Nama Kasus Uji	Kasus uji berhasil menambah data pengguna.
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu pengguna sistem 3. Menekan menu pengguna 4. Tekan tombol buat pengguna 5. Melakukan pengisian seluruh data pengguna 6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Berhasil menambah data pengguna
Hasil	Berhasil menambah data pengguna
Status	Valid.

Tabel 6.25 Kasus Uji Tidak berhasil Menambah Data Pengguna

Nama Kasus Uji	Kasus uji tidak berhasil menambah data pengguna.
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu pengguna sistem 3. Menekan menu pengguna 4. Menekan tombol buat pengguna 5. Tidak melakukan pengisian salah satu kolom 6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Memperlihatkan pesan "harap isi kolom kosong"
Hasil	Memperlihatkan pesan "harap isi kolom kosong"
Status	Valid.

6.3.10 Pengujian Validasi Menghapus Data Pengguna

Tabel 6.26 Kasus Uji Berhasil Menghapus Data Pengguna

Nama Kasus Uji	Kasus uji berhasil menghapus data pengguna
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 1. Menekan menu pengguna sistem 2. Menekan menu pengguna 3. Menekan tombol <i>delete</i> pada salah satu pengguna 4. Tekan tombol delete
Hasil yang diharapkan	Memperlihatkan pesan " <i>The item has been deleted successfully</i> "



Tabel 6.26 Kasus Uji Berhasil Menghapus Data Pengguna (Lanjutan)

Hasil	Menampilkan pesan "The item has been deleted successfully"
Status	Valid.

6.3.11 Pengujian Validasi Melihat Data Mata Pelajaran

Tabel 6.27 Kasus Uji Berhasil Melihat Data Mata Pelajaran

Nama Kasus Uji	Kasus uji berhasil melihat data mata pelajaran
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu mata pelajaran
Hasil yang diharapkan	Berhasil melihat data mata pelajaran
Hasil	Berhasil melihat data mata pelajaran
Status	Valid.

6.3.12 Pengujian Validasi Mengubah Data Mata Pelajaran

Tabel 6.28 Kasus Uji Berhasil Mengubah Data Mata Pelajaran

Nama Kasus Uji	Kasus uji berhasil mengubah data mata pelajaran
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu mata pelajaran 4. Tekan tombol <i>edit</i> di mata pelajaran yang hendak diubah datanya. 5. Melakukan perubahan data mata pelajaran 6. Tekan tombol "<i>save and back</i>"
Hasil yang diharapkan	Berhasil mengubah data mata pelajaran
Hasil	Berhasil mengubah data mata pelajaran
Status	Valid.

Tabel 6.29 Kasus Uji Tidak berhasil Mengubah Data Mata Pelajaran

Nama Kasus Uji	Kasus uji tidak berhasil mengubah data mata pelajaran
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu mata pelajaran 4. Tekan tombol <i>edit</i> pada mata pelajaran yang hendak diubah. 5. Tidak melakukan pengisian salah satu kolom



Tabel 6.29 Kasus Uji Tidak berhasil Mengubah Data Mata Pelajaran (Lanjutan)

Prosedur	6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Memperlihatkan pesan “harap isi kolom kosong”
Hasil	Memperlihatkan pesan “harap isi kolom kosong”
Status	Valid.

6.3.13 Pengujian Validasi Menambah Data Mata Pelajaran

Tabel 6.30 Kasus Uji Berhasil Menambah Data Mata Pelajaran

Nama Kasus Uji	Kasus uji berhasil menambah data mata pelajaran
Prosedur	1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu mata pelajaran 4. Menekan tombol buat mata pelajaran 5. Melakukan pengisian data mata pelajaran 6. Tekan tombol “ <i>Save and back</i> ”
Hasil yang diharapkan	Berhasil melakukan penambahan mata pelajaran.
Hasil	Berhasil menambah data mata pelajaran.
Status	Valid.

Tabel 6.31 Kasus Uji Tidak berhasil Menambah Data Mata Pelajaran

Nama Kasus Uji	Kasus uji tidak berhasil menambah data mata pelajaran
Prosedur	1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu mata pelajaran 4. Menekan tombol buat mata pelajaran 5. Tidak melakukan pengisian salah satu kolom 6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Memperlihatkan pesan “harap isi kolom kosong”
Hasil	Memperlihatkan pesan “harap isi kolom kosong”
Status	Valid.

6.3.14 Pengujian Validasi Menghapus Data Mata Pelajaran

Tabel 6.32 Kasus Uji Berhasil Menghapus Data Mata Pelajaran

Nama Kasus Uji	Kasus uji berhasil menghapus data mata pelajaran
Prosedur	1. Login sebagai admin



Tabel 6.32 Kasus Uji Berhasil Menghapus Data Mata Pelajaran (Lanjutan)

	<ol style="list-style-type: none"> 2. Menekan menu administrasi 3. Menekan menu mata pelajaran 4. Tekan tombol <i>delete</i> pada mata pelajaran yang hendak dihilangkan datanya 5. Tekan tombol delete
Hasil yang diharapkan	Memperlihatkan pesan " <i>The item has been deleted successfully</i> "
Hasil	Memperlihatkan pesan " <i>The item has been deleted successfully</i> "
Status	Valid.

6.3.15 Pengujian Validasi Melihat Data Kepemimpinan

Tabel 6.33 Kasus Uji Berhasil Melihat Data Kepemimpinan

Nama Kasus Uji	Kasus uji berhasil melihat data kepemimpinan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu kepemimpinan
Hasil yang diharapkan	Berhasil melihat data kepemimpinan
Hasil	Berhasil melihat data kepemimpinan
Status	Valid.

