



PENGEMBANGAN PERANGKAT LUNAK PENDAMPINGAN MAHASISWA DISABILITAS (STUDI KASUS : PUSAT STUDI LAYANAN DISABILITAS UNIVERSITAS BRAWIJAYA)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Harun Arrasyid Hasibuan

NIM: 165150201111037



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

2020



PENGESAHAN

PENGEMBANGAN PERANGKAT LUNAK PENDAMPINGAN MAHASISWA
DISABILITAS (STUDI KASUS : PUSAT STUDI LAYANAN DISABILITAS UNIVERSITAS
BRAWIJAYA)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Harun Arrasyid Hasibuan
NIM: 165150201111037

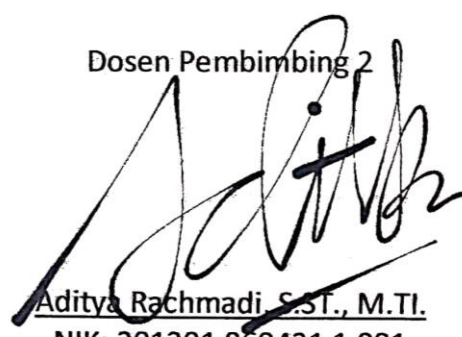
Skrripsi ini telah diuji dan dinyatakan lulus pada
4 Juni 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1

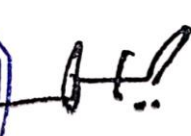

Fajar Pradana, S.ST., M.Eng.
NIP: 19871121 201504 1 004

Dosen Pembimbing 2


Aditya Rachmadi, S.ST., M.TI.
NIK: 201201 860421 1 001

Mengetahui
Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 197105182003121001



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, 12 Mei 2020



Harun Arrasyid Hasibuan

NIM: 165150201111037

PRAKATA

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Pengembangan Perangkat Lunak Pendampingan Mahasiswa Disabilitas (Studi Kasus : Pusat Studi Layanan Disabilitas Universitas Brawijaya)” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Allah SWT yang telah memberi kemudahan dalam semua proses penulisan skripsi ini.
2. Ayahanda dan Ibunda dan seluruh keluarga besar atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini,
3. Bapak Fajar Pradana, S.ST., M.Eng selaku dosen pembimbing I skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
4. Bapak Aditya Rachmadi, S.ST., M.TI selaku dosen pembimbing II skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
5. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D dan Bapak Agus Wahyu Widowo, S.T, M.Cs selaku Ketua Jurusan Teknik Informatika dan Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
6. Pusat Studi dan Layanan Disabilitas Universitas Brawijaya Malang yang bersedia bekerja sama dengan penulis dalam menyediakan data untuk skripsi ini,
7. Seluruh teman-teman dari Fakultas Ilmu Komputer Universitas Brawijaya yang telah membantu baik tenaga dan pikiran selama pengerjaan skripsi ini,

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

Malang, 12 Mei 2020

Penulis

harunhsb@student.ub.ac.id

ABSTRAK

Harun Arrasyid Hasibuan, Pengembangan Perangkat Lunak Pendampingan Mahasiswa Disabilitas (Studi Kasus : Pusat Studi Layanan Disabilitas Universitas Brawijaya)

Pembimbing: Fajar Pradana, S.ST., M.Eng. dan Aditya Rachmadi, S.ST., M.TI.

Pusat Studi dan Layanan Disabilitas Universitas Brawijaya melakukan proses pendampingan mahasiswa disabilitas secara manual setiap harinya. Proses penjadwalan pendampingan mahasiswa disabilitas dilakukan dengan melakukan pencocokan antara jadwal mahasiswa disabilitas dan volunteer serta melakukan verifikasi pendampingan secara manual dengan melihat *logbook* pendampingan yang telah diisi oleh mahasiswa volunteer setiap minggunya sehingga menyebabkan penjadwalan dan verifikasi hasil pendampingan tidak efektif dikarenakan proses yang panjang dan tidak bisa diketahui hasilnya secara langsung. PSDL juga belum memiliki media pelaporan yang dapat digunakan untuk menyalurkan aspirasi maupun keluhan yang dialami oleh mahasiswa sehingga mahasiswa tidak dapat melakukan pelaporan secara langsung dan tidak bisa mengetahui perkembangan pelaporan yang diajukan. Selama ini proses pelaporan hanya dilakukan dengan menghubungi koordinator pendampingan melalui aplikasi *whatsapp* dimana pelaporan yang telah dikirimkan melalui *whatsapp* tersebut harus dicatat kembali pada dokumen *excel* sebagai bahan arsip. Perangkat lunak pendampingan mahasiswa disabilitas adalah sistem yang memiliki fungsi untuk menyelesaikan permasalahan yang dialami oleh Pusat Studi dan Layanan Disabilitas Universitas Brawijaya. Proses penjadwalan, pendampingan, izin pendampingan hingga pelaporan pendampingan dapat dilakukan secara *online*. Sistem dapat melakukan penjadwalan pendampingan secara otomatis berdasarkan jadwal kosong dari mahasiswa, melakukan rekap proses pendampingan yang dijalankan, serta mampu menyediakan fungsi untuk melakukan pelaporan. Sistem dikembangkan melalui tahapan penentuan kebutuhan pengguna, melakukan perancangan, implementasi serta pengujian. Pengujian dilakukan menggunakan pengujian unit, validasi dan *compatibility* dengan hasil 100% valid serta seluruh fungsional dapat berjalan secara baik pada beberapa lingkungan uji.

Kata Kunci : Penjadwalan, pendampingan, *online*, pelaporan.



ABSTRACT

Harun Arrasyid Hasibuan, *Development of Disability Student Assistance Software (Case Study : Center for Disability Studies and Services (CDSS) Universitas Brawijaya)*

Supervisors: Fajar Pradana, S.ST., M.Eng. and Aditya Rachmadi, S.ST., M.TI.

Center for Disability Studies and Services conduct the mentoring process for students with disabilities manually every day. The scheduling process for mentoring students with disabilities is done by matching the disability student and volunteer schedules and verifying mentoring manually by looking at the mentoring logbook that has been filled out by volunteer students every week, which makes scheduling and verification of mentoring results ineffective because of the long and unknowable process the results directly. PSDL also does not yet have a reporting medium that can be used to channel aspirations and complaints experienced by students so that students cannot report directly and cannot know the progress of the proposed report. So far, the reporting process has only been carried out by contacting the assistance coordinator through the whatsapp application where reporting that has been sent via whatsapp must be recorded again in the excel document as archival material. Disability student assistance software is a system that has a function to solve problems experienced by UB's Disability Study and Service Center. The process of scheduling, mentoring, licensing assistance to report assistance can be done online. The system can schedule the mentoring automatically based on the empty schedule of students, recap the mentoring process, and can provide functions for reporting. The system is developed through the stages of determining user needs, carrying out design, implementation and testing. Testing is done using unit testing, validation and compatibility with 100% valid results and all functions can run well in several test environments.

Keywords: Scheduling, assistance, online, reporting.



DAFTAR ISI

PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	Error! Bookmark not defined.
PRAKATA	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xviii
DAFTAR LAMPIRAN	xx
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Disabilitas	6
2.3 Pusat Studi dan Layanan Disabilitas (PSLD)	6
2.4 Pendampingan Mahasiswa Disabilitas	6
2.5 <i>Software Development Life Cycle : Prototyping</i>	7
2.6 <i>Requirement Prototyping</i>	8
2.7 Pendekatan Berbasis Objek	10
2.8 <i>Unified Modelling Language (UML)</i>	11
2.8.1 <i>Use Case</i>	11
2.8.2 <i>Sequence Diagram</i>	12
2.8.3 <i>Class Diagram</i>	14
2.9 Perancangan Perangkat Lunak	15
2.9.1 Perancangan Arsitektur	16



2.9.2 Perancangan Antarmuka.....	17
2.9.3 Perancangan Basis Data.....	17
2.10 Strategi Pengujian Perangkat Lunak.....	17
2.11 Pengujian Perangkat Lunak.....	17
2.11.1 Pengujian Validasi.....	18
2.11.2 Pengujian Unit.....	18
2.11.3 Pengujian Kompatibilitas.....	18
2.12 Teknologi Pengembangan.....	18
2.12.1 <i>Personal Home Page (PHP)</i>	19
2.12.2 <i>Hypertext Markup Language (HTML)</i>	19
2.12.3 <i>Cascading Style Sheet (CSS)</i>	19
2.12.4 <i>Laravel</i>	20
2.12.5 <i>MySQL</i>	20
2.12.6 <i>SortSite</i>	20
2.12.7 <i>PHPUnit</i>	20
2.12.8 <i>Bootstrap</i>	21
BAB 3 METODOLOGI.....	22
3.1 Studi Literatur.....	23
3.2 Landasan Pustaka.....	23
3.3 Analisis Kebutuhan.....	23
3.3.1 <i>Prototype</i>	24
3.3.2 Evaluasi.....	24
3.3.3 Tinjau dan Pembaruan.....	24
3.4 Perancangan.....	24
3.4.1 Perancangan Arsitektur.....	24
3.4.2 Perancangan Antarmuka.....	25
3.4.3 Perancangan Basis Data.....	25
3.5 Implementasi.....	25
3.6 Pengujian.....	25
3.6.1 Pengujian Unit.....	25
3.6.2 Pengujian Validasi.....	26
3.6.3 Pengujian Kompatibilitas.....	26



3.7 Penarikan Kesimpulan dan Saran	26
BAB 4 ANALISIS KEBUTUHAN	27
4.1 Proses Bisnis Pendampingan Mahasiswa Disabilitas	27
4.1.1 Proses Bisnis Pengajuan Pendampingan	27
4.1.2 Proses Bisnis Penjadwalan Pendampingan	28
4.1.3 Proses Bisnis Pelaksanaan Pendampingan	29
4.2 Gambaran Umum Sistem	30
4.3 Identifikasi Aktor	30
4.4 Iterasi Pertama	31
4.4.1 Daftar Kebutuhan Fungsional	31
4.4.2 Daftar Kebutuhan Non Fungsional	37
4.4.3 Pemodelan Kebutuhan	37
4.4.4 Analisis Data	55
4.4.5 Evaluasi	56
4.5 Iterasi Kedua	57
4.5.1 Daftar Kebutuhan Fungsional	57
4.5.2 Pemodelan Kebutuhan	66
4.5.3 Analisis Data	80
4.5.4 Evaluasi	81
BAB 5 PERANCANGAN DAN IMPLEMENTASI	83
5.1 Perancangan Perangkat Lunak	83
5.1.1 Perancangan Arsitektur	83
5.1.2 Perancangan Antarmuka	102
5.2 Implementasi Sistem	108
5.2.1 Spesifikasi Sistem	108
5.2.2 Implementasi Kelas	109
5.2.3 Implementasi Algoritme	112
5.2.4 Implementasi Basis Data	116
5.2.5 Implementasi Antarmuka	117
BAB 6 PENGUJIAN	121
6.1 Pengujian Unit	121
6.1.1 Pengujian Unit pada <i>Method generate_jadwal()</i>	121



6.1.2 Pengujian Unit pada Method konfirmasi_pendampingan()	123
6.1.3 Pengujian Unit pada Method upload_pengajuan_pendampingan_skripsi()	126
6.2 Pengujian Validasi	131
6.2.1 Pengujian Validasi pada Fungsional Login	131
6.2.2 Pengujian Validasi pada Fungsional Login Admin	132
6.2.3 Pengujian Validasi pada Fungsional Hapus Jadwal Pendampingan	133
6.2.4 Pengujian Validasi pada Fungsional Pilih Pendamping Pengganti	134
6.2.5 Pengujian Validasi pada Fungsional Generate Jadwal Pendampingan	134
6.2.6 Pengujian Validasi pada Fungsional Lihat Pengajuan Izin	135
6.2.7 Pengujian Validasi pada Fungsional Lihat Pengajuan Pendampingan Perkuliahan	135
6.2.8 Pengujian Validasi pada Fungsional Lihat Pengajuan Pendampingan Skripsi	136
6.2.9 Pengujian Validasi pada Fungsional Lihat Pengajuan Pendampingan Tugas Akhir	136
6.2.10 Pengujian Validasi pada Fungsional Lihat Jadwal Mahasiswa	137
6.2.11 Pengujian Validasi pada Fungsional Lihat Pendaftaran	137
6.2.12 Pengujian Validasi pada Fungsional Lihat Detail Pendaftaran	137
6.2.13 Pengujian Validasi pada Fungsional Ubah Jadwal Mahasiswa	138
6.2.14 Pengujian Validasi pada Fungsional Logout	139
6.2.15 Pengujian Validasi pada Fungsional Ubah Profil	139
6.2.16 Pengujian Validasi pada Fungsional Ajukan Pendampingan Skripsi	141
6.2.17 Pengujian Validasi pada Fungsional Ajukan Pendampingan Tugas Akhir	142
6.2.18 Pengujian Validasi pada Fungsional Lihat Detail Pengajuan Pendampingan Perkuliahan	144



6.2.19 Pengujian Validasi pada Fungsional Lihat Detail Pengajuan Pendampingan Skripsi	145
6.2.20 Pengujian Validasi pada Fungsional Lihat Detail Pengajuan Pendampingan Tugas Akhir	145
6.2.21 Pengujian Validasi pada Fungsional Pilih Pendamping Skripsi	145
6.2.22 Pengujian Validasi pada Fungsional Pilih Pendamping Tugas Akhir	146
6.2.23 Pengujian Validasi pada Fungsional Terima Pengajuan Pendampingan Perkuliahan	147
6.2.24 Pengujian Validasi pada Fungsional Lihat Pelaporan	147
6.2.25 Pengujian Validasi pada Fungsional Lihat Detail Pelaporan	148
6.2.26 Pengujian Validasi pada Fungsional Balas Laporan	148
6.2.27 Pengujian Validasi pada Fungsional Tulis Pelaporan	149
6.2.28 Pengujian Validasi pada Fungsional Lihat Detail Pendampingan Skripsi	151
6.2.29 Pengujian Validasi pada Fungsional Lihat Detail Pendampingan Tugas Akhir	151
6.2.30 Pengujian Validasi pada Fungsional Lihat Jadwal	152
6.2.31 Pengujian Validasi pada Fungsional Lihat Detail Jadwal	152
6.2.32 Pengujian Validasi pada Fungsional Tambah Jadwal	152
6.2.33 Pengujian Validasi pada Fungsional Hapus Jadwal	154
6.2.34 Pengujian Validasi pada Fungsional Lihat Jadwal Pendampingan	154
6.2.35 Pengujian Validasi pada Fungsional Lihat Detail Jadwal Pendampingan	154
6.2.36 Pengujian Validasi pada Fungsional Mulai Pendampingan	155
6.2.37 Pengujian Validasi pada Fungsional Tampilkan QR Code	155
6.2.38 Pengujian Validasi pada Fungsional Isi <i>Logbook</i> Pendampingan Skripsi	156
6.2.39 Pengujian Validasi pada Fungsional Isi <i>Logbook</i> Pendampingan Tugas Akhir	157
6.2.40 Pengujian Validasi pada Fungsional Konfirmasi <i>Logbook</i> Pendampingan Skripsi	158

6.2.41 Pengujian Validasi pada Fungsional Konfirmasi Logbook	
Pendampingan Tugas Akhir.....	159
6.3 Pengujian Kompatibilitas	159
BAB 7 PENUTUP	162
7.1 Kesimpulan.....	162
7.2 Saran	163
DAFTAR REFERENSI	176



DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka	5
Tabel 2.2 Perbandingan Teknik <i>Requirement Validation</i>	9
Tabel 2.3 Notasi <i>Use Case</i>	11
Tabel 2.4 Notasi <i>Sequence Diagram</i>	13
Tabel 2.5 Notasi <i>Class Diagram</i>	14
Tabel 4.1 Identifikasi Aktor	30
Tabel 4.2 Kebutuhan Fungsional <i>Guest</i>	32
Tabel 4.3 Kebutuhan Fungsional <i>Admin</i>	32
Tabel 4.4 Kebutuhan Fungsional Mahasiswa Disabilitas	33
Tabel 4.5 Kebutuhan Fungsional Mahasiswa <i>Volunteer</i>	34
Tabel 4.6 Spesifikasi Kebutuhan	35
Tabel 4.7 Kebutuhan Non Fungsional	37
Tabel 4.8 <i>Use Case Scenario Login</i>	39
Tabel 4.9 <i>Use Case Scenario Register</i>	39
Tabel 4.10 <i>Use Case Scenario</i> Lihat Pendaftaran	40
Tabel 4.11 <i>Use Case Scenario</i> Lihat Detail Pendaftaran	41
Tabel 4.12 <i>Use Case Scenario</i> Terima Pendaftaran	41
Tabel 4.13 <i>Use Case Scenario</i> Tambah Jadwal Pendampingan	42
Tabel 4.14 <i>Use Case Scenario</i> Hapus Jadwal Pendampingan	43
Tabel 4.15 <i>Use Case Scenario</i> Lihat Pengajuan Izin	43
Tabel 4.16 <i>Use Case Scenario</i> Lihat Pengajuan Pendampingan Skripsi	44
Tabel 4.17 <i>Use Case Scenario</i> Lihat Pengajuan Pendampingan	44
Tabel 4.18 <i>Use Case Scenario</i> Pilih Pendamping Skripsi	45
Tabel 4.19 <i>Use Case Scenario</i> Lihat Jadwal Mahasiswa	46
Tabel 4.20 <i>Use Case Scenario</i> Ubah Jadwal Mahasiswa	46
Tabel 4.21 <i>Use Case Scenario</i> Tambah Jadwal	47
Tabel 4.22 <i>Use Case Scenario</i> Hapus Jadwal	48
Tabel 4.23 <i>Use Case Scenario</i> Lihat Pendampingan Perkuliahan	49
Tabel 4.24 <i>Use Case Scenario</i> Lihat Detail Jadwal Pendampingan	49
Tabel 4.25 <i>Use Case</i> Ubah Profil	50



Tabel 4.26 <i>Use Case</i> Izin Pendampingan.....	50
Tabel 4.27 <i>Use Case Scenario</i> Ajukan Pendampingan Skripsi.....	51
Tabel 4.28 <i>Use Case Scenario</i> Lihat Detail Pendampingan Skripsi.....	52
Tabel 4.29 <i>Use Case Scenario</i> Tampilkan Qr Code.....	53
Tabel 4.30 <i>Use Case Scenario</i> Mulai Pendampingan.....	53
Tabel 4.31 <i>Use Case Scenario</i> Logout	54
Tabel 4.32 Kebutuhan Fungsional Iterasi Kedua <i>Admin</i>	58
Tabel 4.33 Kebutuhan Fungsional Iterasi Kedua Mahasiswa Disabilitas.....	60
Tabel 4.34 Kebutuhan Fungsional Iterasi Kedua Mahasiswa <i>Volunteer</i>	62
Tabel 4.35 Spesifikasi Kebutuhan	63
Tabel 4.36 <i>Use Case Scenario</i> Login <i>Admin</i>	68
Tabel 4.37 <i>Use Case Scenario</i> Generate Jadwal Pendampingan.....	69
Tabel 4.38 <i>Use Case Scenario</i> Lihat Pengajuan Pendampingan Tugas Akhir.....	70
Tabel 4.39 <i>Use Case Scenario</i> Lihat Detail Pengajuan Pendampingan Tugas Akhir.....	70
Tabel 4.40 <i>Use Case Scenario</i> Pilih Pendamping Pengganti	71
Tabel 4.41 <i>Use Case Scenario</i> Lihat Pelaporan.....	72
Tabel 4.42 <i>Use Case Scenario</i> Lihat Detail Pelaporan.....	72
Tabel 4.43 <i>Use Case Scenario</i> Balas Laporan.....	73
Tabel 4.44 <i>Use Case Scenario</i> Tulis Laporan	74
Tabel 4.45 <i>Use Case Scenario</i> Ajukan Pendampingan Tugas Akhir	74
Tabel 4.46 <i>Use Case Scenario</i> Lihat Detail Pendampingan Tugas Akhir.....	75
Tabel 4.47 <i>Use Case Scenario</i> Konfirmasi <i>Logbook</i>	76
Tabel 4.48 <i>Use Case Scenario</i> Ajukan <i>Logbook</i>	76
Tabel 4.49 <i>Use Case Scenario</i> Pilih Pendamping Tugas Akhir.....	77
Tabel 4.50 <i>Use Case Scenario</i> Lihat Pengajuan Pendampingan Perkuliahan.....	78
Tabel 4.51 <i>Use Case Scenario</i> Lihat Detail Pengajuan Pendampingan Perkuliahan.....	79
Tabel 4.52 <i>Use Case Scenario</i> Terima Pengajuan Pendampingan Perkuliahan.....	79
Tabel 4.53 <i>Use Case Scenario</i> Lihat Jadwal.....	80
Tabel 5.1 <i>Pseudocode</i> Pengajuan Pendampingan Skripsi.....	100
Tabel 5.2 <i>Pseudocode</i> Generate Jadwal Pendampingan	101