6.3.16 Pengujian Validasi Mengubah Data Kepemimpinan

Tabel 6.34 Kasus Uji Berhasil Mengubah Data Kepemimpinan

Nama Kasus Uji	Kasus uji berhasil mengubah data kepemimpinan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu kepemimpinan 4. Tekan tombol <i>edit</i> di data kepemimpinan yang hendak diubah 5. Mengubah data kepemimpinan 6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Berhasil mengubah data kepemimpinan
Hasil	Berhasil mengubah data kepemimpinan
Status	Valid



Tabel 6.35 Kasus Uji Tidak berhasil Mengubah Data Kepemimpinan

Nama Kasus Uji	Kasus uji tidak berhasil mengubah data kepemimpinan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu kepemimpinan 4. Tekan tombol <i>edit</i> di data kepemimpinan yang hendak diubah 5. Tidak melakukan pengisian salah satu kolom 6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Memperlihatkan pesan "harap isi kolom kosong"
Hasil	Memperlihatkan pesan "harap isi kolom kosong"
Status	Valid.

6.3.17 Pengujian Validasi Menambah Data Kepemimpinan

Tabel 6.36 Kasus Uji Berhasil Menambah Data Kepemimpinan

Nama Kasus Uji	Kasus uji berhasil menambah data Kepemimpinan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu Kepemimpinan 4. Tekan tombol buat Kepemimpinan 5. Melakukan pengisian data Kepemimpinan 6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Berhasil menambah data Kepemimpinan.
Hasil	Berhasil menambah data Kepemimpinan.
Status	Valid.

Tabel 6.37 Kasus Uji Tidak berhasil Menambah Data Kepemimpinan

Nama Kasus Uji	Kasus uji tidak berhasil menambah data kepemimpinan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu kepemimpinan 4. Menekan tombol buat kepemimpinan 5. Tidak melakukan pengisian salah satu kolom 6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Memperlihatkan pesan "harap isi kolom kosong"
Hasil	Memperlihatkan pesan "harap isi kolom kosong"



Tabel 6.37 Kasus Uji Tidak berhasil Menambah Data Kepemimpinan (Lanjutan)

Status	Valid.
--------	--------

6.3.18 Pengujian Validasi Menghapus Data Kepemimpinan

Tabel 6.38 Kasus Uji Berhasil Menghapus Data Kepemimpinan

Nama Kasus Uji	Kasus uji berhasil menghapus data kepemimpinan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu kepemimpinan 4. Tekan tombol <i>delete</i> di data kepemimpinan yang hendak dihapus 5. Tekan tombol delete
Hasil yang diharapkan	Memperlihatkan pesan " <i>The item has been deleted successfully</i> "
Hasil	Memperlihatkan pesan " <i>The item has been deleted successfully</i> "
Status	Valid.

6.3.19 Pengujian Validasi Melihat Data Hafalan

Tabel 6.39 Kasus Uji Berhasil Melihat Data Hafalan

Nama Kasus Uji	Kasus uji berhasil melihat data hafalan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu hafalan
Hasil yang diharapkan	Berhasil melihat data hafalan
Hasil	Berhasil melihat data hafalan
Status	Valid.

6.3.20 Pengujian Validasi Mengubah Data Hafalan

Tabel 6.40 Kasus Uji Berhasil Mengubah Data Hafalan

Nama Kasus Uji	Kasus uji berhasil mengubah data hafalan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu hafalan 4. Tekan tombol <i>edit</i> di data hafalan yang hendak diubah. 5. Melakukan perubahan pada data hafalan 6. Menekan tombol <i>save and back</i>



Tabel 6.40 Kasus Uji Berhasil Mengubah Data Hafalan (Lanjutan)

Hasil yang diharapkan	Berhasil mengubah data hafalan
Hasil	Berhasil mengubah data hafalan
Status	Valid.

Tabel 6.41 Kasus Uji Tidak berhasil mengubah data hafalan

Nama Kasus Uji	Kasus uji tidak berhasil mengubah data hafalan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu hafalan 4. Tekan tombol <i>edit</i> di hafalan yang hendak diubah 5. Tidak melakukan pengisian salah satu kolom 6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Memperlihatkan pesan "harap isi kolom kosong"
Hasil	Memperlihatkan pesan "harap isi kolom kosong"
Status	Valid.

6.3.21 Pengujian Validasi Menambah Data Hafalan

Tabel 6.42 Kasus Uji Berhasil Menambah Data Hafalan

Nama Kasus Uji	Kasus uji berhasil menambah data hafalan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu hafalan 4. Tekan tombol buat hafalan 5. Melakukan pengisian data hafalan 6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Berhasil menambah data hafalan.
Hasil	Berhasil menambah data hafalan.
Status	Valid

Tabel 6.43 Kasus Uji Tidak berhasil Menambah Data Hafalan

Nama Kasus Uji	Kasus uji tidak berhasil menambah data hafalan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu hafalan 4. Menekan tombol buat hafalan 5. Tidak melakukan pengisian salah satu kolom



Tabel 6.43 Kasus Uji Tidak berhasil Menambah Data Hafalan (Lanjutan)

	6. Menekan tombol <i>save and back</i>
Hasil yang diharapkan	Memperlihatkan pesan “harap isi kolom kosong”
Hasil	Memperlihatkan pesan “harap isi kolom kosong”
Status	Valid.

6.3.22 Pengujian Validasi Menghapus Data Hafalan

Tabel 6.44 Kasus Uji Berhasil Menghapus Data Hafalan

Nama Kasus Uji	Kasus uji berhasil menghapus data hafalan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu administrasi 3. Menekan menu hafalan 4. Tekan tombol <i>delete</i> di hafalan yang dikehendaki 5. Tekan tombol delete
Hasil yang diharapkan	Memperlihatkan pesan “ <i>The item has been deleted successfully</i> ”
Hasil	Memperlihatkan pesan “ <i>The item has been deleted successfully</i> ”
Status	Valid.

6.3.23 Pengujian Validasi Daftar Melihat Pelanggaran Santri

Tabel 6.45 Kasus Uji Berhasil Melihat Pelanggaran Santri

Nama Kasus Uji	Kasus uji berhasil melihat pelanggaran santri
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai ustaz atau <i>murobbi</i> 2. Membuka menu pelanggaran santri.
Hasil yang diharapkan	Berhasil melihat pelanggaran santri
Hasil	Berhasil melihat pelanggaran santri
Status	Valid.