Tabel 5.3 <i>Pseudocode</i> Mulai Pendampingan	102
Tabel 5.4 Komponen Perancangan Antarmuka Pengajuan Pendampingan Skripsi	103
Tabel 5.5 Komponen Perancangan Antarmuka Generate Jadwal Pendampingan	105
Tabel 5.6 Komponen Perancangan Antarmuka Mulai Pendampingan	107
Tabel 5.7 Spesifikasi Perangkat Lunak	108
Tabel 5.8 Spesifikasi Perangkat Keras	109
Tabel 5.9 Implementasi Kelas	109
Tabel 5.10 Kode Program <i>Method</i> generate_jadwal()	112
Tabel 5.11 Kode Program <i>Method</i> konfirmasi_pendampingan()	113
Tabel 5.12 Kode Program <i>Method</i> upload_pengajuan_pendampingan_skripsi()	114
Tabel 6.1 <i>Pseudocode method</i> generate_jadwal()	121
Tabel 6.2 <i>Test Case Method</i> generate_jadwal()	123
Tabel 6.3 <i>Pseudocode method</i> konfirmasi_pendampingan()	123
Tabel 6.4 <i>Test Case Method</i> konfirmasi_pendampingan()	126
Tabel 6.5 <i>Pseudocode method</i> upload_pengajuan_pendampingan_skripsi()	127
Tabel 6.6 <i>Test Case Method</i> upload_pengajuan_pendampingan_skripsi()	128
Tabel 6.7 Skenario Uji Validasi Fungsional Login	131
Tabel 6.8 Skenario Uji Validasi Fungsional Login Admin	133
Tabel 6.9 Skenario Uji Validasi Fungsional Hapus Jadwal Pendampingan	134
Tabel 6.10 Skenario Uji Validasi Fungsional Pilih Pendamping Pengganti	134
Tabel 6.11 Skenario Uji Validasi Fungsional Generate Jadwal Pendampingan	134
Tabel 6.12 Skenario Uji Validasi Fungsional Lihat Pengajuan Izin	135
Tabel 6.13 Skenario Uji Validasi Fungsional Lihat Pengajuan Pendampingan Perkuliahan	135
Tabel 6.14 Skenario Uji Validasi Fungsional Lihat Pengajuan Pendampingan Skripsi	136
Tabel 6.15 Skenario Uji Validasi Fungsional Lihat Pengajuan Pendampingan	136
Tabel 6.16 Skenario Uji Validasi Fungsional Lihat Jadwal Mahasiswa	137
Tabel 6.17 Skenario Uji Validasi Fungsional Lihat Pendaftaran	137
Tabel 6.18 Skenario Uji Validasi Fungsional Lihat Detail Pendaftaran	137



Tabel 6.19 Skenario Uji Validasi Fungsional Lihat Jadwal Mahasiswa.....	138
Tabel 6.20 Skenario Uji Validasi Fungsional Logout.....	139
Tabel 6.21 Skenario Uji Validasi Fungsional Ubah Profil.....	139
Tabel 6.22 Skenario Uji Validasi Fungsional Ajukan Pendampingan Skripsi.....	141
Tabel 6.23 Skenario Uji Validasi Fungsional Ajukan Pendampingan Tugas Akhir.....	143
Tabel 6.24 Skenario Uji Validasi Fungsional Lihat Detail Pengajuan Pendampingan Perkuliahan	144
Tabel 6.25 Skenario Uji Validasi Fungsional Lihat Detail Pengajuan Pendampingan Skripsi	145
Tabel 6.26 Skenario Uji Validasi Fungsional Tambah Jadwal.....	145
Tabel 6.27 Skenario Uji Validasi Fungsional Pilih Pendamping Skripsi	146
Tabel 6.28 Skenario Uji Validasi Fungsional Pilih Pendamping Tugas Akhir.....	146
Tabel 6.29 Skenario Uji Validasi Fungsional Terima Pengajuan Pendampingan Perkuliahan	147
Tabel 6.30 Skenario Uji Validasi Fungsional Lihat Pelaporan.....	147
Tabel 6.31 Skenario Uji Validasi Fungsional Lihat Detail Pelaporan.....	148
Tabel 6.32 Skenario Uji Validasi Fungsional Balas Laporan	148
Tabel 6.33 Skenario Uji Validasi Fungsional Tulis Laporan	150
Tabel 6.34 Skenario Uji Validasi Fungsional Lihat Detail Pendampingan Skripsi.....	151
Tabel 6.35 Skenario Uji Validasi Fungsional Lihat Detail Pendampingan Tugas Akhir.....	151
Tabel 6.36 Skenario Uji Validasi Fungsional Lihat Jadwal.....	152
Tabel 6.37 Skenario Uji Validasi Fungsional Lihat Jadwal.....	152
Tabel 6.38 Skenario Uji Validasi Fungsional Tambah Jadwal.....	153
Tabel 6.39 Skenario Uji Validasi Fungsional Hapus Jadwal.....	154
Tabel 6.40 Skenario Uji Validasi Fungsional Lihat Jadwal Pendampingan.....	154
Tabel 6.41 Skenario Uji Validasi Fungsional Lihat Detail Jadwal Pendampingan.....	154
Tabel 6.42 Skenario Uji Validasi Fungsional Mulai Pendampingan	155
Tabel 6.43 Skenario Uji Validasi Fungsional Tampilkan QR Code	155
Tabel 6.44 Skenario Uji Validasi Fungsional Isi <i>Logbook</i> Pendampingan Skripsi.....	156
Tabel 6.45 Skenario Uji Validasi Fungsional Isi <i>Logbook</i> Pendampingan Tugas Akhir	157



Tabel 6.46 Skenario Uji Validasi Fungsional Konfirmasi *Logbook* Pendampingan Skripsi.....158

Tabel 6.47 Skenario Uji Validasi Fungsional Konfirmasi *Logbook* Pendampingan Tugas Akhir.....159

Tabel 6.48 Parameter Kompatibilitas pada Sortsite.....160



DAFTAR GAMBAR

Gambar 2.1 Proses Pengembangan <i>Prototype</i>	8
Gambar 2.2 Grafik Penyebab Kesalahan pada Pengembangan Perangkat Lunak..	9
Gambar 2.3 Model Umum dari Proses Perancangan Perangkat Lunak.....	16
Gambar 3.1 Diagram Metodologi Penelitian	22
Gambar 4.1 Bisnis Proses Pengajuan Pendampingan.....	28
Gambar 4.2 Bisnis Proses Penjadwalan Pendampingan.....	29
Gambar 4.3 Bisnis Proses Pelaksanaan Pendampingan.....	30
Gambar 4.4 <i>Use Case Diagram</i> Perangkat Lunak Pendampingan Mahasiswa Disabilitas.....	38
Gambar 4.5 <i>Entity Relationship Diagram</i> Iterasi Pertama Perangkat Lunak Pendampingan Mahasiswa Disabilitas.....	55
Gambar 4.6 <i>Use Case Diagram</i> Iterasi Kedua Perangkat Lunak Pendampingan Mahasiswa Disabilitas.....	67
Gambar 4.7 <i>Entity Relationship Diagram</i> Iterasi Kedua Perangkat Lunak Pendampingan Mahasiswa Disabilitas.....	81
Gambar 5.1 <i>Sequence Diagram</i> Mulai Pendampingan.....	84
Gambar 5.2 <i>Sequence Diagram</i> Generate Jadwal Pendampingan.....	86
Gambar 5.3 <i>Sequence Diagram</i> Pengajuan Pendampingan Skripsi.....	87
Gambar 5.4 <i>Class Diagram</i> Perangkat Lunak Pendampingan Mahasiswa Disabilitas.....	88
Gambar 5.5 <i>Class</i> PendampinganController.....	89
Gambar 5.6 <i>Class</i> PenjadwalanController.....	90
Gambar 5.7 <i>Class</i> TutorController.....	91
Gambar 5.8 <i>Class</i> LoginController.....	91
Gambar 5.9 <i>Class</i> RegisterController.....	92
Gambar 5.10 <i>Class</i> HomeController.....	92
Gambar 5.11 <i>Class</i> JadwalController.....	93
Gambar 5.12 <i>Class</i> ProfilController.....	94
Gambar 5.13 <i>Class</i> AdminLoginController.....	94
Gambar 5.14 <i>Class</i> PendaftaranController.....	95
Gambar 5.15 <i>Class</i> PelaporanController.....	95



Gambar 5.16 <i>Class</i> Jadwal_pendampingan	96
Gambar 5.17 <i>Class</i> Pendampingan_skripsi	96
Gambar 5.18 <i>Class</i> User	97
Gambar 5.19 <i>Class</i> Jadwal	97
Gambar 5.20 <i>Class</i> Logbook_pendampingan_skripsi	98
Gambar 5.21 <i>Class</i> Logbook_pendampingan_tugas_akhir	98
Gambar 5.22 <i>Class</i> Pendampingan_tugas_akhir	98
Gambar 5.23 <i>Class</i> Pelaporan	99
Gambar 5.24 <i>Class</i> Respon_laporan	99
Gambar 5.25 <i>Class</i> UsersAdmin	99
Gambar 5.26 <i>Class</i> Pendampingan_perkuliah	100
Gambar 5.27 <i>Class</i> Data_fakulta	100
Gambar 5.28 Perancangan Antarmuka Pengajuan Pendampingan Skripsi	103
Gambar 5.29 Perancangan Antarmuka Generate Jadwal Pendampingan	105
Gambar 5.30 Perancangan Antarmuka Mulai Pendampingan	107
Gambar 5.31 Implementasi Basis Data	117
Gambar 5.32 Implementasi Antarmuka Halaman Detail Pendampingan	118
Gambar 5.33 Implementasi Antarmuka Halaman Pengajuan Pendampingan Skripsi	119
Gambar 5.34 Implementasi Antarmuka Generate Jadwal	120
Gambar 6.1 <i>Basic Path Pseudocode</i> generate_jadwal()	122
Gambar 6.2 <i>Basic Path Pseudocode</i> konfirmasi_pendampingan()	125
Gambar 6.3 <i>Basic Path Pseudocode</i> upload_pengajuan_pendampingan_skripsi()	128
Gambar 6.4 Hasil Pengujian Kompatibilitas dari Perangkat Lunak Pendampingan Mahasiswa Disabilitas	160



DAFTAR LAMPIRAN

LAMPIRAN A Surat Kelayakan Etik / Izin	164
LAMPIRAN B Wawancara.....	165
B.1 Wawancara Mekanisme Pendampingan.....	165
B.2 Wawancara Mekanisme Pendampingan Skripsi, Tugas Akhir Serta Pelaporan.....	167
LAMPIRAN C Formulir Pelaksanaan Pendampingan Mahasiswa Disabilitas.....	170
C.1 Formulir Pengajuan Pendampingan/Tutor.....	170
C.2 Logbook Pendampingan/Tutor Skripsi/Tugas Akhir.....	172
LAMPIRAN D Pengujian Unit Menggunakan Library PHPUNIT.....	173
D.1 Pengujian Unit Method generate_jadwal() Jalur 1.....	173
D.2 Pengujian Unit Method generate_jadwal() Jalur 2.....	173
D.3 Pengujian Unit Method konfirmasi_pendampingan() Jalur 1.....	173
D.4 Pengujian Unit Method konfirmasi_pendampingan() Jalur 2.....	174
D.5 Pengujian Unit Method konfirmasi_pendampingan() Jalur 3.....	174
D.6 Pengujian Unit Method upload_pengajuan_pendampingan_skripsi() Jalur 1.....	175
D.7 Pengujian Unit Method upload_pengajuan_pendampingan_skripsi() Jalur 2.....	175
D.8 Pengujian Unit Method upload_pengajuan_pendampingan_skripsi() Jalur 3.....	175



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Disabilitas merupakan kondisi adanya keterbatasan fisik dan fungsi fisik ataupun mental yang menyebabkan keterbatasan bagi penyandanginya dalam jangka waktu yang lama (Riadi, 2018). Kondisi disabilitas ini dapat dirasakan oleh siapa saja termasuk oleh mahasiswa terutama mahasiswa di Universitas Brawijaya. Untuk mengakomodir kondisi disabilitas ini, maka didirikanlah Pusat Studi dan Layanan Disabilitas Universitas Brawijaya (PSLD UB). PSLD UB adalah lembaga yang berdiri di lingkungan Universitas Brawijaya. PSLD UB memiliki fungsi untuk pelayanan penyandang difabel dan pusat studi untuk isu disabilitas di lingkungan Universitas Brawijaya. Dari sekian banyak pelayanan yang diberikan oleh PSLD UB, terdapat suatu layanan penting yang diberikan oleh PSLD UB kepada mahasiswa penyandang disabilitas, yaitu pendampingan.

Pendampingan adalah kegiatan dimana mahasiswa Universitas Brawijaya yang berstatus sebagai penyandang disabilitas diberikan bantuan berupa pendampingan oleh mahasiswa yang berstatus sebagai volunteer. Kegiatan pendampingan ini dilaksanakan dengan tujuan untuk membantu aktivitas yang dilakukan oleh mahasiswa disabilitas di lingkungan Universitas Brawijaya seperti pengerjaan tugas kuliah, pengurusan administrasi perkuliahan, dan memberikan bantuan pendampingan kepada mahasiswa disabilitas dalam melaksanakan kegiatan perkuliahan sehari-hari. Manajemen pendampingan terutama pada penjadwalan pendampingan perlu diperhatikan untuk memastikan kualitas dari proses pendampingan dan menunjang proses perkuliahan dari mahasiswa disabilitas. Pada pelaksanaan kegiatan pendampingan mahasiswa disabilitas, terdapat beberapa permasalahan yang dikemukakan pada saat wawancara yang dilakukan bersama koordinator pendampingan. Permasalahan terdiri dari proses penjadwalan, pendampingan, pengajuan izin hingga pelaporan.

Permasalahan pertama adalah manajemen penjadwalan pendampingan masih dilakukan menggunakan aplikasi Microsoft Excel. Proses penjadwalan pendampingan menggunakan Microsoft excel berdampak terhadap waktu yang dibutuhkan untuk melakukan penjadwalan karena koordinator pendampingan harus melakukan pencocokan jadwal kuliah mahasiswa disabilitas dengan waktu kosong yang dimiliki oleh mahasiswa volunteer.

Permasalahan kedua adalah proses pendampingan masih dilakukan secara manual dengan mengisi lembar *logbook* sehingga berdampak pada proses verifikasi pendampingan yang tidak bisa dilakukan secara langsung. Selain itu, proses pengajuan izin juga dilakukan secara manual dengan menghubungi koordinator pendampingan melalui *whatsapp* maksimal satu hari sebelum dilaksanakannya pendampingan. PSLD UB juga belum memiliki media pelaporan yang ditujukan untuk mahasiswa disabilitas maupun volunteer untuk menyampaikan aspirasi serta keluhan selama melakukan proses pendampingan.



Berdasarkan permasalahan tersebut, akan dilakukan penelitian untuk mengembangkan sistem pendampingan yang mampu untuk digunakan mengakomodir manajemen pendampingan dan kegiatan pendampingan yang dilakukan. Untuk menyelesaikan permasalahan ini, akan dikembangkan perangkat lunak pendampingan mahasiswa disabilitas berbasis *responsive website*. *Responsive website* adalah teknik yang digunakan untuk membangun suatu *website* yang mampu secara mandiri melakukan adaptasi pada setiap perangkat dan mampu untuk melakukan pengaturan ulang dari sisi bentuk secara mandiri berdasarkan ukuran dari layar suatu perangkat (Giurgiu & Gligorea, 2017). Pengembangan perangkat lunak pendampingan mahasiswa disabilitas berbasis *responsive website* bertujuan agar perangkat lunak yang dikembangkan dapat dijalankan pada seluruh perangkat tanpa terbatas dari *platform* yang digunakan. Penelitian ini diharapkan mampu menentukan kebutuhan pendampingan mahasiswa disabilitas, mampu merancang, melakukan implementasi serta pengujian terhadap perangkat lunak pendampingan mahasiswa disabilitas dengan harapan mampu mengatasi permasalahan pendampingan mahasiswa disabilitas di Universitas Brawijaya.

1.2 Rumusan Masalah

Dari penjelasan yang telah di sampaikan, dapat disimpulkan tema masalah yang akan diangkat yaitu apakah aplikasi yang bertujuan untuk pendampingan mahasiswa disabilitas bisa dikembangkan? Maka dari itu, dibuatlah beberapa spesifikasi rumusan masalah untuk menjawab rumusan masalah umum yang telah dijabarkan diatas yang terdiri dari:

1. Bagaimana hasil yang diperoleh di fase analisis spesifikasi kebutuhan dari pengembangan perangkat lunak pendampingan mahasiswa disabilitas?
2. Bagaimana hasil dari fase perancangan yang dilakukan dalam pengembangan perangkat lunak pendampingan mahasiswa disabilitas?
3. Bagaimana hasil dari implementasi yang dilakukan pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas?
4. Bagaimana hasil dari fase pengujian yang dilakukan dalam pengembangan perangkat lunak pendampingan mahasiswa disabilitas?

1.3 Tujuan

Penelitian ini ditujukan untuk:

1. Mengetahui kebutuhan dari mahasiswa penyandang disabilitas dan kebutuhan dari mahasiswa *volunteer* yang akan diimplementasikan pada perangkat lunak pendampingan mahasiswa disabilitas.
2. Mampu merancang aplikasi yang mampu membantu mahasiswa disabilitas dan mahasiswa *volunteer* dalam melakukan pendampingan



3. Mampu mengimplementasikan rancangan aplikasi yang telah di buat menjadi sebuah perangkat lunak yang dapat digunakan oleh mahasiswa disabilitas maupun mahasiswa *volunteer*.
4. Mengetahui hasil dari pengujian perangkat lunak terhadap perangkat lunak pendampingan mahasiswa disabilitas.

1.4 Manfaat

Adapun manfaat dari penelitian ini adalah sebagai berikut:

- Mengembangkan kemampuan dan *skill* dalam bidang pemrograman website serta kemampuan dalam menganalisis, implementasi, dan pengujian suatu perangkat lunak.
- Menjadi solusi untuk membantu proses pendampingan mahasiswa disabilitas yang dilakukan mahasiswa *volunteer* khususnya di lingkungan Universitas Brawijaya.
- Sebagai bahan edukasi untuk masyarakat mengenai pentingnya kesadaran akan pendampingan kepada mahasiswa disabilitas dalam melaksanakan kegiatan sehari – hari terutama di bidang akademik.

1.5 Batasan Masalah

Dari hasil penjabaran rumusan masalah, maka penelitian ini hanya dilakukan dalam batas berikut :

1. Penelitian dilakukan pada mahasiswa Universitas Brawijaya dengan status penyandang disabilitas.
2. Penelitian berfokus kepada studi kasus dari Pusat Layanan Studi Disabilitas Universitas Brawijaya.
3. Dilakukan impementasi apilkasi berbasis *responsive web* untuk perangkat lunak pendampingan mahasiswa disabilitas.

1.6 Sistematika Pembahasan

Penelitian ini akan disusun dengan acuan seperti berikut:

1. BAB 1 – PENDAHULUAN

Bab pendahuluan akan menjabarkan mengenai latar belakang, rumusan masalah, tujuan, manfaat dan batasan masalah.

2. BAB 2 – LANDASAN KEPUSTAKAAN

Bab landasan kepastakaan akan menjabarkan mengenai tinjauan pustaka serta dasar teori yang menjadi acuan untuk membantu penyelesaian permasalahan yang dikemukakan.

3. BAB 3 – METODOLOGI



Bab metodologi akan menjabarkan mengenai metode yang diimplementasikan untuk menyelesaikan permasalahan yang dikemukakan.

4. BAB 4 – ANALISIS KEBUTUHAN

Bab analisis kebutuhan akan menjabarkan proses analisis untuk mendapatkan daftar kebutuhan perangkat lunak yang menjadi acuan pada proses selanjutnya.

5. BAB 5 – PERANCANGAN DAN IMPLEMENTASI

Bab perancangan dan implementasi akan menjabarkan proses perancangan yang dilakukan dengan mengacu pada daftar kebutuhan yang didapat dari fase analisis kebutuhan serta proses implementasi yang dilakukan dengan mengacu pada perancangan yang telah dilakukan.

6. BAB 6 – PENGUJIAN

Bab pengujian menjabarkan proses pengujian yang dilakukan terhadap perangkat lunak yang telah dikembangkan pada fase implentasi.

7. BAB 8 – PENUTUP

Bab penutup menjabarkan kesimpulan yang didapat dari proses penelitian yang telah dikerjakan serta saran yang dapat diberikan berkaitan dengan proses pengembangan.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dibahas mengenai tinjauan pustaka dan dasar teori yang digunakan dalam melakukan penelitian ini. Tinjauan pustaka yang digunakan adalah penelitian – penelitian sebelumnya yang dapat dijadikan referensi dalam pengembangan perangkat lunak pendampingan mahasiswa disabilitas.