6.3.24 Pengujian Validasi Menambah Pelanggaran Santri

Tabel 6.46 Kasus Uji Berhasil Menambah Pelanggaran Santri

Nama Kasus Uji	Kasus uji berhasil menambah pelanggaran santri
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai ustaz atau <i>murobbi</i> 2. Membuka menu pelanggaran santri. 3. Menekan tombol buat pelanggaran santri 4. Melakukan pengisian data pelanggaran santri 5. Tekan tombol “<i>Save and back</i>”



Tabel 6.46 Kasus Uji Berhasil Menambah Pelanggaran Santri (Lanjutan)

Hasil yang diharapkan	Berhasil melakukan penambahan pelanggaran santri
Hasil	Berhasil melakukan penambahan pelanggaran santri
Status	Valid

Tabel 6.47 Kasus Uji Tidak berhasil Menambah Pelanggaran Santri

Nama Kasus Uji	Kasus uji tidak berhasil menambah pelanggaran santri
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai ustaz atau <i>murobbi</i> 2. Membuka menu pelanggaran santri 3. Menekan tombol buat pelanggaran santri 4. Tidak melakukan pengisian salah satu data pelanggaran santri 5. Menekan tombol "Save and back"
Hasil yang diharapkan	Memperlihatkan pesan "harap isi kolom kosong"
Hasil	Memperlihatkan pesan "harap isi kolom kosong"
Status	Valid

6.3.25 Pengujian Validasi Melihat Nilai Mata Pelajaran

Tabel 6.48 Kasus Uji Berhasil Melihat Nilai Mata Pelajaran

Nama Kasus Uji	Kasus uji berhasil melihat nilai mata pelajaran
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai ustaz atau wali santri 2. Membuka menu penilaian 3. Menekan jenis penilaian 4. Menekan kelas dan mata pelajaran
Hasil yang diharapkan	Berhasil melihat nilai mata pelajaran
Hasil	Berhasil melihat nilai mata pelajaran
Status	Valid

6.3.26 Pengujian Validasi Menambah Nilai Mata Pelajaran

Tabel 6.49 Kasus Uji Berhasil Menambah Nilai Mata Pelajaran

Nama Kasus Uji	Kasus uji berhasil menambah nilai mata pelajaran
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai ustaz 2. Membuka menu penilaian 3. Menekan jenis penilaian 4. Menekan kelas dan mata pelajaran 5. Tekan tombol tambah nilai mata pelajaran



Tabel 6.49 Kasus Uji Berhasil Menambah Nilai Mata Pelajaran (Lanjutan)

Prosedur	6. Melakukan pengisian nilai mata pelajaran 7. Tekan tombol <i>save</i>
Hasil yang diharapkan	Berhasil menambah nilai mata pelajaran
Hasil	Berhasil menambah nilai mata pelajaran
Status	Valid

Tabel 6.50 Kasus Uji Berhasil Menambah Nilai Mata Pelajaran dengan *file excel*

Nama Kasus Uji	Kasus uji berhasil menambah nilai mata pelajaran dengan <i>file excel</i>
Prosedur	1. Login sebagai ustaz 2. Membuka menu penilaian 3. Menekan jenis penilaian 4. Menekan kelas dan mata pelajaran 5. Menekan tombol tambah nilai mata pelajaran 6. Mengunduh template <i>file excel</i> 7. Memasukkan nilai di <i>file excel</i> 8. <i>Upload file excel</i> 9. Tekan tombol <i>save</i>
Hasil yang diharapkan	Berhasil menambah nilai mata pelajaran dengan <i>file excel</i>
Hasil	Berhasil menambah nilai mata pelajaran dengan <i>file excel</i>
Status	Valid.

Tabel 6.51 Kasus Uji Berhasil *Reset* Nilai Mata Pelajaran dengan *File Excel*

Nama Kasus Uji	Kasus uji berhasil <i>reset</i> nilai mata pelajaran
Prosedur	1. Login sebagai ustaz 2. Membuka menu penilaian 3. Menekan jenis penilaian 4. Menekan kelas dan mata pelajaran 5. <i>Reset</i> nilai mata pelajaran menjadi kosong 6. Tekan tombol <i>save</i>
Hasil yang diharapkan	Berhasil <i>reset</i> nilai mata pelajaran
Hasil	Berhasil <i>reset</i> nilai mata pelajaran
Status	Valid.



6.3.27 Pengujian Validasi Mengubah Nilai Mata Pelajaran

Tabel 6.52 Kasus Uji Berhasil Mengubah Nilai Mata Pelajaran

Nama Kasus Uji	Kasus uji berhasil menambah nilai mata pelajaran
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai ustaz 2. Membuka menu penilaian 3. Menekan jenis penilaian 4. Menekan kelas dan mata pelajaran 5. Menekan tombol <i>edit</i> 6. Memasukkan nilai mata pelajaran 7. Tekan tombol <i>save</i>
Hasil yang diharapkan	Berhasil menambah nilai mata pelajaran
Hasil	Berhasil menambah nilai mata pelajaran
Status	Valid

6.3.28 Pengujian Validasi Melihat Nilai Kepemimpinan

Tabel 6.53 Kasus Uji Berhasil Melihat Nilai Kepemimpinan

Nama Kasus Uji	Kasus uji berhasil melihat nilai kepemimpinan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai <i>murobbi</i> atau wali santri 2. Menekan menu penilaian 3. Tekan menu nilai kepemimpinan
Hasil yang diharapkan	Berhasil melihat nilai kepemimpinan
Hasil	Berhasil melihat nilai kepemimpinan
Status	Valid