2.1 Tinjauan Pustaka

Pengembangan perangkat lunak pendampingan mahasiswa disabilitas dilakukan dengan melakukan tinjauan kepada beberapa pustaka. Tinjauan pustaka diperlukan sebagai pertimbangan serta menjadi landasan dalam melakukan penelitian dengan melihat dari penelitian – penelitian serupa yang telah dilakukan. Pada penelitian ini, dilakukan tinjauan pada dua pustaka yaitu pembuatan model kebutuhan pengguna untuk universitas dan pemodelan rancangan penjadwalan mata kuliah *International Program* di Unikom. Dari kedua pustaka ini didapatkan kesimpulan jika proses pengembangan yang dilakukan pada kedua penelitian tersebut sama – sama menggunakan metode *prototype*. Kesimpulan ini menjadi landasan dalam melakukan pengembangan perangkat lunak pendampingan mahasiswa disabilitas menggunakan metode *prototype*. Landasan pustaka dapat tersebut dapat dilihat pada Tabel 2.1

Tabel 2.1 Tinjauan Pustaka

Judul	Penulis	Tahun	Metode	Hasil
Implementasi Metode Prototipe Pada Website Penelusuran Minat Peserta Didik Sebagai Layanan Bimbingan Konseling	- Febria Sri Handayani - Meidyan Permata Putri	2018	Pengembangan perangkat lunak <i>Prototype</i>	Penggunaan teknik <i>prototyping</i> digunakan untuk mendapatkan umpan balik sehingga dapat digunakan untuk memperbaiki spesifikasi kebutuhan (Handayani & Putri, 2018).
Pemodelan Rancangan Proses Penjadwalan Mata Kuliah Di International Program Kedalam Sistem	- Andri Sahata Sitanggang	2016	Pengembangan perangkat lunak <i>Prototype</i>	Pendekatan <i>prototyping</i> digunakan dengan alasan dapat menggambarkan sistem secara berulang – ulang



Judul	Penulis	Tahun	Metode	Hasil
Informasi Unikom Berbasis Android				sehingga dapat membantu memberikan gambaran pada setiap revisinya (Sitanggang, 2016).

2.2 Disabilitas

Disabilitas atau difabel adalah suatu keadaan dimana terdapat suatu keterbatasan fisik maupun fungsi fisik serta mental yang menyebabkan terbatasnya suatu aktifitas tertentu. Disabilitas berasal dari kata different ability yang memiliki makna perbedaan kemampuan. Disabilitas memiliki makna “Penyandang disabilitas adalah setiap orang yang mengalami keterbatasan fisik, intelektual, mental, dan atau sensorik dalam jangka waktu lama yang dalam berinteraksi dengan lingkungan dapat mengalami hambatan dan kesulitan untuk berpartisipasi secara penuh dan efektif dengan warga Negara lainnya berdasarkan kesamaan hak” (Sekretariat Kementrian Negara, 2015). Pada umumnya, terdapat beberapa jenis disabilitas yaitu cacat fisik, cacat mental dan cacat fisik maupun mental.

2.3 Pusat Studi dan Layanan Disabilitas (PSLD)

Pusat Studi dan Layanan Disabilitas (PSLD) Universitas Brawijaya merupakan lembaga yang didirikan oleh Universitas Brawijaya pada tahun 2012. Pendirian lembaga PSLD dilatarbelakangi atas kesulitan yang dialami penyandang disabilitas yang disebabkan sarana dan prasarana yang sulit diakses mahasiswa disabilitas dan pengetahuan Universitas Brawijaya akan isu – isu disabilitas yang masih sedikit. PSLD didirikan sebagai wadah pemberian layanan serta penelitian terhadap pelayanan serta isu disabilitas. PSLD didirikan untuk memfasilitasi kegiatan mahasiswa disabilitas di lingkungan Universitas Brawijaya sehingga mahasiswa disabilitas mampu membaur dengan keadaan di lingkungan Universitas Brawijaya (“Sejarah PSLD,” 2013).

2.4 Pendampingan Mahasiswa Disabilitas

Pendampingan mahasiswa disabilitas merupakan kegiatan yang diselenggarakan oleh Pusat Studi dan Layanan Disabilitas (PSLD) Universitas Brawijaya. Pendampingan mahasiswa disabilitas diperlukan untuk memberikan bantuan kepada mahasiswa disabilitas dalam menjalankan kegiatan akademik di Universitas Brawijaya. Pendampingan dilakukan dengan bantuan mahasiswa *volunteer* yang bertindak sebagai pihak yang melakukan pendampingan kepada mahasiswa disabilitas di lapangan.

Pendampingan mahasiswa disabilitas diawali dari proses pendaftaran mahasiswa disabilitas. Proses pendaftaran terbagi dalam dua tahap yaitu melalui



data mahasiswa yang didapatkan dari Seleksi Program Khusus Penyandang Disabilitas (SPKPD) dan melalui proses pengajuan. Pendaftaran mahasiswa disabilitas melalui tahap pengajuan diawali dengan pengisian formulir pengajuan dan melengkapi formulir dengan tanda tangan. Proses pengajuan dilanjutkan dengan menyerahkan formulir yang telah diisi ke koordinator pendampingan. Koordinator pendampingan akan melakukan konfirmasi pada pengajuan yang telah dilakukan oleh mahasiswa disabilitas.

Mahasiswa disabilitas yang telah dikonfirmasi oleh PSLD diwajibkan untuk menyerahkan jadwal kegiatannya, yaitu jadwal perkuliahan untuk pengajuan pendampingan perkuliahan dan jadwal tugas akhir / skripsi untuk pengajuan pendampingan tugas akhir / skripsi. PSLD akan melakukan rekap pada jadwal yang telah diberikan oleh mahasiswa disabilitas dan melakukan penentuan jadwal pendampingan terhadap pendampingan perkuliahan serta menentukan pendamping terhadap pendampingan tugas akhir / skripsi. PSLD akan memberikan hasil penjadwalan dan pemilihan pendamping kepada mahasiswa melalui grup *whatsapp* yang telah disediakan sebelumnya. Mahasiswa disabilitas maupun *volunteer* diwajibkan untuk melakukan pemeriksaan terhadap hasil penjadwalan serta melakukan konfirmasi apabila terdapat perubahan pada jadwal yang dimiliki untuk dilakukan penyesuaian penjadwalan. Setelah jadwal didapatkan, proses pendampingan siap dilaksanakan.

Pelaksanaan pendampingan didasarkan pada jadwal pendampingan dan pendamping yang telah ditentukan oleh PSLD. Pendampingan perkuliahan diawali dengan mahasiswa *volunteer* menghubungi mahasiswa disabilitas untuk melakukan koordinasi terkait dengan pelaksanaan pendampingan seperti titik pertemuan serta alat bantu yang dibutuhkan jika ada. Mahasiswa *volunteer* akan menemui serta melaksanakan pendampingan sesuai dengan kesepakatan dengan mahasiswa disabilitas. Setelah proses pendampingan selesai, mahasiswa *volunteer* mengisi *logbook* pendampingan serta meminta tanda tangan mahasiswa disabilitas sebagai konfirmasi hasil pendampingan yang telah dilakukan. Pada pendampingan tugas akhir / skripsi, pendampingan diwajibkan dilakukan di lingkungan kantor PSLD sebagai syarat yang telah ditentukan PSLD. Pendampingan tugas akhir / skripsi dilakukan di lingkungan kantor PSLD untuk memastikan proses pendampingan berjalan dengan baik dikarenakan jadwal pendampingan yang bersifat fleksibel bergantung antara kesepakatan mahasiswa disabilitas dengan *volunteer*.

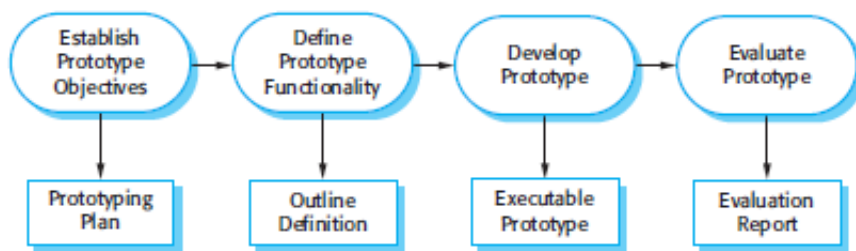
2.5 Software Development Life Cycle : Prototyping

Prototyping adalah *software development life cycle* yang menawarkan pembuatan purwarupa sebagai versi permulaan dari sistem. *Prototype* digunakan untuk mendemonstrasikan desain, konsep serta mampu menggambarkan kemungkinan solusi yang dapat ditawarkan. *Prototype* mampu membantu memberikan ide baru dari kebutuhan sistem. *Prototype* dapat digunakan untuk mencari kelemahan, kesalahan serta kekurangan dari suatu spesifikasi kebutuhan sehingga bisa dilakukan perbaikan pada spesifikasi kebutuhan.



Prototyping dapat dikembangkan tanpa membuat aplikasi yang dapat dieksekusi sebagai purwarupa nya. Purwarupa berbasis maket dari antarmuka sistem dapat digunakan sebagai purwarupa yang efektif untuk diserahkan kepada *stakeholder* karena dapat dikembangkan secara cepat dan berbiaya rendah. *Prototyping* dapat digunakan untuk melakukan elisitasi serta validasi kebutuhan pada tahap rekayasa kebutuhan (Sommerville, 2011).

Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas, *Prototyping* akan dilakukan untuk membantu dalam proses elisitasi kebutuhan serta proses validasi hasil dari proses elisitasi pada tahap rekayasa kebutuhan. Rekayasa kebutuhan dari perangkat lunak pendampingan mahasiswa disabilitas diawali dengan melakukan melakukan elisitasi kebutuhan, dilanjutkan dengan mendefinisikan fungsionalitas dari *prototype* yang akan dikembangkan sesuai dengan elisitasi kebutuhan yang telah dilakukan, kemudian dilanjutkan dengan mengembangkan *prototype* dalam bentuk maket dan diakhiri dengan evaluasi dari *prototype* yang telah dibuat. Proses dari pengembangan *prototype* dapat dilihat dalam Gambar 2.1.



Gambar 2.1 Proses Pengembangan *Prototype*

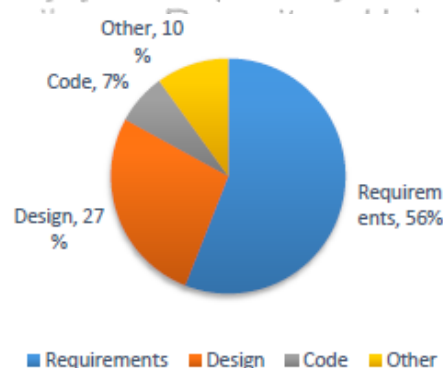
Sumber : (Sommerville, 2011)

2.6 Requirement Prototyping

Dalam pengembangan suatu perangkat lunak, perlu dilakukannya pengumpulan kebutuhan yang didapatkan dari calon pengguna terkait. Pengumpulan kebutuhan diperlukan sebagai syarat awal dalam melakukan spesifikasi kebutuhan. Spesifikasi kebutuhan menjadi dasar dalam penentuan fitur dari suatu perangkat lunak yang akan dikembangkan dimana fitur yang dikembangkan diharapkan dapat menyelesaikan permasalahan pengguna. Melihat pentingnya dari suatu spesifikasi kebutuhan, maka perlu dilakukannya proses validasi untuk memastikan spesifikasi kebutuhan yang telah didefinisikan telah sesuai dan mampu menyelesaikan permasalahan yang dihadapi pengguna. Untuk memastikan spesifikasi kebutuhan telah sesuai dengan permasalahan pengguna, maka akan dilakukan validasi kebutuhan (*requirement validation*). *Requirement validation* memastikan spesifikasi kebutuhan yang telah didefinisikan telah lengkap dan konsisten serta sesuai dengan kebutuhan dari pengguna (Anas, Ilyas, Tariq, & Hummayun, 2016).



Menurut penelitian berjudul “*Requirements Validation Techniques: An Empirical Study*” yang dilakukan oleh Anas dkk.(2016), *requirement validation* diperlukan untuk mengatasi semua konflik yang terjadi pada daftar spesifikasi kebutuhan. *Requirement validation* penting dilakukan mengingat kesalahan yang terdeteksi pada tahap akhir dari pengembangan perangkat lunak yang disebabkan kesalahan pada spesifikasi kebutuhan jauh lebih mahal untuk dilakukan perbaikan. Hal ini disebabkan perbaikan yang dilakukan mulai dari spesifikasi kebutuhan akan berdampak kepada artifak – artifak yang harus menyesuaikan pula dengan perbaikan pada spesifikasi kebutuhan. Selain itu, menurut penelitian yang dilakukan juga diketahui sekitar 56 % dari kesalahan yang terjadi pada suatu pengembangan perangkat lunak disebabkan oleh kesalahan pada penentuan kebutuhan. Penyebab kesalahan pada suatu pengembangan perangkat lunak dapat dilihat dalam Gambar 2.2



Gambar 2.2 Grafik Penyebab Kesalahan pada Pengembangan Perangkat Lunak

Sumber : (Anas et al., 2016)

Berdasarkan hasil yang ditunjukan pada Gambar 2.1, penting dilakukannya validasi terhadap spesifikasi kebutuhan mengingat sekitar 56% kesalahan terjadi pada tahap spesifikasi kebutuhan. Untuk menghindari kesalahan pada tahap spesifikasi kebutuhan, perlu dilakukan *requirement validation*. Pada penelitian yang dilakukan tersebut dijelaskan empat jenis *requirement validation* yang dapat digunakan. Keempat jenis itu antara lain *requirements inspection*, *requirements validation*, *requirement testing* dan *viewpoint-oriented requirements*. Perbandingan dari keempat teknik *requirement validation* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Perbandingan Teknik *Requirement Validation*

	<i>Requirements Inspection</i>	<i>Requirements Prototyping</i>	<i>Requirements Testing</i>	<i>Viewpoint-oriented Requirements</i>
Besar tim	Besar	Kecil	Besar	Kecil
Biaya	Lebih mahal	Lebih murah	Lebih mahal	Lebih murah
Besar	Besar	Besar dan	Besar	Besar dan



	Requirements Inspection	Requirements Prototyping	Requirements Testing	Viewpoint- oriented Requirements
organisasi		kecil		kecil
Penggunaan ulang	Tidak	Iya	Iya	Tidak
Keterlibatan pengguna	Tidak	Iya	Tidak	Iya

Berdasarkan perbandingan yang dijelaskan pada Tabel 2.2, dapat disimpulkan bahwa teknik *requirements validation* yang paling tepat digunakan pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas ini adalah teknik *requirements prototyping*. *Requirements prototyping* digunakan didasarkan pada karakteristik pengembangan perangkat lunak pendampingan mahasiswa disabilitas yang memerlukan biaya pengembangan yang kecil, jumlah tim yang sedikit serta memerlukan keterlibatan pengguna dalam proses pengembangannya. Selain itu, perlu dilakukannya validasi dalam bentuk *prototype* ditujukan agar calon pengguna dari perangkat lunak pendampingan mahasiswa disabilitas dapat memahami spesifikasi kebutuhan yang akan dikembangkan.

2.7 Pendekatan Berbasis Objek

Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas, dilakukan dengan pendekatan berbasis objek. Pendekatan berbasis objek dilakukan dengan membuat objek yang dapat saling berinteraksi dan mengatur operasinya sendiri. Pendekatan berbasis objek mempermudah dalam proses perubahan pada pengembangan sistem karena perubahan pada suatu objek tidak akan berpengaruh pada objek lainnya (Sommerville, 2011). Pendekatan berbasis objek terbagi menjadi perancangan berbasis objek (*object oriented design*) dan pengembangan berbasis objek (*object oriented programming*).

Perancangan berbasis objek adalah proses kegiatan perancangan yang terdiri dari perancangan arsitektur, penggambaran desain menggunakan objek yang berbeda, perancangan dokumen antarmuka serta mengidentifikasi objek dalam sistem. Perancangan berbasis objek digunakan untuk melakukan desain dari arsitektur sistem, membangun model desain dan mendefinisikan antarmuka (Sommerville, 2011). Pengembangan berbasis objek adalah teknik pengembangan yang dilakukan dengan mengelompokkan kode – kode ke dalam suatu class dimana setiap class menyediakan fungsi – fungsi yang dijalankan oleh objek dari class tersebut (Azis, 2005). Pengembangan berbasis objek digunakan dengan harapan mampu mempercepat proses pengembangan dikarenakan setiap fungsi yang dapat digunakan berkali – kali. Selain itu, pengembangan berbasis objek memudahkan dalam proses perubahan dikarenakan struktur dari perangkat lunak yang kokoh.



2.8 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah suatu standar dalam melakukan pemodelan bahasa dengan konsep pemodelan berbasis objek. UML memiliki berbagai jenis diagram yang membantu sebagai pemodelan dari berbagai model sistem. Pada penerapannya, ada lima jenis diagram UML utama yang dapat digunakan untuk melakukan pemodelan pada suatu sistem diantaranya adalah *Activity Diagram*, *Use Case Diagram*, *Sequence Diagram*, *Class Diagram* dan *State Diagram* (Sommerville, 2011).

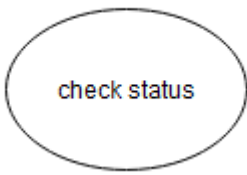
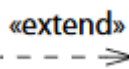
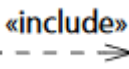
2.8.1 Use Case

Use case merupakan fitur fundamental dari sebuah UML. *Use case* digunakan untuk menggambarkan interaksi dari aktor serta tipe interaksi yang dilakukan oleh aktor. Sekumpulan dari *use case* akan menggambarkan bagaimana kemungkinan interaksi yang mungkin terjadi yang akan dijelaskan pada kebutuhan sistem. Pada *use case*, aktor menggambarkan orang atau sistem lain yang melakukan interaksi kepada sistem. Kemudian *use case* menjadi perwakilan dari kelas yang berinteraksi dengan aktor yang kemudian dihubungkan dengan garis untuk menggambarkan bahwa aktor dan *use case* melakukan interaksi. *Use case* digunakan secara luas sebagai teknik dalam melakukan elisitasi kebutuhan (Sommerville, 2011). Notasi *use case* dapat dilihat pada Tabel 2.3 berikut.

Tabel 2.3 Notasi Use Case

Gambar	Nama	Penjelasan
--------	------	------------



Gambar	Nama	Penjelasan
	Actor	Actor merupakan representasi dari pengguna atau sistem lain di luar sistem yang melakukan interaksi dengan sistem (Song, 2013).
	Use Case	Use case adalah representasi fungsionalitas yang disediakan oleh sistem yang diekspresikan sebagai urutan pesan antara use case maupun aktor (Rumbaugh, Jacobson, & Booch, 2013).
	Extend	Extend merepresentasikan penyisipan suatu use case ke dalam use case lainnya. (Rumbaugh et al., 2013).
	Association	Association merepresentasikan hubungan antara use case dan aktor (Rumbaugh et al., 2013).
	Include	Include merepresentasikan penyisipan suatu perilaku ke dalam use case utama dimana use case utama memerlukan sisipan perilaku tersebut (Rumbaugh et al., 2013).

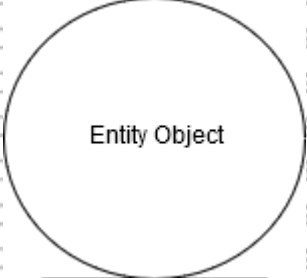
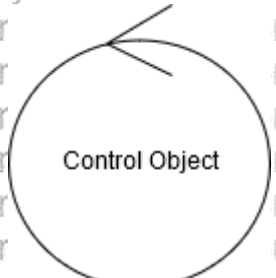
2.8.2 Sequence Diagram

Sequence diagram adalah sebuah diagram yang mampu menggambarkan interaksi antar objek pada suatu sistem. Sequence diagram dapat menjelaskan urutan dari interaksi yang terjadi pada suatu use case (Sommerville, 2011). Pada sequence diagram akan terdapat beberapa notasi pendukung seperti *activation*, *lifeline*, *object*, *actor*, *boundary*, dan *control*. Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas, sequence diagram diperlukan pada tahapan perancangan untuk mengetahui secara detail bagaimana interaksi dan alur kerja dari perangkat lunak pendampingan mahasiswa disabilitas. Untuk dapat menjelaskan bagaimana alur sistem bekerja, sequence diagram menggunakan beberapa notasi yang dijelaskan pada Tabel 2.4 berikut.

Tabel 2.4 Notasi *Sequence Diagram*

Gambar	Nama	Penjelasan
	<i>Activation</i>	<i>Activation</i> merepresentasikan periode waktu dari operasi yang dilakukan suatu elemen (Rumbaugh et al., 2013).
	<i>Lifeline</i>	<i>Lifeline</i> merupakan garis putus-putus yang merepresentasikan partisipasi setiap objek pada suatu waktu (Rumbaugh et al., 2013).
	<i>Object</i>	<i>Object</i> merepresentasikan kelas atau objek yang berinteraksi pada suatu sequence diagram yang belum dispesifikasikan menjadi aktor, <i>boundary</i> , maupun <i>entity</i> .
	<i>Boundary</i>	<i>Boundary</i> merepresentasikan suatu antarmuka yang menghubungkan antara sistem dan aktor dari luar sistem. <i>Boundary</i> harus selalu ada pada sistem untuk mendukung alur kerja dari sistem (Rumbaugh et al., 2013).
	<i>Actor</i>	<i>Actor</i> merupakan representasi dari pengguna atau sistem lain di luar sistem yang melakukan interaksi dengan sistem (Song, 2013).



	<i>Entity</i>	<i>Entity</i> merepresentasikan penyimpanan data atau informasi pada suatu sistem. <i>Entity</i> bersifat pasif dikarenakan tidak melakukan inisiasi interaksi pada suatu sistem (Rumbaugh et al., 2013).
	<i>Control</i>	<i>Control</i> merepresentasikan elemen yang menghubungkan dan melakukan kontrol antara <i>entity</i> dan <i>boundary</i> (Rumbaugh et al., 2013).

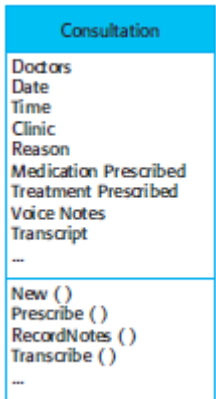
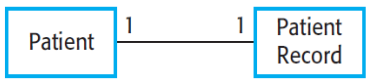
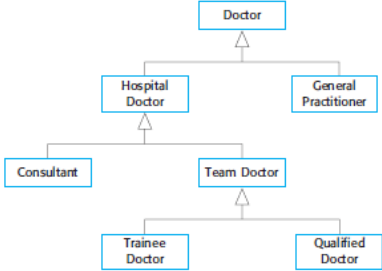
2.8.3 Class Diagram

Class diagram merupakan diagram yang digunakan untuk merepresentasikan hubungan antar kelas pada suatu pengembangan perangkat lunak. *Class diagram* digunakan ketika dilakukannya pengembangan perangkat lunak berbasis objek. Selain digunakan untuk merepresentasikan hubungan antar kelas, *class diagram* juga dapat digunakan untuk menjelaskan informasi yang terkandung dari suatu kelas secara detail seperti atribut dan operasi yang terdapat dari suatu kelas. Pada *class diagram* terdapat beberapa notasi seperti yang dijelaskan pada Tabel 2.5.

Tabel 2.5 Notasi *Class Diagram*

Gambar	Nama	Penjelasan
--------	------	------------

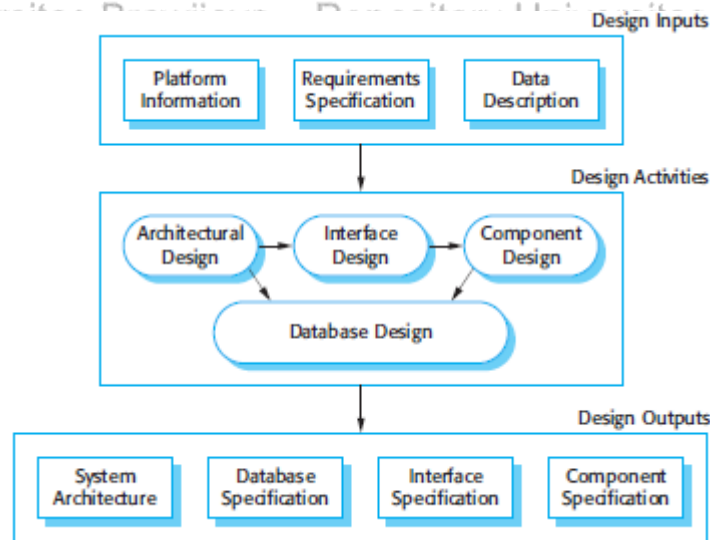


Gambar	Nama	Penjelasan
	Class	Class merepresentasikan kelas pada suatu perangkat lunak. Pada class terdapat atribut dan operasi. Atribut merepresentasikan atribut yang terdapat pada suatu kelas sedangkan operasi merepresentasikan fungsi yang dapat dijalankan pada kelas tersebut.
	Asosiasi	Asosiasi merepresentasikan hubungan yang dimiliki antar kelas.
	Generalisasi	Generalisasi merepresentasikan hubungan yang dimiliki antar kelas dimana terdapat <i>super class</i> yang akan mewarisi atribut maupun operasi ke <i>sub class</i> yang terhubung dengannya.

Sumber : (Sommerville, 2011)

2.9 Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan proses yang berpengaruh pada kualitas dari perangkat lunak. Perancangan perangkat lunak dilakukan dengan melakukan identifikasi dan menjabarkan fundamental perangkat lunak secara umum serta hubungan – hubungannya. Perancangan perangkat lunak dilakukan secara berurutan dimana masing – masing tahapan perancangan akan memberikan dampak pada perancangan lainnya. Perancangan perangkat lunak bergantung kepada kebutuhan dari pengembangannya (Sommerville, 2011). Tahapan perancangan perangkat lunak dapat dilihat dalam Gambar 2.3.



Gambar 2.3 Model Umum dari Proses Perancangan Perangkat Lunak

Sumber : (Sommerville, 2011)

2.9.1 Perancangan Arsitektur

Perancangan arsitektur merupakan proses untuk identifikasi struktur perangkat lunak secara keseluruhan. Perancangan arsitektur juga dilakukan untuk menggambarkan modul dari perangkat lunak yang akan dikembangkan, hubungan antar modul serta cara pendistribusian dari modul yang akan dikembangkan (Sommerville, 2011). Pada perangkat lunak pendampingan mahasiswa disabilitas, akan digunakan perancangan arsitektur antara lain perancangan *sequence diagram*, *class diagram* dan algoritme.

2.9.1.1 Perancangan Sequence Diagram

Perancangan *sequence diagram* diperlukan untuk memodelkan interaksi antara aktor dan objek pada sistem. *Sequence diagram* dapat menggambarkan secara lengkap alur interaksi yang dilakukan untuk menyelesaikan suatu operasi. *Sequence diagram* juga dapat membantu dalam menentukan kasus uji karena *sequence diagram* mampu mendeskripsikan masukan yang diperlukan serta keluaran yang dihasilkan (Sommerville, 2011).

2.9.1.2 Perancangan Class Diagram

Perancangan *class diagram* dilakukan agar dapat diketahui atribut serta operasi yang dapat dilakukan oleh objek dari suatu class. Perancangan *class diagram* dilakukan pada pengembangan pada bahasa pemrograman berbasis objek (Sommerville, 2011).

2.9.1.3 Perancangan Algoritme

Perancangan algoritme diperlukan untuk mengetahui bagaimana langkah – langkah penyelesaian dari suatu kebutuhan yang dimiliki pengguna. Perancangan algoritme membantu untuk mengimajinasikan bagaimana perangkat lunak akan



menyelesaikan suatu permasalahan secara terstruktur dan berurutan dan membantu dalam proses implementasi dari perangkat lunak.

2.9.2 Perancangan Antarmuka

Perancangan antarmuka merupakan proses untuk identifikasi antarmuka dari masing – masing komponen / modul dari perangkat lunak. Perancangan antarmuka membantu untuk mengetahui bagaimana suatu komponen akan dikembangkan. Perancangan antarmuka yang dilakukan akan membantu menghindari kondisi ketidakpastian sehingga komponen dapat dikembangkan secara berkelanjutan (Sommerville, 2011).

2.9.3 Perancangan Basis Data

Perancangan basis data merupakan proses untuk identifikasi dari struktur data dari perangkat lunak yang akan dikembangkan. Perancangan basis data akan menggambarkan bagaimana data yang diperlukan oleh perangkat lunak disajikan dalam suatu basis data. Perancangan basis data akan mendeskripsikan entitas dari data, atribut yang terkait dengan suatu entitas serta hubungan antar masing – masing entitas (Sommerville, 2011).

2.10 Strategi Pengujian Perangkat Lunak

Strategi pengujian perangkat lunak adalah sekumpulan aktifitas spesifik yang dilakukan oleh penguji untuk mendapatkan hasil yang spesifik. Strategi pengujian dilakukan untuk meminimalisir terjadinya kesalahan – kesalahan yang terjadi selama proses pengujian berlangsung. Kesalahan pengujian yang dimaksud antara lain biaya yang melebihi batas, jadwal yang melebar, kesalahan penting pada perangkat lunak serta kesalahan lainnya (DESAI & SRIVASTAVA, 2016). Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas, strategi pengujian digunakan untuk menentukan pengujian apa saja yang akan dilakukan. Pengujian yang dilakukan yaitu pengujian unit untuk melakukan uji terhadap modul, pengujian validasi untuk melakukan uji terhadap fungsional yang telah di implementasikan dan pengujian kompatibilitas dilaksanakan untuk menguji kebutuhan non fungsional dari perangkat lunak. Pengujian ini dilakukan untuk memastikan jika perangkat lunak dapat dijalankan pada lingkungan yang berbeda.

2.11 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses verifikasi untuk penilaian serta peningkatan kualitas dari perangkat lunak. Pengujian perangkat lunak dibagi menjadi dua yaitu analisis secara statis dan analisis secara dinamis. Analisis statis dilakukan berdasarkan pada dokumen seperti *source code*, dokumen desain, dokumen persyaratan dan lainnya. Analisis dinamis dilakukan berdasarkan uji coba dengan menjalankan perangkat lunak untuk menemukan kemungkinan kegagalan dari perangkat lunak. Pada pengujian perangkat lunak dilakukan proses verifikasi dan validasi. Verifikasi dilakukan untuk memastikan proses



pengembangan telah sesuai dengan persyaratan. Validasi dilakukan untuk memastikan apakah perangkat lunak telah memenuhi harapan penggunaanya (Naik, 2008). Pengujian perangkat lunak diperlukan untuk memastikan semua kebutuhan telah diimplementasikan secara benar dan memastikan perangkat lunak pendampingan mahasiswa disabilitas telah sesuai dan siap untuk digunakan dalam proses pendampingan mahasiswa disabilitas. Pada proses pengujian akan dilakukan menggunakan tiga metode uji yaitu pengujian validasi, pengujian unit dan pengujian kompatibilitas.

2.11.1 Pengujian Validasi

Pengujian validasi merupakan proses uji yang dijalankan dengan menggunakan spesifikasi perangkat lunak sebagai landasannya. Pengujian validasi dilakukan dengan memperlakukan proses pengujian seolah – olah penguji tidak mengetahui isi dari materi yang diuji sehingga proses pengujian hanya bisa ditentukan dengan memperhatikan kondisi masukan yang diberikan pada perangkat lunak serta kondisi keluaran dihasilkan dari perangkat lunak tersebut. Pengujian validasi dapat pula dilaksanakan apabila penguji tidak memiliki hak khusus untuk mengakses sumber kode dari perangkat lunak (Sommerville, 2011).

2.11.2 Pengujian Unit

Pengujian unit merupakan proses uji yang dijalankan dengan menggunakan kode sumber perangkat lunak sebagai landasannya. Pengujian unit dapat dilakukan ketika penguji memiliki hak untuk akses sumber kode dari perangkat lunak yang diuji. Pengujian unit diawali dengan menentukan *flow-graph* dari sumber kode sehingga didapatkan jalur logis dari perangkat lunak. Proses pengujian dilanjutkan dengan memastikan setiap jalur yang didapatkan dijalankan dengan memastikan semua jalur logis dijalankan minimal sekali dengan memperhatikan apakah setiap jalur tersebut dapat berjalan sesuai kondisi yang ditetapkan (Sommerville, 2011).

2.11.3 Pengujian Kompatibilitas

Pengujian kompatibilitas merupakan proses uji yang dijalankan dengan menggunakan lingkungan uji sebagai landasannya. Pengujian kompatibilitas dilakukan untuk memastikan perangkat lunak dapat dijalankan pada setiap lingkungan kerja yang berbeda seperti *platform*, sistem operasi ataupun manajemen basis data (Naik, 2008). Pada pengujian kompatibilitas dari perangkat lunak pendampingan mahasiswa disabilitas, akan dilakukan untuk memastikan perangkat lunak dapat berjalan pada *browser* yang berbeda.

2.12 Teknologi Pengembangan

Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas, digunakan beberapa teknologi untuk menunjang pengembangan perangkat lunak ini. Teknologi pengembangan yang digunakan terdiri dari bahasa pemrograman dan *framework* untuk memastikan kebutuhan perangkat lunak



yang telah didefinisikan pada tahap analisis kebutuhan dapat dikembangkan. Teknologi pengembangan yang digunakan antara lain *Personal Home Page* (PHP), *HyperText Markup Language* (HTML), *Cascading Style Sheet* (CSS), *Laravel* dan *MySQL*.

2.12.1 Personal Home Page (PHP)

Personal Home Page (PHP) adalah sebuah bahasa pemrograman yang didesain khusus untuk Web. PHP memiliki kemampuan untuk berinteraksi dengan database. PHP bekerja bersamaan dengan web server dan didesain untuk diimplementasikan dengan mudah pada file berbasis *Hypertext Markup Language* (HTML) (Valade, 2004). Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas ini, bahasa pemrograman PHP akan digunakan sebagai bahasa pemrograman utama yang digunakan untuk melakukan pemrosesan data pada server. PHP akan menyediakan fungsi yang diperlukan agar perangkat lunak pendampingan mahasiswa disabilitas dapat berjalan sesuai dengan kebutuhan pengguna.

2.12.2 Hypertext Markup Language (HTML)

Hypertext Markup Language (HTML) adalah bahasa pemrograman website yang memiliki fungsi untuk memberikan struktur dari suatu konten dengan mendefinisikan elemen – elemen pada konten tersebut. Sebagai contoh HTML akan mendefinisikan paragraf, gambar, penomoran, dan lainnya pada suatu konten. HTML terdiri dari beberapa tag utama yang menjadi kerangka utamanya yaitu `<html>`, `<head>` dan `<body>` (Howe, 2014). Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas ini, HTML akan digunakan untuk memberikan kerangka dasar dari perangkat lunak. Dengan menggunakan HTML, informasi yang diperlukan pada perangkat lunak pendampingan mahasiswa disabilitas ini dapat ditampilkan dengan baik menggunakan browser dikarenakan browser akan secara otomatis menafsirkan kode – kode HTML secara benar.

2.12.3 Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) adalah bahasa pemrograman yang digunakan untuk memberikan desain dan mengatur tata letak dari suatu halaman website. Dengan menggunakan CSS, dapat dilakukan pengaturan desain mulai dari ukuran, warna, maupun letak dari elemen – elemen yang ada pada HTML (Howe, 2014). Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas ini, CSS akan digunakan untuk memberikan desain dan tata letak yang paling optimal untuk mempermudah pengguna khususnya mahasiswa disabilitas dalam menggunakan perangkat lunak ini. CSS akan dipadukan dengan HTML yang akan memberikan kerangka utama dari perangkat lunak yang akan dikembangkan.



2.12.4 Laravel

Laravel adalah framework PHP yang fleksibel. *Laravel* bersifat *Model – View – Controller (MVC)* dan memiliki kelebihan dapat dijalankan pada beberapa sistem operasi terutama pada sistem operasi berbasis Linux dan Mac OS. Hal ini dikarenakan beberapa perintah *bash* yang dapat digunakan dalam menunjang pengembangan *website* menggunakan *laravel* (Stauffer, 2017). Dengan menggunakan *laravel*, pengembangan perangkat lunak berbasis *website* akan lebih cepat dilakukan. Hal ini dikarenakan file PHP yang digunakan dalam pengembangan lebih terstruktur sehingga dapat menunjang pengembangan perangkat lunak pendampingan mahasiswa disabilitas.

2.12.5 MySQL

MySQL adalah sistem manajemen *database* untuk *web server* paling populer. *MySQL* menggunakan bahasan *Structured Query Language (SQL)* yang digunakan oleh *database* lainnya seperti Oracle dan Microsoft SQL Server. *Database MySQL* berisi satu atau lebih tabel dimana setiap tabel memiliki baris dimana setiap baris memiliki kolom yang mengandung data. *MySQL* dapat diakses melalui tiga cara menggunakan antarmuka web seperti phpMyAdmin, menggunakan bahasa pemrograman seperti PHP dan menggunakan baris perintah (Nixon, 2012). Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas ini, *MySQL* digunakan dikarenakan dapat diakses menggunakan bahasa pemrograman PHP yang merupakan bahasa pemrograman utama dalam pengembangan perangkat lunak pendampingan mahasiswa disabilitas.

2.12.6 SortSite

SortSite merupakan alat yang digunakan untuk melakukan pengujian pada perangkat lunak berbasis *website*. *SortSite* akan melakukan analisa terhadap *website* dengan memindai seluruh halaman, kemudian memberikan laporan pemindaian secara rinci terhadap setiap halaman berdasarkan beberapa poin pemeriksaan seperti error, aksesibilitas, kompatibilitas peramban, validasi standar web, usabilitas hingga optimasi pencarian (PowerMapper, 2019).

2.12.7 PHPUnit

PHPUnit merupakan *framework* pengujian untuk perangkat lunak berbasis PHP berorientasi *programmer*. *PHPUnit* akan melakukan pengujian secara otomatis pada kondisi benar dari suatu unit pada perangkat lunak. *PHPUnit* menyediakan infrastruktur pengujian sehingga pengujian dapat dilakukan secara cepat, terisolasi, mudah untuk dibaca dan mudah untuk dijalankan. *PHPUnit* akan memberikan laporan berdasarkan jumlah pengujian yang benar termasuk apabila terdapat hasil yang berbeda dengan kondisi benar yang telah ditentukan sebelumnya (Bergmann, 2005).

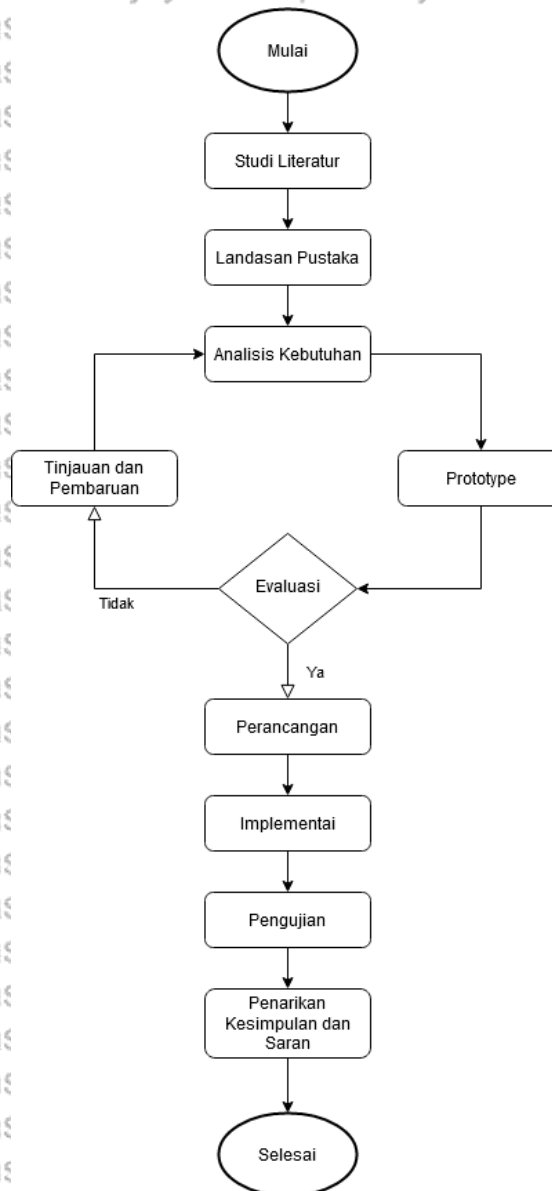


2.12.8 Bootstrap

Bootstrap merupakan *framework* pengembangan *website* berbasis *open source*. *Bootstrap* mampu memudahkan pengembang *website* untuk mengembangkan antarmuka secara cepat dikarenakan *bootstrap* adalah *framework* berbasis *cascading style sheet (CSS)* dan *JavaScript*. *Bootstrap* memiliki kemampuan untuk kompatibel di berbagai macam peramban berbeda, memiliki kapabilitas pengembangan *website* yang responsif, serta mendukung tampilan antarmuka pada ponsel (Jakobus & Marah, 2016). Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas, *bootstrap* digunakan untuk mengembangkan tampilan antarmuka dari sistem dikarenakan sifatnya yang responsif dan mendukung antarmuka pada perangkat ponsel. Selain itu, komponen interaktif pada *Bootstrap* telah menerapkan atribut yang direkomendasikan oleh *Web Accessibility Initiative – Accessible Rich Internet Applications* sebagai penyedia standar *website* untuk disabilitas sehingga komponen yang terdapat pada *Bootstrap* dapat dipahami oleh alat bantu disabilitas seperti *screen readers* sehingga *bootstrap* dapat digunakan untuk membantu mengembangkan perangkat lunak pendampingan mahasiswa disabilitas agar ramah untuk digunakan oleh mahasiswa disabilitas.

BAB 3 METODOLOGI

Pada bagian metodologi ini akan dijelaskan metode yang diimplementasikan untuk mengembangkan perangkat lunak pendampingan mahasiswa disabilitas. Dalam pengembangannya, aplikasi ini akan menggunakan *prototyping* dikarenakan perlunya umpan balik dari *stakeholder* yaitu koordinator pendampingan Pusat Studi dan Layanan Disabilitas Universitas Brawijaya untuk mendapatkan spesifikasi kebutuhan yang sesuai untuk mengembangkan perangkat lunak ini. Teknik *prototyping* dapat direpresentasikan seperti dalam Gambar 3.1.



Gambar 3.1 Diagram Metodologi Penelitian



3.1 Studi Literatur

Studi literatur merupakan kegiatan dalam mencari informasi yang dibutuhkan dalam melakukan penelitian. Dengan melakukan studi literatur diharapkan agar mendapat informasi yang bisa mendukung pengembangan perangkat lunak pendampingan mahasiswa disabilitas. Pada penelitian ini, studi literatur dilakukan kepada Lembaga Pusat Studi dan Layanan Disabilitas Universitas Brawijaya. Selain itu, studi literatur dilakukan pula secara langsung kepada para mahasiswa *volunteer* maupun disabilitas.

Dalam studi literatur ini, data yang didapatkan adalah data yang bersifat kualitatif. Data kualitatif adalah data yang didapatkan dari objek penelitian yang merupakan data terkait dengan penelitian dimana data yang disajikan berupa kalimat atau perkataan dari objek penelitian tersebut (Anggito & Johan, 2018). Hasil dari data kualitatif ini akan dilanjutkan oada tahap analisis kebutuhan untuk mengetahui kebutuhan apa saja yang perlu diimplementasikan untuk menyelesaikan permasalahan yang dihadapi oleh pihak terkait.