6.3.29 Pengujian Validasi Mengubah Nilai Kepemimpinan

Tabel 6.54 Kasus Uji Berhasil Mengubah Nilai Kepemimpinan

Nama Kasus Uji	Kasus uji berhasil mengubah nilai kepemimpinan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai <i>murobbi</i> 2. Menekan menu penilaian 3. Menekan menu nilai kepemimpinan 4. Tekan tombol <i>edit</i> 5. Melakukan pengisian perubahan nilai kepemimpinan 6. Tekan tombol <i>save</i>
Hasil yang diharapkan	Berhasil mengubah nilai kepemimpinan
Hasil	Berhasil mengubah nilai kepemimpinan
Status	Valid



Tabel 6.55 Kasus Uji Tidak berhasil Mengubah Nilai Kepemimpinan

Nama Kasus Uji	Kasus uji tidak berhasil mengubah nilai kepemimpinan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai <i>murobbi</i> 2. Menekan menu penilaian 3. Menekan menu nilai kepemimpinan 4. Tekan tombol <i>edit</i> 5. Tidak melakukan pengisian salah satu penilaian kepemimpinan 6. Tekan tombol <i>save</i>
Hasil yang diharapkan	Memperlihatkan pesan “harap isi kolom kosong”
Hasil	Memperlihatkan pesan “harap isi kolom kosong”
Status	Valid.

6.3.30 Pengujian Validasi Menambah Nilai Kepemimpinan

Tabel 6.56 Kasus Uji Berhasil Menambah Nilai Kepemimpinan

Nama Kasus Uji	Kasus uji tidak berhasil mengubah nilai kepemimpinan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai <i>murobbi</i> 2. Menekan menu penilaian 3. Menekan menu nilai kepemimpinan 4. Menekan tombol tambah nilai kepemimpinan 5. Melakukan pengisian nilai kepemimpinan 6. Tekan tombol <i>save</i>
Hasil yang diharapkan	Berhasil menambah nilai kepemimpinan
Hasil	Berhasil menambah nilai kepemimpinan
Status	Valid.

6.3.31 Pengujian Validasi Melihat Nilai Hafalan

Tabel 6.57 Kasus Uji Berhasil Melihat Nilai Hafalan

Nama Kasus Uji	Kasus uji berhasil melihat nilai hafalan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai <i>murobbi</i> atau wali santri 2. Menekan menu penilaian 3. Tekan menu nilai hafalan
Hasil yang diharapkan	Berhasil melihat nilai hafalan
Hasil	Berhasil melihat nilai hafalan.
Status	Valid.



6.3.32 Pengujian Validasi Mengubah Nilai Hafalan

Tabel 6.58 Kasus Uji Berhasil Mengubah Nilai Hafalan

Nama Kasus Uji	Kasus uji berhasil mengubah nilai hafalan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai <i>murobbi</i> 2. Menekan menu penilaian 3. Menekan menu nilai hafalan 4. Tekan tombol <i>edit</i> 5. Melakukan pengisian perubahan nilai hafalan 6. Tekan tombol <i>save</i>
Hasil yang diharapkan	Berhasil mengubah nilai hafalan
Hasil	Berhasil mengubah nilai hafalan
Status	Valid.

6.3.33 Pengujian Validasi Menambah Nilai Hafalan

Tabel 6.59 Kasus Uji Berhasil Menambah Nilai Hafalan

Nama Kasus Uji	Kasus uji berhasil mengubah nilai hafalan
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai <i>murobbi</i> 2. Menekan menu penilaian 3. Menekan menu nilai hafalan 4. Menekan tombol tambah 5. Melakukan pengisian nilai hafalan 6. Tekan tombol <i>save</i>
Hasil yang diharapkan	Berhasil menambah nilai hafalan
Hasil	Berhasil menambah nilai hafalan
Status	Valid.

6.3.34 Pengujian Validasi Mencetak Rapor

Tabel 6.60 Kasus Uji Berhasil Mencetak Rapor

Nama Kasus Uji	Kasus uji berhasil mencetak rapor
Prosedur	<ol style="list-style-type: none"> 1. Login sebagai admin 2. Menekan menu cetak rapor 3. Menekan kelas 4. Tekan tombol cetak
Hasil yang diharapkan	Berhasil mencetak rapor
Hasil	Berhasil mencetak rapor
Status	Valid.



6.4 Pengujian Usabilitas

Dalam tahap pengujian *usability* beberapa responden yang secara acak dipilih dan dilibatkan dalam melakukan percobaan pada sistem ini. Pengujian *usability* terhadap sistem memakai metode *System Usability Scale* (SUS).

6.4.1 Prosedur Pengujian *Usability*

Pengujian *usability* menggunakan metode SUS dilakukan dengan memberikan beberapa pertanyaan kepada sejumlah responden yang diberikan kesempatan untuk mencoba mengoperasikan sistem. Dalam pengujian ini, responden yang berpartisipasi adalah orang yang akan menjadi calon pengguna dan dipilih secara acak. Responden dipersilakan untuk menjawab pertanyaan-pertanyaan yang diberikan setelah mencoba sistem yang telah dibangun. Daftar pertanyaan menggunakan metode SUS dapat dilihat pada Tabel 6.61.

Tabel 6.61 Daftar Pertanyaan Pengujian *Usability* menggunakan Metode SUS

No.	Pertanyaan
1.	Saya merasa ingin memakai sistem ini secara berkelanjutan
2.	Saya merasakan bahwa penggunaan sistem ini cukup rumit
3.	Saya merasakan bahwa sistem ini mudah untuk digunakan
4.	Saya merasa memerlukan bantuan teknis dari orang yang ahli untuk memakai sistem ini
5.	Saya menemukan beragam fitur yang terdapat dalam sistem terintegrasi dengan baik
6.	Saya merasa banyak ketidakketetapan yang terdapat dalam sistem ini
7.	Saya merasa orang lain akan mempelajari cara penggunaan sistem ini dengan mudah dan cepat
8.	Saya berasumsi bahwa sistem ini tidak praktis untuk dipakai
9.	Saya mempunyai rasa percaya diri ketika menggunakan sistem ini
10.	Saya merasa harus mempelajari lebih banyak mengenai penggunaan sistem ini

6.4.2 Analisis dan Hasil Pengujian *Usability*

Sesudah selesai melakukan pengujian *usability* dan nilai respon yang didapat telah diberikan oleh responden, proses selanjutnya adalah menganalisa hasil pengujian tersebut. Hasil pengujian berdasarkan pertanyaan yang diberikan kepada responden dapat dilihat pada Tabel 6.62.