3.2 Landasan Pustaka

Landasan pustaka merupakan kegiatan yang dilakukan untuk mendapatkan dan mengumpulkan data terkait yang akan digunakan untuk menunjang proses penelitian dan pengembangan dari perangkat lunak pendampingan mahasiswa disabilitas. Kegiatan ini dilakukan dengan mencari data yang terkait dengan pengembangan perangkat lunak pendampingan mahasiswa disabilitas. Landasan pustaka dapat bersumber dari jurnal, penelitian sebelumnya, serta pustaka lainnya yang berkaitan dengan penelitian pengembangan perangkat lunak pendampingan mahasiswa disabilitas.

3.3 Analisis Kebutuhan

Dalam pengembangan aplikasi, Analisis kebutuhan diperlukan untuk mengembangkan aplikasi yang sesuai dengan kebutuhan yang dimiliki pengguna. Dimulai dari penentuan deskripsi umum dari aplikasi yang akan dibuat, dilanjutkan dengan penentuan kebutuhan pengguna yang terbagi menjadi kebutuhan fungsional dan non fungsional. Kemudian kebutuhan – kebutuhan yang telah ditentukan sebelumnya diimplementasikan ke dalam *use case diagram* dan *use case scenario*. Pada penentuan kebutuhan dari pengguna, akan difokuskan kepada proses wawancara kepada masing – masing *stakeholder*. Proses penggalian kebutuhan ini terfokus untuk mempelajari bagaimana proses pendampingan kepada mahasiswa disabilitas dilakukan. Dengan melihat pengalaman yang dimiliki masing – masing *stakeholder*, maka didapatkan daftar kebutuhan pengguna yang akan diimplementasikan pada perangkat lunak. Hasil dari penggalian kebutuhan ini akan diproses sehingga menghasilkan *use case diagram* dan *use case scenario* yang akan digunakan sebagai dasar dalam melakukan perancangan sistem.



3.3.1 Prototype

Fase *prototype* merupakan kegiatan untuk membuat purwarupa atau rancangan sementara yang merupakan gambaran untuk mewakili sistem yang akan dikembangkan. Kegiatan *prototype* bertujuan agar calon pengguna dapat memahami dengan jelas sistem yang akan dikembangkan. Pada fase ini akan dilakukan pembuatan *prototype* terhadap antarmuka dari sistem yang akan dikembangkan.

3.3.2 Evaluasi

Evaluasi adalah kegiatan yang dilakukan untuk melihat tanggapan dari calon pengguna. Evaluasi didasarkan kepada *prototype* yang telah dibuat sebelumnya. Dengan melakukan evaluasi, dapat diketahui tanggapan calon pengguna mengenai kelayakan dari sistem yang akan dibuat serta menjadi pertimbangan dari pengembangan sistem yang akan dibuat selama melakukan penelitian.

3.3.3 Tinjau dan Pembaruan

Tinjauan dan Pembaruan merupakan kegiatan yang dilakukan untuk mempertimbangkan hasil dari tahap evaluasi. Pada tahap ini akan dilakukan peninjauan dan mengambil keputusan apakah *prototype* yang telah dibuat dilakukan pembaruan atau tidak. Jika dirasa perlu dilakukannya pembaruan, akan dilakukan pembaruan dari *prototype* yang telah dibuat sebelumnya dengan pertimbangan hasil yang didapat dari tahap evaluasi.

3.4 Perancangan

Proses perancangan sistem untuk perangkat lunak pendampingan mahasiswa disabilitas ini terbagi menjadi perancangan arsitektur, perancangan basis data dan perancangan antarmuka.

3.4.1 Perancangan Arsitektur

Perancangan arsitektur dilakukan dengan membuat *sequence diagram*, *class diagram*, serta fungsi tambahan dari perangkat lunak pendampingan mahasiswa disabilitas ini. Pada perancangan arsitektur akan dilakukan identifikasi dari interaksi yang terjadi pada sistem yang akan dikembangkan mulai dari aktor yang berperan pada sistem, daftar kebutuhan yang direpresentasikan sebagai *use case*, relasi antar *use case* serta relasi aktor dengan *use case* dimana hasil identifikasi interaksi ini akan direpresentasikan pada *use case diagram*. Setelah *use case diagram* terbentuk, langkah selanjutnya adalah melakukan identifikasi terhadap objek yang terdapat pada sistem beserta alur kerjanya yang akan direpresentasikan ke dalam *sequence diagram*. Objek yang telah teridentifikasi akan dikelompokkan ke dalam kelas – kelas yang menjadi kerangka pembentuk dari sistem. Selain itu diidentifikasi pula hubungan antar kelas tersebut yang akan direpresentasikan ke dalam *class diagram*.



3.4.2 Perancangan Antarmuka

Pada tahap perancangan antarmuka, akan dilakukan perancangan terhadap antarmuka dari kebutuhan fungsional yang telah ditentukan. Perancangan antarmuka ini dilakukan untuk menggambarkan bagaimana hasil akhir dari perangkat lunak pendampingan mahasiswa disabilitas. Perancangan antarmuka akan menjadi dasar dalam menjalankan tahap implementasi antarmuka.

3.4.3 Perancangan Basis Data

Pada tahap perancangan basis data, akan dilakukan perancangan dari basis data perangkat lunak pendampingan mahasiswa disabilitas. Perancangan basis data didasarkan pada data yang dibutuhkan sesuai dengan spesifikasi kebutuhan yang telah ditentukan. Perancangan basis data akan menunjang proses implementasi basis data.

3.5 Implementasi

Pada fase ini, implementasi dari perangkat lunak pendampingan mahasiswa disabilitas akan dilakukan. Tahapan implementasi dari pengembangan perangkat lunak pendampingan mahasiswa disabilitas akan dilakukan berdasar pada fase perancangan. Implementasi dilakukan ketika *prototype* yang dihasilkan telah memenuhi ekspektasi dari calon pengguna dan layak untuk dikembangkan.

Proses implementasi diawali dengan melakukan implementasi pada basis data perangkat lunak sesuai dengan perancangan basis data yang telah dilakukan. Implementasi basis data dilakukan menggunakan *MySQL*. Setelah implementasi basis data selesai dilakukan, langkah selanjutnya melakukan implementasi terhadap arsitektur dan antarmuka yang didapatkan dari proses perancangan yang telah dilakukan. Implementasi arsitektur dilakukan dengan memberikan fungsi pada sistem menggunakan *PHP* sedangkan implementasi antarmuka dikembangkan menggunakan *HTML*, *CSS* dan *JavaScript*. Proses implementasi dilakukan menggunakan bantuan *framework laravel* untuk arsitektur dan *bootstrap* untuk antarmuka nya.

3.6 Pengujian

Pada proses pengujian, akan digunakan tiga metode uji yaitu pengujian unit, pengujian validasi dan pengujian kompatibilitas. Pada pengujian unit, dilakukan menggunakan alat bantu berupa *PHPUnit* sedangkan pada pengujian kompatibilitas akan dilakukan menggunakan alat bantu berupa aplikasi *SortSite*.

3.6.1 Pengujian Unit

Pada tahap pengujian unit, akan dilakukan pengujian terhadap *method* / operasi yang terdapat pada *class* dari perangkat lunak pendampingan mahasiswa disabilitas. Pengujian unit akan dilakukan berdasarkan kasus uji yang didapatkan dari kondisi pada *sequence diagram*.



3.6.2 Pengujian Validasi

Pada tahap pengujian validasi, akan dilakukan pengujian terhadap sistem melalui antarmuka dari perangkat lunak pendampingan mahasiswa disabilitas. Pengujian validasi dilakukan tanpa melihat algoritma serta kode sumber dari perangkat lunak pendampingan mahasiswa disabilitas.

3.6.3 Pengujian Kompatibilitas

Pada tahap pengujian kompatibilitas, akan dilakukan pengujian antara perangkat lunak pendampingan mahasiswa disabilitas dengan lingkungan uji. Pengujian kompatibilitas dilakukan untuk mengetahui tingkat kemampuan dari perangkat lunak untuk dijalankan pada lingkungan yang berbeda. Lingkungan uji pada pengujian kompatibilitas adalah beberapa jenis peramban sesuai dengan spesifikasi pada alat bantu *SortSite*.

3.7 Penarikan Kesimpulan dan Saran

Pada tahap penarikan kesimpulan dan saran, akan dilakukan penarikan kesimpulan dari proses pengembangan perangkat lunak pendampingan mahasiswa disabilitas mulai dari proses analisis kebutuhan, perancangan, implementasi dan pengujian yang telah dilakukan. Dari hasil penarikan kesimpulan ini juga dilakukan penarikan saran yang berguna untuk proses pengembangan perangkat lunak pendampingan mahasiswa disabilitas selanjutnya.



BAB 4 ANALISIS KEBUTUHAN

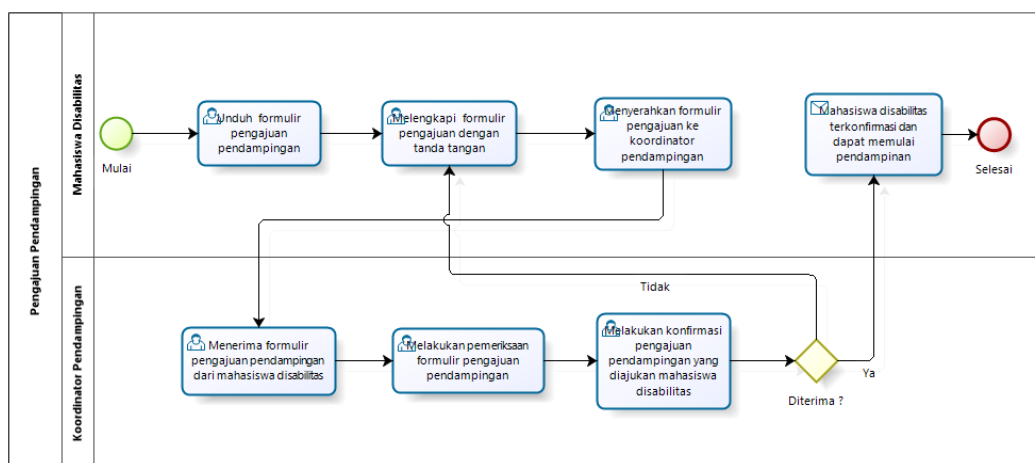
Pada bagian analisis kebutuhan, akan dibahas mengenai hasil analisis kebutuhan yang dilakukan untuk mengetahui daftar spesifikasi kebutuhan yang akan diimplementasikan pada perangkat lunak pendampingan mahasiswa disabilitas. Daftar spesifikasi kebutuhan yang telah didapatkan akan dikelompokkan menjadi dua jenis kebutuhan yaitu kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional merupakan kebutuhan yang harus diimplementasikan menjadi fitur dari perangkat lunak pendampingan mahasiswa disabilitas. Kebutuhan non fungsional merupakan pendukung dari kebutuhan fungsional.

4.1 Proses Bisnis Pendampingan Mahasiswa Disabilitas

Analisis kebutuhan akan dilakukan untuk mengetahui daftar kebutuhan yang dapat diimplementasikan pada perangkat lunak yang akan dikembangkan. Proses bisnis dari Pusat Studi dan Layanan Disabilitas (PSLD) diperlukan untuk mengetahui mekanisme yang dilakukan oleh PSLD dalam menjalankan pendampingan mahasiswa disabilitas. Proses bisnis pendampingan mahasiswa disabilitas ini didapatkan dari hasil wawancara yang dilakukan kepada koordinator pendampingan dari PSLD. Wawancara yang dilakukan memberikan hasil bahwa terdapat tiga proses bisnis utama dalam proses pendampingan antara lain pengajuan pendampingan, penjadwalan pendampingan dan pelaksanaan pendampingan. Proses bisnis pendampingan mahasiswa disabilitas akan direpresentasikan ke dalam *Business Process and Notation (BPMN)*.

4.1.1 Proses Bisnis Pengajuan Pendampingan

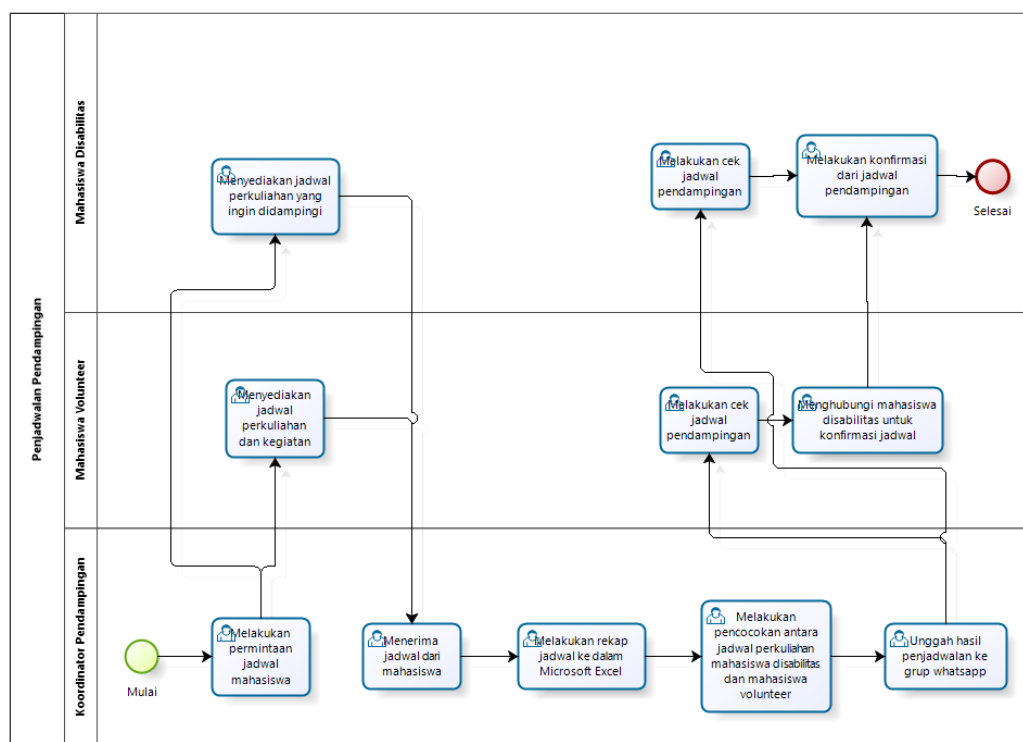
Proses pengajuan pendampingan diawali dari mahasiswa disabilitas melakukan pengajuan dengan mengunduh formulir pengajuan pendampingan. Mahasiswa disabilitas harus mengisi formulir sekaligus menambahkan tanda tangan pada formulir yang telah diisi. Formulir pengajuan kemudian diserahkan kepada koordinator pendampingan. Koordinator pendampingan akan memeriksa pengajuan pendampingan oleh mahasiswa disabilitas serta melakukan konfirmasi dari pengajuan pendampingan. Jika pengajuan pendampingan diterima, mahasiswa disabilitas akan dilanjutkan ke proses pendampingan. Jika pengajuan ditolak, koordinator pendampingan akan memberikan informasi penolakan dan mahasiswa disabilitas dapat mengulang proses pengajuan pendampingan kembali. Proses bisnis pengajuan pendampingan dapat dilihat dalam Gambar 4.1.



Gambar 4.1 Bisnis Proses Pengajuan Pendampingan

4.1.2 Proses Bisnis Penjadwalan Pendampingan

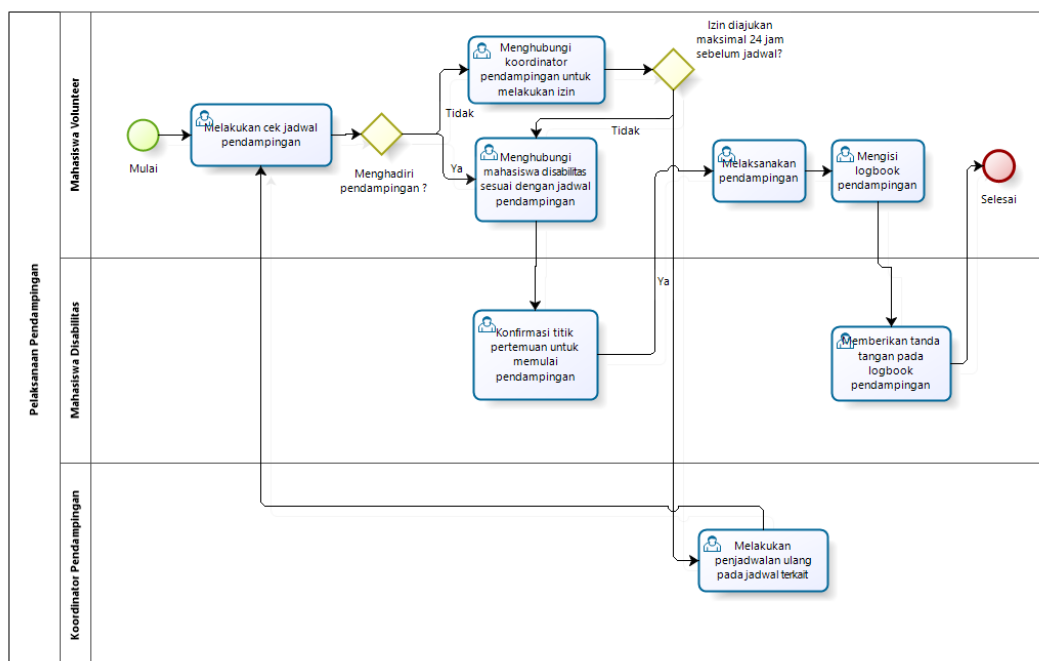
Proses penjadwalan pendampingan diawali dari koordinator pendampingan meminta jadwal dari mahasiswa disabilitas dan mahasiswa *volunteer*. Mahasiswa disabilitas dan mahasiswa *volunteer* akan menyerahkan jadwal perkuliahan serta jadwal kegiatannya kepada koordinator pendampingan. Koordinator pendampingan melakukan rekap sekaligus menjadwalkan proses pendampingan sesuai dengan jadwal yang dimiliki mahasiswa disabilitas dan mahasiswa *volunteer*. Hasil penjadwalan akan diunggah oleh koordinator pendampingan ke grup *whatsapp* sehingga mahasiswa disabilitas dan mahasiswa *volunteer* dapat melakukan cek jadwal pendampingannya. Mahasiswa *volunteer* akan menghubungi mahasiswa disabilitas untuk melakukan konfirmasi jadwal pendampingan sehingga proses pendampingan dapat dilaksanakan sesuai jadwal. Proses bisnis penjadwalan pendampingan dapat dilihat dalam Gambar 4.2.



Gambar 4.2 Bisnis Proses Penjadwalan Pendampingan

4.1.3 Proses Bisnis Pelaksanaan Pendampingan

Proses pelaksanaan pendampingan diawali dari mahasiswa *volunteer* melakukan cek jadwal pendampingan apakah mahasiswa pendampingan dapat menghadiri jadwal pendampingan. Jika mahasiswa *volunteer* tidak dapat hadir, mahasiswa harus menghubungi koordinator pendampingan untuk melakukan izin. Jika izin dilakukan 24 jam sebelum jadwal pendampingan maka koordinator pendampingan akan menjadwalkan ulang jadwal terkait, jika tidak maka mahasiswa *volunteer* tetap diwajibkan melakukan pendampingan. Jika mahasiswa *volunteer* dapat menghadiri pendampingan, mahasiswa *volunteer* akan menghubungi mahasiswa disabilitas sesuai jadwal. Mahasiswa disabilitas akan melakukan konfirmasi titik pertemuan untuk memulai pendampingan. Setelah bertemu, mahasiswa *volunteer* akan melaksanakan pendampingan terhadap mahasiswa disabilitas. Mahasiswa *volunteer* akan mengisi *logbook* pendampingan dan mahasiswa disabilitas memberikan tanda tangan pada *logbook* pendampingan sehingga proses pelaksanaan pendampingan dapat diakhiri. Proses bisnis pelaksanaan pendampingan dapat dilihat dalam Gambar 4.3.



Gambar 4.3 Bisnis Proses Pelaksanaan Pendampingan

4.2 Gambaran Umum Sistem

Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas ini, akan dikembangkan perangkat lunak berbasis web yang akan digunakan oleh tiga aktor. Aktor tersebut antara lain admin, mahasiswa *volunteer* dan mahasiswa disabilitas. Secara umum perangkat lunak pendampingan mahasiswa disabilitas ini akan memberikan fungsi untuk admin dalam mengelola pengajuan pendampingan yang dilakukan oleh mahasiswa disabilitas dan mengelola penjadwalan pendampingan yang akan dilakukan. Adapun dari sisi mahasiswa disabilitas dan mahasiswa *volunteer* akan disediakan fungsi yang mampu mengakomodir kegiatan pendampingan yang akan dijalankan berdasarkan penjadwalan yang telah dibuat sebelumnya oleh *admin*.

4.3 Identifikasi Aktor

Pada pengembangan perangkat lunak ini, akan terdapat tiga aktor yang dapat menggunakan fungsi – fungsi yang terdapat pada perangkat lunak pendampingan mahasiswa disabilitas. Aktor tersebut dibagi sesuai dengan peran yang dimilikinya pada sistem sebagaimana yang dijelaskan pada Tabel 4.1 berikut.