Tabel 6.62 Hasil Kuisioner Responden Terhadap Pertanyaan Metode SUS

Nama	Pertanyaan									
	1	2	3	4	5	6	7	8	9	10
Gelis Marga	4	1	5	3	4	2	4	3	4	2
Alirsyad Pahlevi Susilo	3	1	5	2	4	2	4	3	4	2
Dzaky Fadhillah Guci	5	2	5	3	4	1	4	2	4	3
Firdaus Arsy Ramadhan	2	2	4	3	4	2	4	2	4	3
Muhammad Fadel	4	2	4	2	4	2	3	2	4	2

Setelah kuisioner diisi oleh responden, kemudian dilakukan perhitungan skor akhir yang didasarkan penilaian yang diberikan responden dalam kuisioner tersebut. Perhitungan jumlah skor akhir mengambil nilai yang terdapat pada Tabel 6.61 dengan ketentuan nilai dari pertanyaan ganjil akan dikurangi 1 (nilai) dan 5 (lima) nilai akan dikurangi nilai dari pertanyaan genap. Hasil perhitungan skor akhir dapat dilihat pada Tabel 6.63.

Tabel 6.63 Hasil Perhitungan Skor Akhir Penilaian Kuisioner

Nama	Pernyataan										Jumlah Score	Dikali 2,5
	1	2	3	4	5	6	7	8	9	10		
Gelis Marga	3	4	4	2	3	3	3	2	3	3	30	75
Alirsyad Pahlevi Susilo	2	4	4	3	3	3	3	2	3	3	30	75
Dzaky Fadhillah Guci	4	3	4	2	3	4	3	3	3	2	31	77,5
Firdaus Arsy Ramadhan	1	3	3	2	3	3	3	3	3	2	26	65
Muhammad Fadel	3	3	3	3	3	3	2	3	3	3	28	70
Rata-rata												72,5

Berdasarkan hasil skor akhir yang didapatkan pada Tabel 6.63, skor akhir yang diberikan dari kelima responden dalam pengujian usability sistem adalah 72,5. Dengan rata-rata skor tersebut dapat disimpulkan bahwa sistem masuk ke dalam kategori *acceptable* yang dapat diartikan bahwa sistem yang diimplementasikan mencapai kesesuaian dengan apa yang dibutuhkan oleh pengguna.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasar pada hasil penelitian dari awal sampai dengan akhir maka dapat disimpulkan:

1. Hasil analisis kebutuhan pengembangan sistem penilaian santri terpadu di al-izzah *leadership school* berbasis web, memiliki kebutuhan fungsional sebanyak tiga puluh empat dan satu kebutuhan non-fungsional yang menjadi perolehan hasil di tahap ini. Selain itu, admin, ustaz, *murobbi*, dan wali santri ialah aktor yang berpartisipasi pada sistem. Hasil analisis dari studi literatur, wawancara, dan observasi telah disesuaikan dengan kebutuhan yang dibutuhkan oleh pondok pesantren Al-Izzah.
2. Tahap perancangan menggunakan hasil yang didapat dari analisis kebutuhan. Hasil yang diperoleh dari tahap perancangan adalah perancangan sequence diagram, perancangan class diagram, perancangan algoritme, perancangan basis data dan perancangan antarmuka.
3. Spesifikasi pengembangan terhadap sistem, implementasi basis data, implementasi kode program dan implementasi antarmuka merupakan hasil yang didapatkan pada tahap implementasi dari sistem penilaian santri terpadu di al-izzah *leadership school* berbasis web.
4. Pada penelitian ini dilakukan pengujian unit dan pengujian integrasi yang termasuk dalam *whitebox testing*, dan pengujian validasi yang termasuk dalam *blackbox testing*. Dalam tahap pengujian validasi, hasil 100% valid didapatkan untuk pengujian kebutuhan fungsional. Hasil skor 72,5 dan dikategorikan acceptable didapatkan dari pengujian usability yang termasuk dalam pengujian kebutuhan non-fungsional dan menandakan bahwa sistem yang dibangun sudah memenuhi kebutuhan pengguna.

7.2 Saran

Pada penelitian selanjutnya terhadap pengembangan sistem penilaian santri terpadu, terdapat beberapa saran yang dapat dilakukan yaitu:

1. Menambahkan sistem administrasi *online* untuk memudahkan para wali santri dalam hal pembayaran uang sumbangan pembinaan pendidikan (SPP) dan untuk pembayaran daftar ulang setiap semesternya.
2. Menambahkan fitur pengelolaan prestasi santri diluar akademik untuk menjadi bekal tambahan pada tahap kelulusan dalam melakukan pendaftaran ke universitas.
3. Menambahkan fitur notifikasi langsung ke gawai para wali santri agar mendapatkan berita terbaru dari perkembangan anak dalam akademik maupun hafalan Qur'an mereka.



DAFTAR REFERENSI

- Al-Izzah, M., 2015. *Kategori : Sejarah Al-Izzah*. [online] Available at: <<https://alizzah-batu.sch.id/category/sejarah-alizzah/>> [Accessed 28 Sep. 2019].
- Booch, G., Rumbaugh, J. and Jacobson, I., 2005. *The Unified Modeling Language User Guide SECOND EDITION*. Language.
- Ceka, A. and Murati, R., 2016. The Role of Parents in the Education of Children. *Journal of Education and Practice*.
- V. Forrester, V., 2019. SCHOOL MANAGEMENT INFORMATION SYSTEMS: CHALLENGES TO EDUCATIONAL DECISIONMAKING IN THE BIG DATA ERA. *International Journal on Integrating Technology in Education*.
- Fu, M., Bo, W.V., Xue, Y. and Yuan, T.E., 2017. Parental Absence Accompanies Worse Academic Achievements: Evidence Based upon a Sample of Left-Behind Children in Rural China. *Frontiers in Education*.
- Ghuman, S.S., 2014. International Journal of Computer Science and Mobile Computing. Software Testing Techniques. *International Journal of Computer Science and Mobile Computing*.
- Hinze-Pifer, R. and Ramsey, D.S., 2011. Evaluating Education Information Systems: Implementation of Longitudinal Student Data Systems in Six School Districts. *Policy Perspectives*.
- Kamaruddin, K. and Patak, A.A., 2018. The Role of Islamic Education Teachers in Instilling Student Discipline. *International Journal on Advanced Science, Education, and Religion*.
- M, N., Nagaraj, A., Gattu, H. and Shetty, P.K., 2014. Research Study on Importance of Usability Testing/ User Experience (UX) Testing. *International Journal of Computer Science and Mobile Computing*.
- Nagar, M.A.K., Rahoo, L.A., Renman, H.A. and Arshad, S., 2019. Education Management Information Systems in the Primary Schools of Sindh a case study of Hyderabad Division. In: *2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences, ICETAS 2018*.
- Ndiku, J.M., Oyoo, C.N. and Owano, A., 2014. Student Data Management and School Decision Making in Kenya. *International Journal of Education and Research*.
- Parkar, V.V., Shinde, P.P., Gadade, S.C. and Shinde, P.M., 2016. Utilization of Laravel Framework for Development of Web Based Recruitment Tool. *National Conference On 'Changing Technology and Rural Development'*.
- Sommerville, I., 2016. *Software engineering (10th edition)*. Pearson Education Limited.



Stauffer, M., 2017. *Laravel: Up & Running. Journal of Chemical Information and Modeling.*

Wang, T.H., 2018. Developing a web-based assessment system for evaluating examinee's understanding of the procedure of scientific experiments. *Eurasia Journal of Mathematics, Science and Technology Education.*

Zuhriy, M.S., 2011. BUDAYA PESANTREN DAN PENDIDIKAN KARAKTER PADA PONDOK PESANTREN SALAF. *Walisongo: Jurnal Penelitian Sosial Keagamaan.*



LAMPIRAN A WAWANCARA

A.1 Wawancara

Narasumber : Ustadz Budi Utomo

Posisi : Guru dan Pembimbing Asrama Al-Izzah Leadership School

1. Bagaimana proses pendidikan di Al-Izzah Leadership School ?

Al-Izzah Leadership School mengembankan kurikulum seperti halnya SMP dan SMA pada umumnya, kami menyediakan penjurusan untuk IPS dan IPA pada tahap SMA, dan kami juga mempunyai kurikulum diniyah yang mempelajari pelajaran agama islam dan bahasa arab. Pada pesantren Al-Izzah kami juga mengadakan program hafalan Qur'an, pada program hafalan Qur'an terdapat 2 program yaitu, program intensif yang menargetkan untuk hafal 30Juz dalam kurun waktu 6 tahun SMP dan SMA, ada juga program biasa yang menargetkan minimal 10Juz ketika lulus dari pesantren ini, kami juga memberikan sertifikat penghargaan bagi santri yang dapat menghafal Qur'an 3 Juz atau 5 Juz dalam sekali setoran hafalan.

2. Bagaimanakah proses penilaian di Al-Izzah Leadership School ?

- Pada penilaian mata pelajaran para ustad menginputkan nilai dari tugas harian, ulangan, ujian tengah semester dan ujian akhir semester di dalam file excel yang kemudian di upload ke google drive, kemudian pihak akademik pesantren mengolah data tersebut untuk dikategorikan per santri untuk dijadikan rapor bulanan yang berisikan nilai tugas harian dan nilai ulangan yang dilakukan pada bulan tersebut. Setelah nilai-nilai tersebut sudah menjadi rapor untuk masing-masing santri, rapor tersebut dikirimkan ke google drive dengan format pdf untuk dikirimkan ke google drive masing-masing wali kelas untuk di kirimkan secara pribadi kepada setiap wali santri.
- Untuk penilaian hafalan kami masih menggunakan buku laporan hafalan Qur'an yang dipegang oleh masing-masing pembimbing asrama yang digunakan seminggu sekali ketika proses setoran hafalan Qur'an secara serentak, semua proses hafalan santri dicatat pada buku tersebut.

3. Bagaimana struktur dari Al-Izzah Leadership School ?

Pengurus dari al-izzah leadership school terbagi menjadi tiga, pengurus SMP, pengurus SMA, dan pengurus pesantren. Pengurus SMP dan SMA mengurus segala macam administrasi, kurikulum pembelajaran dan juga mengajar di dalam sekolah, sedangkan pengurus pesantren atau yang kami sebut mudabbir atau murobbi mengurus segala macam kegiatan santri di asrama dan juga mengurus proses hafalan Qur'an santri.



4. Berapakan rapor yang dikeluarkan pondok setiap semester ?

Setiap semester kami mengeluarkan 5 rapor yaitu :

- Akademik Pelajaran Umum
- Akademik Pelajaran Diniyah
- Leadership
- Leadership Culture
- Hafalan Qur'an

5. Apakah sudah ada sistem yang dapat membantu proses penilaian di pesantren Al-Izzah ?

Belum ada mas, kami baru punya sistem informasi yang mendeskripsikan pesantren kami aja bisa di liat di alizzah-batu.sch.id, kalau untuk sistem penilaiannya belum ada mas, sebenarnya sudah pake google drive tapi yang ngerti untuk olah datanya cuma 1 orang di bagian akademik, jadi nanti kalo dia cuti atau resign bingung lagi yang bisa ngolah sistemnya siapa, soalnya kan perhitungannya dia yang ngerti.