Tabel 4.1 Identifikasi Aktor

Aktor	Penjelasan
Guest	Guest merupakan aktor yang belum terautentikasi pada sistem. Pada perangkat



Aktor	Penjelasan
	lunak pendampingan mahasiswa disabilitas, <i>guest</i> mampu melakukan register dan login.
Admin	<i>Admin</i> merupakan aktor yang mampu melakukan login, manajemen pendampingan mahasiswa disabilitas. <i>Admin</i> mampu untuk mengelola penjadwalan pendampingan, melakukan verifikasi jadwal yang di masukkan oleh mahasiswa disabilitas dan mahasiswa <i>volunteer</i> , mengelola data diri mahasiswa disabilitas dan mahasiswa <i>volunteer</i> , melakukan penggantian pendamping, mengubah jadwal pendampingan.
Mahasiswa Disabilitas	Mahasiswa Disabilitas merupakan aktor yang mampu masukan jadwal perkuliahan, melakukan izin pendampingan, ubah profil, melihat jadwal pendampingan, melihat detail pendampingan, melakukan review hasil pendampingan.
Mahasiswa <i>Volunteer</i>	Mahasiswa <i>Volunteer</i> merupakan aktor yang mampu masukan jadwal kegiatan, melakukan izin pendampingan, ubah profil, melihat jadwal pendampingan, melihat detail pendampingan.

4.4 Iterasi Pertama

4.4.1 Daftar Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang harus diimplementasikan pada perangkat lunak pendampingan mahasiswa disabilitas. Kebutuhan fungsional akan menjadi dasar fitur yang akan dikembangkan untuk menunjang perangkat lunak dalam menyelesaikan permasalahan yang dialami oleh aktor

4.4.1.1 Kebutuhan Fungsional

Pada kebutuhan fungsional dari perangkat lunak pendampingan mahasiswa disabilitas, kebutuhan fungsionalitas yang telah dikumpulkan akan dikategorikan berdasarkan aktor – aktor yang telah didefinisikan sebelumnya. Oleh karena itu, kebutuhan fungsionalitas akan terbagi menjadi kebutuhan fungsional *guest*, kebutuhan fungsional *admin*, kebutuhan fungsional mahasiswa disabilitas dan kebutuhan fungsional mahasiswa *volunteer*.



A. Kebutuhan Fungsional Guest

Pada perangkat lunak pendampingan mahasiswa disabilitas ini, aktor *guest* memiliki kebutuhan fungsional berupa *login* dan *register*. *Login* dilakukan sebagai metode autentikasi untuk mengetahui jenis aktor yang terdaftar pada sistem, sedangkan *register* dilakukan untuk melakukan pendaftaran pada sistem. Kebutuhan fungsional *guest* dapat dijabarkan pada Tabel 4.2 berikut.

Tabel 4.2 Kebutuhan Fungsional Guest

No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
1	DIMENT_1_001	<i>Login</i>	Sistem mampu menyediakan fungsi <i>login</i> bagi <i>Guest</i> .
2	DIMENT_1_002	<i>Register</i>	Sistem mampu menyediakan fungsi <i>register</i> bagi <i>Guest</i> .

B. Kebutuhan Fungsional Admin

Pada perangkat lunak pendampingan mahasiswa disabilitas ini, aktor *admin* memiliki kebutuhan fungsional seperti tambah jadwal pendampingan, hapus jadwal pendampingan, ubah pendamping, lihat pengajuan izin, lihat pengajuan pendampingan, ubah profil mahasiswa. Kebutuhan fungsional *admin* dapat dijabarkan pada Tabel 4.3 berikut.

Tabel 4.3 Kebutuhan Fungsional Admin

No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
1	DIMENT_1_003	Lihat pendaftaran.	Sistem mampu menyediakan fungsi untuk melihat pendaftaran oleh mahasiswa.
2	DIMENT_1_004	Lihat detail pendaftaran.	Sistem mampu menyediakan fungsi untuk melihat pendaftaran oleh mahasiswa.
3	DIMENT_1_005	Terima pendaftaran.	Sistem mampu menyediakan fungsi menerima pendaftaran mahasiswa oleh admin.
4	DIMENT_1_006	Tambah jadwal pendampingan.	Sistem mampu menyediakan fungsi tambah jadwal pendampingan mahasiswa disabilitas.
5	DIMENT_1_007	Hapus jadwal pendampingan.	Sistem mampu menyediakan fungsi hapus jadwal pendampingan mahasiswa disabilitas.
6	DIMENT_1_008	Lihat pengajuan	Sistem mampu menyediakan



No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
		izin.	fungsi untuk melihat pengajuan izin pendampingan.
7	DIMENT_1_009	Lihat pengajuan pendampingan skripsi.	Sistem mampu menyediakan fungsi untuk melihat pengajuan pendampingan skripsi.
8	DIMENT_1_010	Lihat detail pengajuan pendampingan skripsi.	Sistem mampu menyediakan fungsi untuk melihat detail pengajuan pendampingan skripsi.
9	DIMENT_1_011	Pilih pendamping skripsi.	Sistem mampu menyediakan fungsi pilih pendamping skripsi untuk mahasiswa disabilitas.
10	DIMENT_1_012	Lihat jadwal mahasiswa.	Sistem mampu menyediakan fungsi untuk melihat jadwal mahasiswa.
12	DIMENT_1_013	Ubah jadwal mahasiswa.	Sistem mampu menyediakan fungsi untuk ubah jadwal mahasiswa.
12	DIMENT_1_036	Logout.	Sistem mampu menyediakan fungsi logout.

C. Kebutuhan Fungsional Mahasiswa Disabilitas

Pada perangkat lunak pendampingan mahasiswa disabilitas ini, aktor mahasiswa disabilitas memiliki kebutuhan fungsional seperti tambah jadwal, hapus jadwal, lihat jadwal pendampingan, lihat detail jadwal pendampingan, review pendampingan, ubah profil, izin pendampingan. Kebutuhan fungsional mahasiswa disabilitas dapat dijabarkan pada Tabel 4.4 berikut.

Tabel 4.4 Kebutuhan Fungsional Mahasiswa Disabilitas

No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
1	DIMENT_1_014	Tambah jadwal.	Sistem mampu menyediakan fungsi tambah jadwal perkuliahan mahasiswa disabilitas.
2	DIMENT_1_015	Hapus jadwal.	Sistem mampu menyediakan fungsi hapus jadwal perkuliahan mahasiswa disabilitas.
3	DIMENT_1_016	Lihat pendampingan perkuliahan.	Sistem mampu menyediakan fungsi melihat jadwal pendampingan mahasiswa



No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
			disabilitas.
4	DIMENT_1_017	Lihat detail jadwal pendampingan.	Sistem mampu menyediakan fungsi melihat detail jadwal pendampingan mahasiswa disabilitas.
5	DIMENT_1_018	Ubah profil.	Sistem mampu menyediakan fungsi untuk mengubah profil mahasiswa disabilitas.
6	DIMENT_1_019	Izin pendampingan.	Sistem mampu menyediakan fungsi untuk mengajukan izin pendampingan.
7	DIMENT_1_020	Ajukan pendampingan skripsi.	Sistem mampu menyediakan fungsi untuk mengajukan pendampingan skripsi.
8	DIMENT_1_021	Lihat detail pendampingan skripsi.	Sistem mampu menyediakan fungsi untuk melihat detail pendampingan skripsi.
9	DIMENT_1_022	Tampilkan <i>qr code</i> .	Sistem mampu menyediakan fungsi untuk menampilkan <i>qr code</i> pendampingan.
10	DIMENT_1_036	Logout.	Sistem mampu menyediakan fungsi logout.

D. Kebutuhan Fungsional Mahasiswa *Volunteer*

Pada perangkat lunak pendampingan mahasiswa disabilitas ini, aktor mahasiswa *volunteer* memiliki kebutuhan fungsional seperti tambah jadwal, hapus jadwal, lihat jadwal pendampingan, lihat detail jadwal pendampingan, ubah profil, ajukan izin pendampingan. Kebutuhan fungsional mahasiswa *volunteer* dapat dijabarkan pada Tabel 4.5 berikut

Tabel 4.5 Kebutuhan Fungsional Mahasiswa *Volunteer*

No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
1	DIMENT_1_014	Tambah jadwal.	Sistem mampu menyediakan fungsi tambah jadwal perkuliahan mahasiswa disabilitas.
2	DIMENT_1_015	Hapus jadwal.	Sistem mampu menyediakan fungsi hapus jadwal perkuliahan mahasiswa disabilitas.
3	DIMENT_1_016	Lihat	Sistem mampu menyediakan



No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
		pendampingan perkuliahan.	fungsi melihat jadwal pendampingan mahasiswa disabilitas.
4	DIMENT_1_017	Lihat detail jadwal pendampingan.	Sistem mampu menyediakan fungsi melihat detail jadwal pendampingan mahasiswa disabilitas.
5	DIMENT_1_018	Ubah profil.	Sistem mampu menyediakan fungsi untuk mengubah profil mahasiswa disabilitas.
6	DIMENT_1_019	Izin pendampingan.	Sistem mampu menyediakan fungsi untuk mengajukan izin pendampingan.
7	DIMENT_1_021	Lihat detail pendampingan skripsi.	Sistem mampu menyediakan fungsi untuk melihat detail pendampingan skripsi.
8	DIMENT_1_023	Mulai Pendampingan.	Sistem mampu menyediakan fungsi untuk memulai pendampingan.
9	DIMENT_1_036	Logout	Sistem mampu menyediakan fungsi logout.

4.4.1.2 Spesifikasi Kebutuhan

Pada spesifikasi kebutuhan, akan dijelaskan mengenai fungsionalitas turunan dari suatu kebutuhan fungsional. Spesifikasi kebutuhan menjadi fungsional pendukung serta pelengkap dari kebutuhan fungsional perangkat lunak pendampingan mahasiswa disabilitas. Penjelasan mengenai spesifikasi kebutuhan dari masing – masing kebutuhan fungsional dari perangkat lunak pendampingan mahasiswa dapat di lihat pada Tabel 4.6.

Tabel 4.6 Spesifikasi Kebutuhan

No	Kode Kebutuhan	Kode Spesifikasi	Deskripsi
1	DIMENT_1_004	DIMENT_1_004_01	Data pendaftaran yang ditampilkan meliputi foto profil, nama, fakultas, semester, tanggal lahir, angkatan, nomor <i>handphone</i> , email serta berkas pendaftaran.
2	DIMENT_1_006	DIMENT_1_006_01	Data masukan yang diperlukan meliputi jadwal pilihan, tanggal



No	Kode Kebutuhan	Kode Spesifikasi	Deskripsi
			pendampingan, jam mulai, jam selesai serta pilihan pendamping.
3	DIMENT_1_008	DIMENT_1_008_01	Data yang ditampilkan meliputi nama mahasiswa, hari, nama mata kuliah, ruangan, jam mulai, jam selesai, alasan serta pilihan pengganti.
4	DIMENT_1_008	DIMENT_1_008_02	Pilihan pendamping pengganti yang ditampilkan adalah mahasiswa yang tidak memiliki jadwal pada jam pendampingan.
5	DIMENT_1_010	DIMENT_1_010_01	Data yang ditampilkan meliputi nama mahasiswa, judul skripsi, nama pembimbing, waktu mulai, waktu selesai serta status pengajuan.
6	DIMENT_1_010	DIMENT_1_010_02	Pilihan pendamping yang ditampilkan adalah mahasiswa yang belum memiliki pendampingan skripsi.
7	DIMENT_1_013	DIMENT_1_013_01	Data yang bisa diubah meliputi keterangan jadwal, hari, ruangan, jam mulai dan jam selesai.
8	DIMENT_1_014	DIMENT_1_014_01	Data yang dimasukan meliputi keterangan jadwal, hari, ruangan, jam mulai dan jam selesai.
9	DIMENT_1_017	DIMENT_1_017_01	Data yang ditampilkan meliputi foto mahasiswa, nama mahasiswa, hari, nama mata kuliah, ruangan jam mulai dan jam selesai.
10	DIMENT_1_018	DIMENT_1_018_01	Data yang dapat diubah meliputi nama, tanggal lahir, angkatan, semester, nomor <i>handphone</i> serta alamat email.
11	DIMENT_1_020	DIMENT_1_020_01	Data yang dimasukan meliputi judul skripsi, nama pembimbing, waktu mulai, waktu selesai serta dokumen pendukung.



No	Kode Kebutuhan	Kode Spesifikasi	Deskripsi
12	DIMENT_1_021	DIMENT_1_021_01	Data yang ditampilkan meliputi waktu, judul skripsi, dosen pembimbing, fakultas serta nomor <i>handphone</i> .

4.4.2 Daftar Kebutuhan Non Fungsional

Setelah kebutuhan fungsional dari perangkat lunak pendampingan mahasiswa disabilitas didefinisikan, perlu didefinisikan pula kebutuhan non fungsional dari perangkat lunak pendampingan mahasiswa disabilitas. Kebutuhan non fungsional akan menjadi pendukung dari kebutuhan fungsional dan memberikan batasan serta standarisasi suatu kebutuhan fungsional. Kebutuhan non fungsional dari perangkat lunak pendampingan mahasiswa disabilitas dapat dilihat pada Tabel 4.7.

Tabel 4.7 Kebutuhan Non Fungsional

No	Kode Kebutuhan	Parameter	Deskripsi
1	DIMENT_2_001	<i>Security</i>	Pada pengisian <i>password</i> yang dimasukan pada <i>field password</i> , karakter akan berubah menjadi tanda '*'.
2	DIMENT_2_002	Kompatibilitas	Perangkat lunak dapat dijalankan secara <i>responsive</i> pada layer dari masing – masing <i>device</i> .

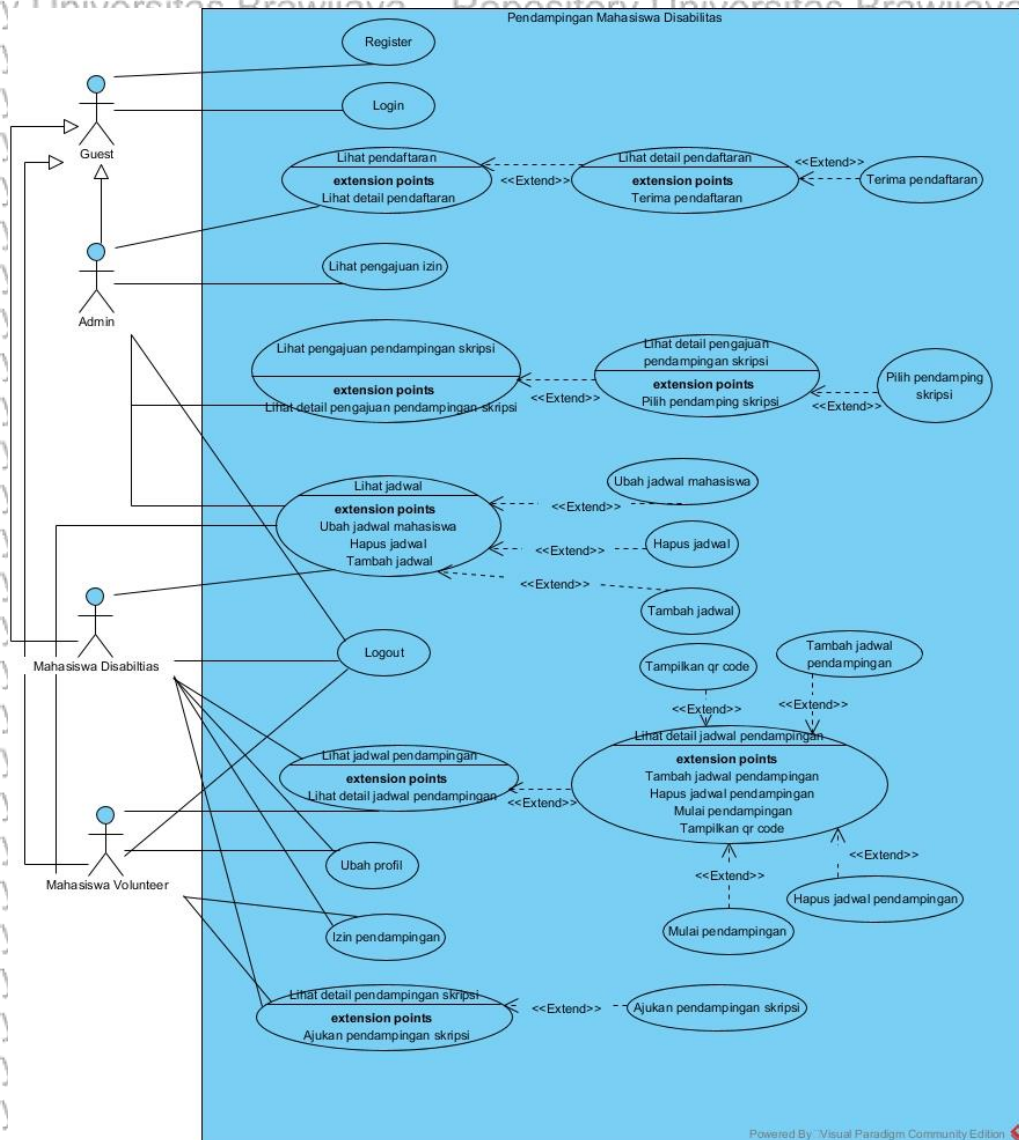
4.4.3 Pemodelan Kebutuhan

Pada bagian ini, akan dijelaskan mengenai pemodelan kebutuhan dari perangkat lunak pendampingan mahasiswa disabilitas. Dalam memodelkan kebutuhan yang telah didapatkan, akan dijabarkan dalam dua bagian yaitu *use case diagram* dan *use case scenario*. *Use case diagram* akan menjelaskan mengenai relasi antara kebutuhan – kebutuhan dengan aktor – aktor yang telah didefinisikan sebelumnya yang dijabarkan ke dalam bentuk diagram, kemudian *use case scenario* digunakan untuk menjabarkan bagaimana scenario berjalannya sistem dilihat dari sudut pandang sistem maupun aktor.

4.4.3.1 Use Case Diagram

Pada bagian ini, akan dijelaskan bagaimana relasi antara aktor dan kebutuhan dari perangkat lunak pendampingan mahasiswa disabilitas sehingga akan tergambar mengenai fungsi – fungsi yang akan tersedia pada perangkat lunak ini. Fungsi – fungsi pada perangkat lunak ini akan direpresentasikan ke dalam bentuk *use case* dimana setiap *use case* akan memiliki interaksi ke masing – masing aktor

yang telah didefinisikan sebelumnya. *Use case diagram* dari perangkat lunak pendampingan mahasiswa disabilitas dapat dilihat dalam Gambar 4.1 berikut.



Gambar 4.4 Use Case Diagram Perangkat Lunak Pendampingan Mahasiswa Disabilitas

4.4.3.2 Use Case Scenario

Pada bagian ini, akan dijelaskan bagaimana skenario yang dimiliki oleh sebuah *use case* dimana skenario ini akan menggambarkan bagaimana fungsi dari perangkat lunak pendampingan mahasiswa ini berjalan. Skenario yang dijabarkan dibagi menjadi *main flow* yang menjelaskan bagaimana fungsi berjalan secara normal, sedangkan *alternative flow* yang menjelaskan bagaimana fungsi berjalan diluar kondisi normal. *Use case scenario* akan dijelaskan pada Tabel 4.8 sampai Tabel 4.31.



A. Use Case Scenario Login

Pada fungsionalitas *login*, aktor yang dapat menjalankan fungsionalitas ini adalah *guest*. Fungsionalitas login memiliki fungsi untuk mengidentifikasi aktor dan melakukan autentikasi dari aktor yang sudah teridentifikasi. Skenario yang dijalankan adalah perangkat lunak melakukan pengecekan terhadap masukan yang diberikan antara lain *email* dan *password*. Kemudian jika data ditemukan maka akan melakukan autentikasi terhadap *guest* dan halaman *dashboard* terbuka. *Use case scenario login* dapat dilihat pada Tabel 4.8.

Tabel 4.8 Use Case Scenario Login

Nama Use Case	<i>Login</i> (DIMENT_1_001).
Objective	<i>Guest</i> dapat melakukan <i>login</i> pada perangkat lunak.
Aktor	<i>Guest</i> .
Pre-condition	1. <i>Guest</i> membuka halaman <i>login</i>. 2. Halaman <i>login</i> ditampilkan.
Main Flow	1. <i>Guest</i> mengisi <i>email</i> dan <i>password</i>. 2. <i>Guest</i> menekan tombol <i>login</i>. 3. Sistem melakukan cek data pengguna ke <i>database</i>. 4. Sistem menyediakan data sesuai <i>email</i> pengguna.
Alternative Flow	1. Perangkat lunak menampilkan informasi “<i>Please fill out this field</i>” jika terdapat <i>field</i> yang kosong. 2. Perangkat lunak menampilkan informasi “Kami tidak menemukan akun terkait.” jika data yang dimasukan salah.
Post-condition	<i>Guest</i> terautentikasi dan diarahkan ke halaman <i>dashboard</i> .

B. Use Case Scenario Register

Pada fungsionalitas *register*, aktor yang dapat menjalankan fungsionalitas ini adalah *guest*. Fungsionalitas login memiliki fungsi untuk mendaftarkan pengguna baru dari perangkat lunak pendampingan mahasiswa disabilitas. Skenario yang dijalankan adalah perangkat lunak melakukan pengecekan terhadap masukan yang diberikan antara lain nama lengkap, fakultas, jurusan, jenis kelamin, angkatan, tanggal lahir, jenis akun, jenis disabilitas (jika memilih sebagai mahasiswa disabilitas), *email* dan *password*. Kemudian data masukan akan di simpan ke *database* dan halaman *dashboard* terbuka. *Use case scenario register* dapat dilihat pada Tabel 4.9.