6. Bagaimanakah pemanfaatan penilaian santri menggunakan sistem file excel dan google drive yang digunakan saat ini ?

Sebenarnya sih sudah lumayan untuk bagian akademiknya, cuma ya itu agak repot juga kalo jadi wali kelas harus download satu persatu raport santri tiap bulan terus harus kirim satu satu ke wali santri, kadang takut juga sih kalo salah kirim nilai santrinya. Sama buat hafalan kan hanya keluar sekali setiap semester, kadang kadang wali santri suka banyak yang nanya juga kalo anaknya hafalannya sudah sampe mana, jadi kadang repot juga saya harus nanya ke pembimbing asramanya dulu dan dia juga harus nyari satu-satu di buku hafalan santrinya.

7. Apakah pendapat ustaz kalau ada sistem penilaian yang mampu terintegrasi dengan semua nilai ?

Ya seneng sekali lah mas, kalau ada sistem yang bisa langsung update gitu di website kan, jadi enak juga kita wali kelas tinggal ngabarin aja di grup kelas wali santri gak perlu ngririm satu satu file nya kan itu bisa ngebantu banget mas gak cuma kita aja wali kelas sama ustaz, pembimbing asrama juga gak perlu pake buku-buku-an lagi kan buat setoran hafalan Qur'an.

Batu, 16 November 2019

Ustaz dan Murobbi Al-Izzah

Budi Utomo



LAMPIRAN B RAPOR

B.1 Rapor Nilai Hafalan

LAPORAN PERKEMBANGAN NILAIAN TENGAH SEMESTER SMA/AL-IZZAH LEADERSHIP SCHOOL BATU

كشف درجات تحفيظ القرآن لتصفيف التحصيل الدراسي (ممارسة الحدة الثانوية للترجمة الترادوية باتو)

الاسم: **S Brawijaya** السنة الدراسية/التربوية: 2019-2020
 Kelas/المصل: X MIA 5 Semester/الفصل الدراسي: Satu/Ganjil

A. CAPAIAN HAFALAN/حصول

JUZYAN HAFALAN/حفظ												
Juz 20	Juz 20	Juz 27	Juz 26	Juz 1	Juz 2	Juz 3	Juz 4	Juz 5	Juz 6	Juz 7	Juz 8	Juz 9
Khatam	Khatam			Khatam	Khatam	Khatam	Khatam	dal.				

B. PENILAIAN HAFALAN/تقييم

NO	URUS EVALUASI	JUZYAN HAFALAN/حفظ										
		Juz 20	Juz 29	Juz 28	Juz 2	Juz 26	Juz 1	Juz 2	Juz 3	Juz 4	Juz 5	Juz 6
1	Ujian Juz Iyah	100	100	100	100	100	100	100	100	100	100	100
2	Ujian Juz Iyah 3 Juz Sekali Duduk	Preparat: Muntaz (Istimewa)										
3	Ujian Juz Iyah 3 Juz Sekali Duduk											

MENTAR GUGH/التقييم: 4.00 (A) dan Syukasi sebagai anugerah dari Allah SWT. Semoga menjadi ilmu yang bermanfaat!

Batu, 12 Oktober 2019
 Kepala Sekolah/مدير المدرسة: [Redacted]
 Kepala UPT/مدير المركز: [Redacted]



B.2 Rapor Nilai Kepemimpinan

LAPORAN LEADERSHIP CULTURE
KEPESANTREHAN AL-IZZAH LEADERSHIP SCHOOL BATU
TAHUN PELAJARAN 2019/2020

NAMA: [REDACTED] SEMESTER: GANJIL
KELAS: X-MIA 5

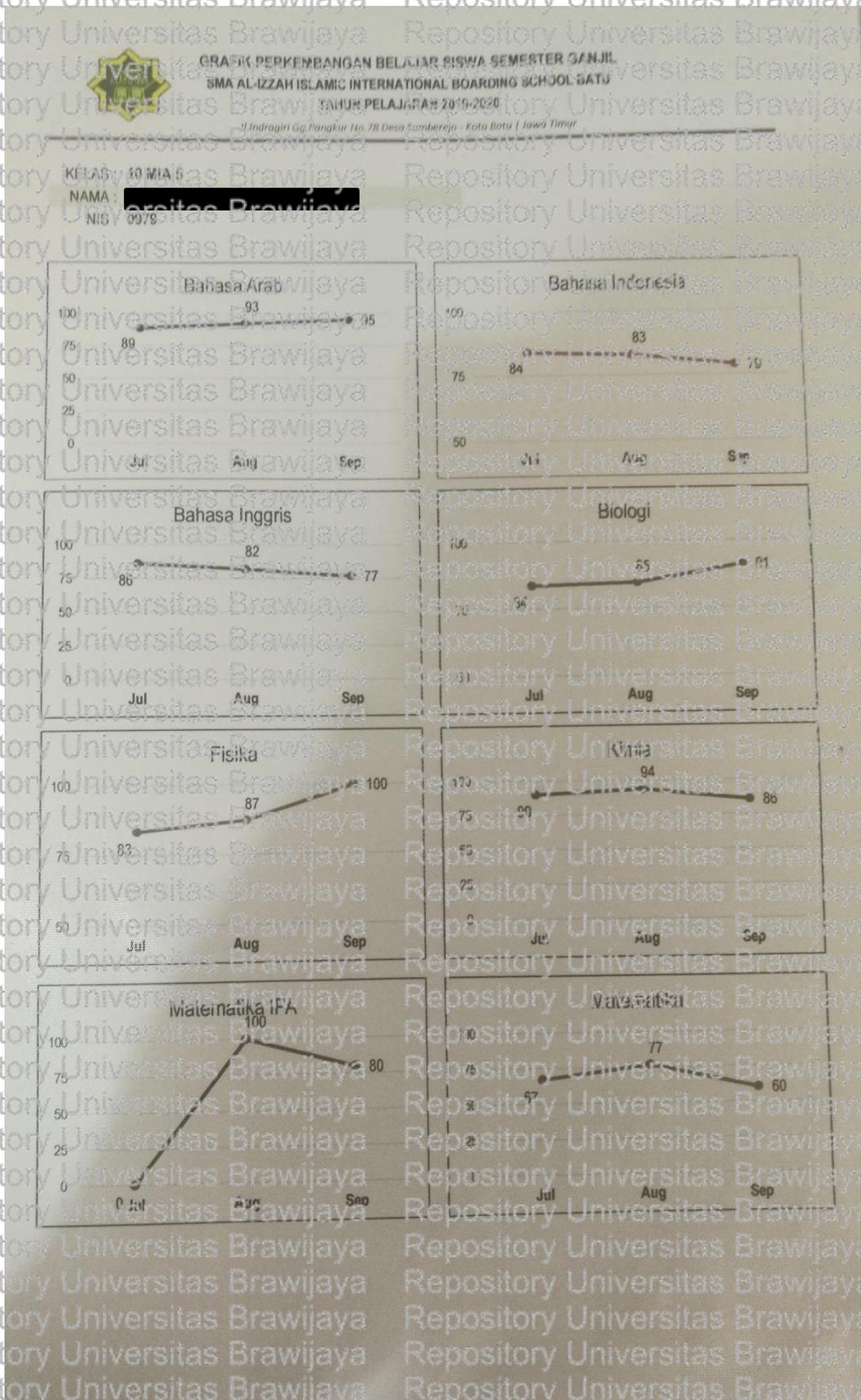
LEADERSHIP CULTURE (7 PILAR KEPEMIMPINAN)	INDIKATOR	NILAI	KETERANGAN
VISIONER	Santri memiliki tujuan yang jelas untuk mencari ilmu di Al-Izzah Leadership School dan mampu menggalang orang lain untuk kerja keras dan kerjasama dalam menggapai tujuan.	A	PENILAIAN A = Sangat Baik
BERJIWA JUANG TINGGI	Santri dapat menunjukkan sikap tangguh, penuh semangat, tidak putus asa, bekerja keras, baah, dan tidak kosa letak.	A	B = Baik
SABAR PANTANG MENYERAH	Santri memiliki kesabaran dan ketabahan dalam berusaha dan memiliki semangat dan tabah uji dari setiap tantangan.	A	C = Cukup
PERANI DAN BERTANGGUNG JAWAB	Santri memiliki jiwa siap memimpin dan siap dipimpic, dan bertanggung jawab atas tindakan dan keputusan yang diambil serta siap menerima segala konsekuensi yang telah ditentukan, dan bertanggung melakukan tindakan-tindakan positif untuk orang lain.	B	D = Kurang
DISIPLIN DAN MANDIRI	Santri mematuhi aturan dan etika yang berlaku, memegang teguh komitmen, dan prinsip-prinsip yang berlaku benar sesuai aturan, dan mampu menyelesaikan dan menajemen aktivitas pribadi dengan baik di asrama.	A	
KREATIF DAN INOVATIF	Santri dapat menunjukkan rasa ingin tahu yang tinggi terhadap dan pengetahuan, dan mampu memberikan gagasan serta berkolaborasi dengan orang lain dan inisiatif, dan mampu mengidentifikasi dan siap permasalah yang ada.		
IKHLAS DAN SEDERHANA	Santri dapat menunjukkan sikap tidak terburu-buru, mandiri, dan disiplin diri, dan lingkungan, rapi dalam berbusana dan meletakkan barang pada tempatnya.	A	
Perbaikan: 0	Keterbatasan: Baik		
Pesan dari Murabbi: Alhamdulillah, semoga Allah SWT memiliki inisiatif yang kuat dalam menjalankan 7 pilar kepemimpinan di asrama, semoga istiqomah dalam menjatinkannya			

Batu, 12 Oktober 2019
Waka Kepengasuhan

Kepala Kepesantrenan
[REDACTED]



B.3 Rapor Nilai Mata Pelajaran





Nama Siswa : ██████████
 Kelas / Semester : X MIA 4 / 1
 No. Induk Sekolah : 0785
 Tahun Pelajaran : 2018-2019

CAPAIAN HASIL BELAJAR

A. SIKAP

A.1. Sikap Spiritual

No.	Uraian	Nilai	Kategori
B.	Selalu bersyukur dan berdzikir sebelum melakukan kegiatan sehari-hari, ketetapan beribadah mulai berkembang. Semoga bisa lebih baik lagi di semester depan.	100	A

A.2. Sikap Sosial

No.	Uraian	Nilai	Kategori
B.	Anda belajar untuk memiliki sikap santun, disiplin, dan tanggung jawab yang baik, responsif dalam pergaulan, sikap kepedulian mulai meningkat. Semoga di karakter selanjutnya hal positif ini mulai meningkat.	100	A

B. PENGETAHUAN DAN KETERAMPILAN

No.	Mata Pelajaran	Nilai 1	Nilai 2	Kategori 1	Kategori 2
Kelompok A (Umum)					
1	Pendidikan Agama dan Budi Pekerti	75	93	A	94
2	Pendidikan Pancasila dan Kewarganegaraan	80	80	C	86
3	Bahasa Indonesia	74	75	C	81
4	Matematika	73	75	C	80
5	Sejarah Indonesia	73	88	B	78
6	Bahasa Inggris	73	91	A	82
Kelompok B (Umum)					
1	Seri Budaya	75	90	B	79
2	Pendidikan Jasmani, Olah Raga dan Kesehatan	75	77	C	81
3	Prakarya dan Kewirausahaan	75	89	B	82
4	Pendidikan Lingkungan Hidup	75	93	A	89
5	Bahasa Arab	75	97	A	95
Kelompok C (Peminatan)					
1	Matematika	75	75	C	75
2	Fisika	75	77	C	78
3	Kimia	75	77	C	81
4	Biologi	75	77	B	80
5	Ekonomi	75	84	B	84
6	Bahasa Arab	75	98	A	98

Pred.	Nilai	Kategori
Sangat Baik	90 - 100	A
Baik	80 - 89	B
Cukup	70 - 79	C
Belum Baik	< 70	D

Ketidakhadiran	Kategori
Sakit	B
Libur	C
Alpa	D