Tabel 4.9 Use Case Scenario Register

Nama Use Case	<i>Register</i> (DIMENT_1_002).
Objective	<i>Guest</i> dapat melakukan <i>register</i> pada perangkat lunak.



Aktor	Guest.
Pre-condition	1. Guest membuka halaman <i>register</i>. 2. Halaman <i>register</i> ditampilkan.
Main Flow	1. Guest mengisi nama lengkap, fakultas, jurusan, jenis kelamin, angkatan, tanggal lahir, jenis akun, jenis disabilitas (jika memilih sebagai mahasiswa disabilitas), <i>email</i> dan <i>password</i>. 2. Guest menekan tombol <i>register</i>. 3. Sistem melakukan cek data apabila <i>email</i> yang didaftarkan telah terdapat di <i>database</i>. 4. Sistem menyimpan data masukan pengguna ke <i>database</i>.
Alternative Flow	1. Perangkat lunak menampilkan informasi "Kami tidak menemukan akun terkait." jika terdapat <i>field</i> yang kosong. 2. Perangkat lunak menampilkan informasi "Akun telah terdaftar sebelumnya" jika <i>email</i> yang dimasukan telah terdaftar di <i>database</i>.
Post-condition	Guest terautentikasi dan diarahkan ke halaman <i>dashboard</i> .

C. Use Case Scenario Lihat Pendaftaran

Pada fungsionalitas lihat pendaftaran, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas lihat pendaftaran memiliki fungsi untuk melihat pendaftaran yang masuk oleh mahasiswa disabilitas maupun mahasiswa *volunteer*. Skenario yang dijalankan adalah *admin* membuka menu pendaftaran kemudian sistem akan menampilkan daftar pendaftaran yang masuk. *Use case scenario* lihat pendaftaran dapat dilihat pada Tabel 4.10.

Tabel 4.10 Use Case Scenario Lihat Pendaftaran

Nama Use Case	Lihat pendaftaran (DIMENT_1_003).
Objective	<i>Admin</i> dapat melihat pendaftaran yang masuk oleh mahasiswa disabilitas maupun mahasiswa <i>volunteer</i> .
Aktor	<i>Admin</i> .
Pre-condition	1. <i>Admin</i> telah ter autentikasi ke sistem.
Main Flow	1. <i>Admin</i> menekan menu pendaftaran. 2. Sistem mengambil data pendaftaran masuk pada <i>database</i>.
Alternative Flow	-
Post-condition	Sistem menampilkan daftar pendaftaran yang masuk.



D. Use Case Scenario Lihat Detail Pendaftaran

Pada fungsionalitas lihat detail pendaftaran, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas lihat detail pendaftaran memiliki fungsi untuk melihat informasi detail pendaftaran yang masuk oleh mahasiswa disabilitas maupun mahasiswa *volunteer*. Skenario yang dijalankan adalah halaman pengajuan pendaftaran telah ditampilkan, kemudian *admin* menekan tombol detail pada pendaftaran yang dipilih dan sistem menampilkan informasi detail pendaftaran. *Use case scenario* lihat detail pendaftaran dapat dilihat pada Tabel 4.11.

Tabel 4.11 Use Case Scenario Lihat Detail Pendaftaran

Nama Use Case	Lihat detail pendaftaran (DIMENT_1_004).
Objective	<i>Admin</i> dapat melihat detail informasi pendaftaran yang masuk oleh mahasiswa disabilitas maupun mahasiswa <i>volunteer</i> .
Aktor	<i>Admin</i> .
Pre-condition	1. Halaman daftar pengajuan pendaftaran telah ditampilkan.
Main Flow	1. <i>Admin</i> menekan tombol detail pada pendaftaran yang dipilih. 2. Sistem mengambil data detail pendaftaran untuk pengguna yang di pilih dari <i>database</i> .
Alternative Flow	-
Post-condition	Sistem menampilkan detail pendaftaran yang dipilih.

E. Use Case Scenario Terima Pendaftaran

Pada fungsionalitas terima pendaftaran, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsional terima pendaftaran memiliki fungsi untuk menerima pendaftaran yang masuk oleh mahasiswa disabilitas maupun mahasiswa *volunteer*. Skenario yang dijalankan adalah halaman detail pendaftaran telah ditampilkan, kemudian *admin* menekan tombol konfirmasi. *Use case scenario* terima pendaftaran dapat dilihat pada Tabel 4.12.

Tabel 4.12 Use Case Scenario Terima Pendaftaran

Nama Use Case	Terima pendaftaran (DIMENT_1_005).
Objective	<i>Admin</i> dapat menerima pengajuan pendaftaran yang masuk oleh mahasiswa disabilitas maupun mahasiswa <i>volunteer</i> .
Aktor	<i>Admin</i> .
Pre-condition	1. Halaman detail pendaftaran telah ditampilkan.
Main Flow	1. <i>Admin</i> menekan tombol konfirmasi pada halaman detail



	pendaftaran.
	2. Sistem melakukan perubahan status dari pengguna yang pendaftarannya di konfirmasi.
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Sistem menampilkan informasi jika pendaftaran berhasil di konfirmasi.

F. Use Case Scenario Tambah Jadwal Pendampingan

Pada fungsionalitas tambah jadwal pendampingan, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas tambah jadwal pendampingan memiliki fungsi untuk menambahkan jadwal pendampingan terhadap masing – masing mahasiswa. Skenario yang dijalankan adalah *admin* mengisi data jadwal pendampingan yang akan ditambahkan seperti memilih mahasiswa disabilitas yang akan ditambah jadwalnya, memilih hari, memilih jam pendampingan, memilih mahasiswa *volunteer*, memilih jenis pendampingan. Kemudian data masukan akan di simpan ke *database* dan halaman jadwal pendampingan terbuka. *Use case scenario* tambah jadwal pendampingan dapat dilihat pada Tabel 4.13.

Tabel 4.13 Use Case Scenario Tambah Jadwal Pendampingan

Nama <i>Use Case</i>	Tambah jadwal pendampingan (DIMENT_1_006).
<i>Objective</i>	<i>Admin</i> dapat menambah jadwal pendampingan.
Aktor	<i>Admin</i> .
<i>Pre-condition</i>	1. <i>Admin</i> membuka halaman jadwal pendampingan. 2. <i>Admin</i> memilih menu tambah jadwal. 3. Halaman tambah jadwal ditampilkan.
<i>Main Flow</i>	1. <i>Admin</i> memilih mahasiswa disabilitas yang akan ditambah jadwalnya, memilih hari, memilih jam pendampingan, memilih mahasiswa <i>volunteer</i>, memilih jenis pendampingan. 2. <i>Admin</i> menekan tombol tambah jadwal. 3. Sistem menyimpan data jadwal pendampingan ke <i>database</i>.
<i>Alternative Flow</i>	1. Perangkat lunak menampilkan informasi “<i>Please fill out this field</i>” jika terdapat <i>field</i> yang kosong.
<i>Post-condition</i>	Jadwal pendampingan berhasil ditambahkan dan halaman jadwal pendampingan ditampilkan.



G. Use Case Scenario Hapus Jadwal Pendampingan

Pada fungsionalitas hapus jadwal pendampingan, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas hapus jadwal pendampingan memiliki fungsi untuk menghapus jadwal pendampingan terhadap masing – masing mahasiswa. Skenario yang dijalankan adalah *admin* memilih jadwal yang akan dihapus dan menekan tombol hapus jadwal. Kemudian perubahan akan di simpan ke *database* dan halaman jadwal pendampingan terbuka. *Use case scenario* hapus jadwal pendampingan dapat dilihat pada Tabel 4.14.

Tabel 4.14 Use Case Scenario Hapus Jadwal Pendampingan

Nama Use Case	Hapus jadwal pendampingan (DIMENT_1_007).
Objective	<i>Admin</i> dapat menghapus jadwal pendampingan.
Aktor	<i>Admin</i> .
Pre-condition	1. <i>Admin</i> membuka halaman jadwal pendampingan. 2. <i>Admin</i> memilih menu hapus jadwal.
Main Flow	1. <i>Admin</i> memilih jadwal yang akan dihapus. 2. <i>Admin</i> menekan tombol hapus jadwal. 3. Sistem melakukan hapus data pada <i>database</i> sesuai dengan jadwal pendampingan yang dipilih <i>admin</i> .
Alternative Flow	1. Perangkat lunak menampilkan informasi “Pilih minimal 1 jadwal” jika belum ada jadwal yang dipilih.
Post-condition	Jadwal pendampingan berhasil dihapus dan halaman jadwal pendampingan ditampilkan.

H. Use Case Scenario Lihat Pengajuan Izin

Pada fungsionalitas lihat pengajuan izin, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas lihat pengajuan izin memiliki fungsi untuk melihat daftar pengajuan izin yang dilakukan masing – masing mahasiswa pada setiap jadwal pendampingan. Skenario yang dijalankan adalah *admin* membuka menu pengajuan izin dan daftar izin yang dilakukan oleh mahasiswa akan ditampilkan. *Use case scenario* lihat pengajuan izin dapat dilihat pada Tabel 4.15.

Tabel 4.15 Use Case Scenario Lihat Pengajuan Izin

Nama Use Case	Lihat pengajuan izin (DIMENT_1_008).
Objective	<i>Admin</i> dapat melihat pengajuan izin yang dilakukan oleh mahasiswa disabilitas dan mahasiswa <i>volunteer</i> .
Aktor	<i>Admin</i> .
Pre-condition	1. <i>Admin</i> telah terautentikasi.



<i>Main Flow</i>	1. <i>Admin</i> membuka menu pengajuan izin. 2. Sistem mengambil data pengajuan izin dari <i>database</i> .
<i>Alternative Flow</i>	
<i>Post-condition</i>	Halaman pengajuan izin ditampilkan.

I. Use Case Scenario Lihat Pengajuan Pendampingan Skripsi

Pada fungsionalitas lihat pengajuan pendampingan skripsi, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas lihat pengajuan pendampingan skripsi memiliki fungsi untuk melihat daftar pengajuan pendampingan skripsi yang dilakukan masing – masing mahasiswa disabilitas. Skenario yang dijalankan adalah *admin* membuka menu pengajuan pendampingan skripsi. Setelah itu daftar pengajuan pendampingan skripsi akan ditampilkan. *Use case scenario* lihat pengajuan pendampingan dapat dilihat pada Tabel 4.16.

Tabel 4.16 Use Case Scenario Lihat Pengajuan Pendampingan Skripsi

Nama <i>Use Case</i>	Lihat pengajuan pendampingan skripsi (DIMENT_1_009).
<i>Objective</i>	<i>Admin</i> dapat melihat pengajuan pendampingan skripsi yang dilakukan oleh mahasiswa disabilitas.
Aktor	<i>Admin</i> .
<i>Pre-condition</i>	1. <i>Admin</i> telah terautentikasi.
<i>Main Flow</i>	1. <i>Admin</i> membuka menu pengajuan pendampingan skripsi. 2. <i>Admin</i> menekan tombol lihat detail pengajuan pada pengajuan yang dipilih. 3. Sistem mengambil data pendampingan skripsi dari <i>database</i> dengan status diajukan.
<i>Alternative Flow</i>	
<i>Post-condition</i>	Halaman pengajuan pendampingan skripsi ditampilkan.

J. Use Case Scenario Lihat Detail Pengajuan Pendampingan Skripsi

Pada fungsionalitas lihat detail pengajuan pendampingan skripsi, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas lihat detail pengajuan pendampingan skripsi memiliki fungsi untuk melihat detail pengajuan pendampingan skripsi yang dilakukan masing – masing mahasiswa disabilitas. Skenario yang dijalankan adalah *admin* membuka menu pengajuan pendampingan skripsi, kemudian memilih pengajuan yang ada dan menekan tombol detail. Setelah itu pengajuan pendampingan skripsi akan ditampilkan. *Use case scenario* lihat pengajuan pendampingan dapat dilihat pada Tabel 4.17.

Tabel 4.17 Use Case Scenario Lihat Pengajuan Pendampingan

Nama <i>Use Case</i>	Lihat detail pengajuan pendampingan skripsi (DIMENT_1_010).
----------------------	---



<i>Objective</i>	<i>Admin</i> dapat melihat detail pengajuan pendampingan skripsi yang dilakukan oleh mahasiswa disabilitas.
Aktor	<i>Admin.</i>
<i>Pre-condition</i>	1. <i>Admin</i> telah terautentikasi. 2. Halaman daftar pengajuan pendampingan telah ditampilkan
<i>Main Flow</i>	1. <i>Admin</i> menekan tombol lihat detail pengajuan pada pengajuan yang dipilih. 2. Sistem mengambil data detail pengajuan yang dipilih dari <i>database</i> .
<i>Alternative Flow</i>	
<i>Post-condition</i>	Halaman detail pengajuan pendampingan skripsi ditampilkan.

K. Use Case Scenario Pilih Pendamping Skripsi

Pada fungsionalitas pilih pendamping skripsi, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas pilih pendamping skripsi memiliki fungsi untuk memilih pendamping skripsi dari pengajuan pendampingan skripsi yang dilakukan masing – masing mahasiswa disabilitas. Skenario yang dijalankan adalah *admin* membuka menu pengajuan pendampingan skripsi, kemudian memilih pengajuan yang ada dan menekan tombol selengkapnya. Setelah itu halaman detail pengajuan pendampingan skripsi ditampilkan dan *admin* dapat memilih pendamping skripsi untuk mahasiswa disabilitas yang mengajukan pendampingan skripsi. *Use case scenario* pilih pendamping skripsi dapat dilihat pada Tabel 4.18.

Tabel 4.18 *Use Case Scenario* Pilih Pendamping Skripsi

Nama <i>Use Case</i>	Pilih pendamping skripsi (DIMENT_1_011).
<i>Objective</i>	<i>Admin</i> dapat memilih pendamping untuk pengajuan pendampingan skripsi yang dilakukan oleh mahasiswa disabilitas.
Aktor	<i>Admin.</i>
<i>Pre-condition</i>	1. <i>Admin</i> telah terautentikasi.
<i>Main Flow</i>	1. <i>Admin</i> membuka menu pengajuan pendampingan skripsi. 2. <i>Admin</i> menekan tombol selengkapnya pada pengajuan yang dipilih. 3. Sistem menampilkan detail pengajuan skripsi dari pengajuan yang dipilih. 4. Sistem menampilkan daftar mahasiswa dengan status <i>volunteer</i> sebagai calon pendamping. 5. <i>Admin</i> memilih pendamping untuk pengajuan pendampingan skripsi.



	6. <i>Admin</i> menekan tombol konfirmasi.
	7. Sistem melakukan simpan data mahasiswa pendamping pada <i>database</i> .
<i>Alternative Flow</i>	
<i>Post-condition</i>	Pendamping skripsi dipilih dan disimpan ke <i>database</i> .

L. Use Case Scenario Lihat Jadwal Mahasiswa

Pada fungsionalitas lihat jadwal mahasiswa, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas lihat jadwal mahasiswa memiliki fungsi untuk melihat jadwal yang dimiliki masing – masing mahasiswa. Skenario yang dijalankan adalah *admin* membuka menu jadwal mahasiswa dan halaman jadwal mahasiswa ditampilkan. *Use case scenario* lihat jadwal mahasiswa dapat dilihat pada Tabel 4.19.

Tabel 4.19 Use Case Scenario Lihat Jadwal Mahasiswa

Nama <i>Use Case</i>	Lihat jadwal mahasiswa (DIMENT_1_012).
<i>Objective</i>	<i>Admin</i> dapat melihat jadwal dari masing – masing mahasiswa.
Aktor	<i>Admin</i> .
<i>Pre-condition</i>	1. <i>Admin</i> telah terautentikasi.
<i>Main Flow</i>	1. <i>Admin</i> membuka menu jadwal mahasiswa. 2. Sistem mengambil data jadwal mahasiswa yang dipilih.
<i>Alternative Flow</i>	
<i>Post-condition</i>	Halaman jadwal mahasiswa ditampilkan.

M. Use Case Scenario Ubah Jadwal Mahasiswa

Pada fungsionalitas ubah jadwal mahasiswa, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas ubah jadwal mahasiswa memiliki fungsi untuk melakukan perubahan jadwal yang dimiliki masing – masing mahasiswa. Skenario yang dijalankan adalah *admin* membuka menu jadwal mahasiswa, menekan tombol ubah pada jadwal yang dipilih dan melakukan perubahan pada jadwal tersebut. *Use case scenario* ubah jadwal mahasiswa dapat dilihat pada Tabel 4.20.

Tabel 4.20 Use Case Scenario Ubah Jadwal Mahasiswa

Nama <i>Use Case</i>	Ubah jadwal mahasiswa (DIMENT_1_013).
<i>Objective</i>	<i>Admin</i> dapat melakukan perubahan jadwal dari masing – masing mahasiswa.
Aktor	<i>Admin</i> .
<i>Pre-condition</i>	1. Halaman jadwal mahasiswa telah ditampilkan.
<i>Main Flow</i>	1. <i>Admin</i> menekan tombol ubah pada jadwal yang dipilih.



	2. Sistem mengambil data dari jadwal mahasiswa yang akan diubah dari <i>database</i> .
	3. <i>Admin</i> melakukan perubahan pada data jadwal mahasiswa.
	4. <i>Admin</i> menekan tombol simpan.
	5. Sistem menyimpan perubahan pada <i>database</i> .
<i>Alternative Flow</i>	1. Sistem menampilkan informasi " <i>Please fill out this field</i> " apabila terdapat data kosong.
<i>Post-condition</i>	Perubahan jadwal mahasiswa disimpan ke <i>database</i> .

N. Use Case Scenario Tambah Jadwal

Pada fungsionalitas tambah jadwal perkuliahan, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas tambah jadwal perkuliahan memiliki fungsi agar mahasiswa disabilitas dan mahasiswa *volunteer* dapat menambahkan jadwal perkuliahannya pada perangkat lunak pendampingan mahasiswa disabilitas. Skenario yang dijalankan adalah mahasiswa disabilitas atau mahasiswa *volunteer* membuka menu jadwal perkuliahan, kemudian menekan tombol tambah jadwal dan mengisi nama perkuliahan, jam perkuliahan, ruang kuliah, hari. Setelah semua terisi kemudian menekan tombol tambahkan jadwal. *Use case scenario* tambah jadwal perkuliahan dapat dilihat pada Tabel 4.21.

Tabel 4.21 Use Case Scenario Tambah Jadwal

Nama <i>Use Case</i>	Tambah jadwal (DIMENT_1_014).
<i>Objective</i>	Mahasiswa disabilitas dan <i>volunteer</i> dapat menambahkan jadwal perkuliahan.
Aktor	1. Mahasiswa disabilitas. 2. Mahasiswa <i>volunteer</i> .
<i>Pre-condition</i>	1. Mahasiswa disabilitas dan <i>volunteer</i> telah terautentikasi.
<i>Main Flow</i>	1. Aktor membuka menu jadwal perkuliahan. 2. Sistem mengambil data seluruh jadwal dari mahasiswa dari <i>database</i> . 3. Aktor menekan tombol tambah jadwal. 4. Aktor mengisi nama perkuliahan, jam perkuliahan, ruang kuliah, hari. 5. Aktor menekan tombol tambahkan jadwal. 6. Sistem melakukan simpan data dari jadwal mahasiswa yang telah di masukan ke <i>database</i> .
<i>Alternative Flow</i>	1. Perangkat lunak menampilkan informasi " <i>Please fill out this field</i> " jika terdapat <i>field</i> yang kosong.
<i>Post-condition</i>	Perangkat lunak menampilkan pesan "Penambahan jadwal perkuliahan berhasil" dan halaman jadwal perkuliahan



ditampilkan.

O. Use Case Scenario Hapus Jadwal

Pada fungsionalitas hapus jadwal perkuliahan, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas hapus jadwal perkuliahan memiliki fungsi agar mahasiswa disabilitas dan mahasiswa *volunteer* dapat menghapus jadwal perkuliahannya pada perangkat lunak pendampingan mahasiswa disabilitas. Skenario yang dijalankan adalah mahasiswa disabilitas atau mahasiswa *volunteer* membuka menu jadwal perkuliahan, memilih jadwal perkuliahan yang akan dihapus, kemudian menekan tombol hapus.. *Use case scenario* hapus jadwal perkuliahan dapat dilihat pada Tabel 4.22.

Tabel 4.22 Use Case Scenario Hapus Jadwal

Nama <i>Use Case</i>	Hapus jadwal (DIMENT_1_015).
<i>Objective</i>	Aktor dapat menghapus jadwal perkuliahan.
Aktor	1. Mahasiswa disabilitas. 2. Mahasiswa <i>volunteer</i> .
<i>Pre-condition</i>	1. Aktor telah terautentikasi.
<i>Main Flow</i>	1. Aktor membuka menu jadwal perkuliahan. 2. Sistem mengambil data seluruh jadwal dari mahasiswa dari database. 3. Aktor memilih jadwal yang akan dihapus. 4. Aktor menekan tombol hapus. 5. Sistem melakukan penghapusan data jadwal dari database pada jadwal yang dipilih.
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Perangkat lunak menampilkan pesan “Penghapusan jadwal perkuliahan berhasil” dan halaman jadwal perkuliahan ditampilkan.

P. Use Case Scenario Lihat Jadwal Pendampingan

Pada fungsionalitas lihat jadwal pendampingan, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas lihat jadwal pendampingan memiliki fungsi agar mahasiswa disabilitas dan mahasiswa *volunteer* dapat melihat jadwal pendampingan yang akan dilakukannya. Skenario yang dijalankan adalah mahasiswa disabilitas atau mahasiswa *volunteer* sudah terautentikasi, kemudian membuka menu jadwal pendampingan. *Use case scenario* lihat jadwal pendampingan dapat dilihat pada Tabel 4.23.

**Tabel 4.23 Use Case Scenario Lihat Pendampingan Perkuliahan**

Nama Use Case	Lihat pendampingan perkuliahan (DIMENT_1_016).
Objective	Aktor dapat melihat jadwal pendampingan.
Aktor	1. Mahasiswa disabilitas. 2. Mahasiswa <i>volunteer</i> .
Pre-condition	1. Aktor telah terautentikasi.
Main Flow	1. Aktor membuka menu jadwal pendampingan. 2. Sistem mengambil data jadwal pendampingan dari <i>database</i> sesuai dengan aktor yang terautentikasi.
Alternative Flow	-
Post-condition	Halaman jadwal pendampingan ditampilkan.

Q. Use Case Scenario Lihat Detail Jadwal Pendampingan

Pada fungsionalitas lihat detail jadwal pendampingan, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas lihat jadwal pendampingan memiliki fungsi agar mahasiswa disabilitas dan mahasiswa *volunteer* dapat melihat detail jadwal pendampingan yang akan dilakukannya. Skenario yang dijalankan adalah mahasiswa disabilitas atau mahasiswa *volunteer* sudah terautentikasi, kemudian membuka menu jadwal pendampingan dan menekan tombol detail pada jadwal yang dipilih. *Use case scenario* lihat detail jadwal pendampingan dapat dilihat pada Tabel 4.24.

Tabel 4.24 Use Case Scenario Lihat Detail Jadwal Pendampingan

Nama Use Case	Lihat detail jadwal pendampingan (DIMENT_1_017).
Objective	Aktor dapat melihat detail jadwal pendampingan.
Aktor	1. Mahasiswa disabilitas. 2. Mahasiswa <i>volunteer</i> .
Pre-condition	1. Aktor telah terautentikasi.
Main Flow	1. Aktor membuka menu jadwal pendampingan. 2. Sistem mengambil data jadwal pendampingan dari <i>database</i> sesuai dengan aktor yang terautentikasi. 3. Aktor menekan tombol detail pada jadwal pendampingan yang dipilih. 4. Sistem mengambil data jadwal pendampingan sesuai dengan jadwal pendampingan yang dipilih dari <i>database</i> .
Alternative Flow	-
Post-condition	Halaman detail jadwal pendampingan ditampilkan.



R. Use Case Scenario Ubah Profil

Pada fungsionalitas ubah profil, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas ubah profil memiliki fungsi agar mahasiswa disabilitas dan mahasiswa *volunteer* dapat melakukan perubahan pada profilnya. Skenario yang dijalankan adalah mahasiswa disabilitas atau mahasiswa *volunteer* membuka menu profil kemudian menekan tombol ubah profil, kemudian dilakukan perubahan pada profilnya. Setelah dilakukan perubahan kemudian menekan tombol simpan. *Use case scenario* ubah profil dapat dilihat pada Tabel 4.25.

Tabel 4.25 Use Case Ubah Profil

Nama Use Case	Ubah profil (DIMENT_1_018).
Objective	Aktor dapat mengubah profil.
Aktor	1. Mahasiswa disabilitas 2. Mahasiswa <i>volunteer</i> .
Pre-condition	1. Aktor telah terautentikasi.
Main Flow	1. Aktor membuka menu profil. 2. Sistem mengambil data profil aktor yang terautentikasi dari <i>database</i> . 3. Aktor menekan tombol ubah profil. 4. Aktor melakukan perubahan pada profilnya. 5. Aktor menekan tombol simpan. 6. Sistem menyimpan data profil aktor pada <i>database</i> sesuai dengan perubahan yang dilakukan.
Alternative Flow	1. Sistem menampilkan informasi jika terdapat data yang kosong.
Post-condition	Perangkat lunak menampilkan pesan "Profil berhasil dirubah" dan halaman profil ditampilkan.

S. Use Case Scenario Izin Pendampingan

Pada fungsionalitas izin pendampingan, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas izin pendampingan memiliki fungsi agar mahasiswa disabilitas dan mahasiswa *volunteer* dapat mengajukan izin pendampingan pada masing – masing jadwal pendampingan. Skenario yang dijalankan adalah mahasiswa disabilitas atau mahasiswa *volunteer* menekan tombol detail pada jadwal yang dipilih, menekan tombol ajukan izin, kemudian mengisi alasan mengajukan izin dan menekan tombol ajukan. *Use case scenario* izin pendampingan dapat dilihat pada Tabel 4.26.

Tabel 4.26 Use Case Izin Pendampingan

Nama Use Case	Izin pendampingan (DIMENT_1_019).
---------------	-----------------------------------



<i>Objective</i>	Aktor dapat melakukan pengajuan izin pendampingan.
Aktor	1. Mahasiswa disabilitas. 2. Mahasiswa <i>volunteer</i> .
<i>Pre-condition</i>	1. Halaman jadwal pendampingan ditampilkan.
<i>Main Flow</i>	1. Aktor menekan tombol detail pada jadwal yang dipilih. 2. Sistem mengambil data jadwal pendampingan yang dipilih dari <i>database</i> . 3. Aktor menekan tombol ajukan izin. 4. Aktor mengisi alasan mengajukan izin. 5. Aktor menekan tombol ajukan. 6. Sistem melakukan perubahan status pendampingan dan menyimpan alasan izin ke <i>database</i> .
<i>Alternative Flow</i>	1. Perangkat lunak menampilkan pesan "Izin tidak dapat dilakukan karena telah kurang dari 1x24 jam"
<i>Post-condition</i>	Perangkat lunak menampilkan pesan "Izin berhasil diajukan" dan halaman detail jadwal pendampingan ditampilkan.

T. Use Case Scenario Ajukan Pendampingan Skripsi

Pada fungsionalitas ajukan pendampingan skripsi, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas. Fungsionalitas ajukan pendampingan skripsi memiliki fungsi agar mahasiswa disabilitas dapat melakukan pengajuan pendampingan untuk proses skripsi yang dilaksanakannya. Skenario yang dijalankan adalah mahasiswa disabilitas mengisi data judul penelitian, waktu penelitian, dosen pembimbing, mengunggah dokumen pengajuan pendampingan skripsi dan menekan tombol ajukan. *Use case scenario* ajukan pendampingan skripsi dapat dilihat pada Tabel 4.27.

Tabel 4.27 Use Case Scenario Ajukan Pendampingan Skripsi

Nama Use Case	Ajukan pendampingan skripsi (DIMENT_1_020).
<i>Objective</i>	Mahasiswa disabilitas dapat mengajukan pendampingan skripsi.
Aktor	Mahasiswa disabilitas.
<i>Pre-condition</i>	1. Mahasiswa disabilitas membuka menu ajukan pendampingan skripsi. 2. Halaman ajukan pendampingan skripsi ditampilkan.
<i>Main Flow</i>	1. Mahasiswa disabilitas membuka menu pendampingan skripsi. 2. Sistem melakukan validasi jika mahasiswa telah melakukan pengajuan pendampingan skripsi sebelumnya. 3. Mahasiswa disabilitas mengisi data judul penelitian, waktu penelitian, dosen pembimbing.



	4. Mahasiswa disabilitas mengunggah dokumen permohonan pengajuan pendampingan skripsi. 5. Mahasiswa disabilitas menekan tombol ajukan. 6. Sistem melakukan validasi dari masukan yang dilakukan oleh mahasiswa disabilitas. 7. Sistem mengunggah dokumen pengajuan pendampingan skripsi dan menyimpan data pengajuan ke <i>database</i> .
<i>Alternative Flow</i>	1. Perangkat lunak menampilkan informasi “ <i>Please fill out this field</i> ” jika terdapat <i>field</i> yang kosong. 2. Sistem menampilkan informasi “Maaf anda telah melakukan pengajuan sebelumnya. Silahkan tunggu konfirmasi” jika mahasiswa disabilitas sudah pernah melakukan pengajuan pendampingan skripsi.
<i>Post-condition</i>	Perangkat lunak menampilkan pesan “Pengajuan pendampingan skripsi berhasil” dan halaman utama ditampilkan.

U. Use Case Scenario Lihat Detail Pendampingan Skripsi

Pada fungsionalitas lihat detail pendampingan skripsi, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas ajukan pendampingan skripsi memiliki fungsi agar mahasiswa disabilitas dan *volunteer* dapat melihat pendampingan skripsi yang akan dilaksanakan. Skenario yang dijalankan adalah mahasiswa disabilitas maupun *volunteer* membuka menu pendampingan skripsi dan sistem akan menampilkan detail pendampingan skripsi. *Use case scenario* detail pendampingan skripsi dapat dilihat pada Tabel 4.28.

Tabel 4.28 Use Case Scenario Lihat Detail Pendampingan Skripsi

Nama <i>Use Case</i>	Lihat detail pendampingan skripsi (DIMENT_1_021).
<i>Objective</i>	Mahasiswa disabilitas maupun <i>volunteer</i> dapat melihat detail pendampingan skripsi yang akan dilaksanakan.
Aktor	1. Mahasiswa disabilitas. 2. Mahasiswa <i>volunteer</i> .
<i>Pre-condition</i>	1. Aktor telah terautentikasi.
<i>Main Flow</i>	1. Aktor menekan tombol menu pendampingan skripsi. 2. Sistem mengambil data pendampingan skripsi dari <i>database</i> sesuai dengan aktor yang terautentikasi.
<i>Alternative Flow</i>	1. Sistem menampilkan pesan “Saat ini kamu belum memiliki pendampingan skripsi.” apabila belum ada pendampingan skripsi.
<i>Post-condition</i>	Halaman detail pendampingan skripsi ditampilkan.



V. Use Case Scenario Tampilkan Qr Code

Pada fungsionalitas tampilkan *qr code*, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas. Fungsionalitas tampilkan *qr code* memiliki fungsi agar mahasiswa disabilitas dapat menampilkan *qr code* dari jadwal pendampingan yang akan dilaksanakan. Skenario yang dijalankan adalah mahasiswa disabilitas membuka menu jadwal pendampingan, kemudian menekan tombol selengkapnya pada jadwal yang dipilih, kemudian menekan tombol lihat *qr code*. *Use case scenario* tampilkan *qr code* dapat dilihat pada Tabel 4.29.

Tabel 4.29 Use Case Scenario Tampilkan Qr Code

Nama Use Case	Tampilkan <i>qr code</i> (DIMENT_1_022).
Objective	Mahasiswa disabilitas dapat menampilkan <i>qr code</i> dari jadwal pendampingan.
Aktor	Mahasiswa disabilitas.
Pre-condition	1. Mahasiswa disabilitas telah terautentikasi.
Main Flow	1. Mahasiswa disabilitas membuka menu jadwal pendampingan. 2. Sistem mengambil data jadwal pendampingan dari <i>database</i> sesuai dengan aktor yang terautentikasi. 3. Mahasiswa disabilitas menekan tombol selengkapnya pada jadwal pendampingan yang dipilih. 4. Sistem mengambil data jadwal pendampingan dari <i>database</i> sesuai dengan jadwal yang dipilih. 5. Mahasiswa disabilitas menekan tombol lihat <i>qr code</i>. 6. Sistem melakukan <i>generate qr code</i> dari jadwal pendampingan.
Alternative Flow	-
Post-condition	<i>Qr code</i> jadwal pendampingan ditampilkan.

W. Use Case Scenario Mulai Pendampingan

Pada fungsionalitas mulai pendampingan, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa *volunteer*. Fungsionalitas mulai pendampingan memiliki fungsi agar mahasiswa *volunteer* dapat memulai pendampingan dari jadwal pendampingan yang akan dilaksanakan. Skenario yang dijalankan adalah mahasiswa *volunteer* membuka menu jadwal pendampingan, kemudian menekan tombol selengkapnya pada jadwal yang dipilih, kemudian menekan tombol *scan qr code*. *Use case scenario* mulai pendampingan dapat dilihat pada Tabel 4.30.

Tabel 4.30 Use Case Scenario Mulai Pendampingan

Nama Use Case	Mulai pendampingan (DIMENT_1_023).
---------------	------------------------------------

Objective	Mahasiswa <i>volunteer</i> dapat memulai pendampingan dari jadwal pendampingan.
Aktor	Mahasiswa <i>volunteer</i> .
Pre-condition	1. Mahasiswa <i>volunteer</i> telah terautentikasi.
Main Flow	1. Mahasiswa <i>volunteer</i> membuka menu jadwal pendampingan. 2. Sistem mengambil data jadwal pendampingan dari database sesuai dengan aktor yang terautentikasi. 3. Mahasiswa <i>volunteer</i> menekan tombol selengkapnya pada jadwal pendampingan yang dipilih. 4. Sistem mengambil data jadwal pendampingan dari database sesuai dengan jadwal yang dipilih. 5. Mahasiswa <i>volunteer</i> menekan tombol <i>scan qr code</i>. 6. Sistem menjalankan fungsi kamera dan menampilkan <i>preview</i> dari <i>scan qr code</i>.
Alternative Flow	1. Sistem menampilkan informasi "Anda tidak berhak memulai pendampingan ini" apabila mahasiswa <i>volunteer</i> melakukan <i>scan</i> pada <i>qr code</i> yang berbeda dari jadwal pendampingannya.
Post-condition	Pendampingan dikonfirmasi dan perubahan status pendampingan disimpan ke <i>database</i> .

X. Use Case Scenario Logout

Pada fungsionalitas *logout*, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*, mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas *logout* memiliki fungsi untuk menghapus sesi dari aktor. Skenario yang dijalankan adalah aktor menekan tombol *logout* dan halaman *login* ditampilkan. *Use case scenario logout* dapat dilihat pada Tabel 4.31.

Tabel 4.31 Use Case Scenario Logout

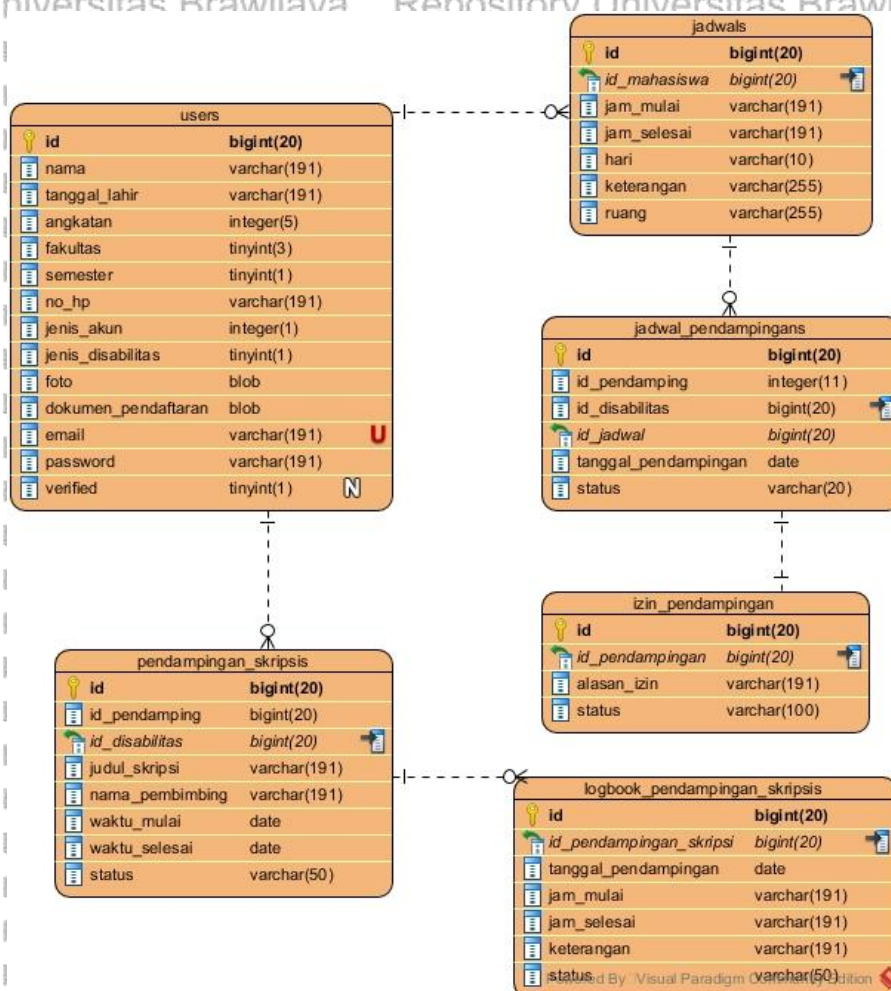
Nama Use Case	<i>Logout</i> (DIMENT_1_036).
Objective	Aktor dapat melakukan <i>login</i> pada perangkat lunak.
Aktor	1. <i>Admin</i>. 2. Mahasiswa disabilitas. 3. Mahasiswa <i>volunteer</i>.
Pre-condition	1. Aktor telah terautentikasi.
Main Flow	1. Aktor menekan tombol <i>logout</i>. 2. Sistem melakukan penghapusan sesi <i>login</i> dari aktor yang terautentikasi.
Alternative Flow	-



Post-condition Sesi dihapus dan halaman *login* ditampilkan.

4.4.4 Analisis Data

Setelah kebutuhan perangkat lunak pendampingan mahasiswa disabilitas didapatkan, langkah selanjutnya yang harus dilakukan adalah melakukan analisis data. Analisis data perlu dilakukan untuk mengetahui data apa saja yang akan digunakan dan diolah agar perangkat lunak pendampingan mahasiswa disabilitas dapat berjalan dengan semestinya. Selain itu analisis data juga diperlukan untuk mengetahui data apa saja yang diperlukan untuk mengakomodir kebutuhan perangkat lunak agar mampu menyelesaikan permasalahan pengguna. Untuk mengetahui bagaimana data dari perangkat lunak pendampingan mahasiswa disabilitas beserta hubungan antara satu data dengan data lainnya maka digunakan *entity relationship diagram* (ERD). Analisis data dari perangkat lunak pendampingan mahasiswa disabilitas dapat dilihat dalam Gambar 4.2.



Gambar 4.5 Entity Relationship Diagram Iterasi Pertama Perangkat Lunak Pendampingan Mahasiswa Disabilitas



Entity relationship diagram yang terdapat dalam Gambar 4.2 memiliki enam entitas. Pada entitas *users* terdapat 14 atribut dan memiliki hubungan *one to many* dengan entitas *jadwals* dan pendampingan_skripsis. Pada entitas *jadwals* terdapat tujuh atribut dan memiliki hubungan *many to one* dengan *users* dan *one to many* dengan *jadwal_pendampingans*. Pada entitas *jadwal_pendampingans* terdapat enam atribut dan memiliki hubungan *many to one* dengan *jadwals* dan *one to one* dengan *izin_pendampingan*. Pada entitas *izin_pendampingan* terdapat empat atribut dan memiliki hubungan *one to one* dengan *jadwal_pendampingans*. Pada entitas *pendampingan_skripsis* terdapat delapan atribut dan memiliki hubungan *many to one* dengan *users* dan *one to many* dengan *logbook_pendampingan_skripsis*. Pada entitas *logbook_pendampingan_skripsis* terdapat tujuh atribut dan memiliki hubungan *many to one* dengan entitas *pendampingan_skripsis*.

4.4.5 Evaluasi

Pada iterasi pertama dari analisis kebutuhan perangkat lunak pendampingan mahasiswa disabilitas, didapatkan hasil berupa terdapat empat aktor yang berperan pada sistem yaitu *guest*, *admin*, mahasiswa disabilitas dan mahasiswa *volunteer*. Pada iterasi pertama ini juga terdapat total 24 kebutuhan fungsional serta terdapat enam entitas pada basis data dari perangkat lunak pendampingan mahasiswa disabilitas. Selain itu, pada iterasi pertama juga terdapat *use case diagram* yang terdiri dari 24 *use case* dimana masing – masing *use case* tersebut memiliki *use case scenario*.

Evaluasi iterasi pertama dilakukan dengan melakukan demonstrasi purwarupa yang telah dibuat sesuai dengan spesifikasi kebutuhan. Evaluasi dari iterasi pertama perangkat lunak pendampingan mahasiswa disabilitas memberikan hasil sebagai berikut :

1. Pendambahan fungsional pendampingan tugas akhir pada perangkat lunak yang akan dikembangkan. Fungsional pendampingan tugas akhir diperlukan untuk akomodasi pendampingan tugas akhir mahasiswa disabilitas dan mahasiswa *volunteer*.
2. Penambahan fungsional pengajuan skripsi untuk aktor mahasiswa disabilitas. Fungsional pengajuan pendampingan skripsi diperlukan karena terdapat syarat pengajuan tambahan yang diperlukan sebelum melaksanakan pendampingan skripsi. Persyaratan yang diperlukan yaitu mahasiswa berada pada semester yang ditentukan untuk melaksanakan skripsi dan mendapat surat pengantar dari fakultas.
3. Penambahan fungsional pengajuan tugas akhir untuk aktor mahasiswa disabilitas.
4. Penambahan fungsional melihat pengajuan tugas akhir, terima pengajuan dan pilih pendamping untuk pendampingan tugas akhir untuk aktor *admin*.



5. Penambahan fungsional pendampingan tugas akhir untuk aktor mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsional pendampingan tugas akhir terdiri dari lihat detail pendampingan tugas akhir, ajukan *logbook* pendampingan untuk mahasiswa *volunteer* dan konfirmasi *logbook* pendampingan untuk mahasiswa disabilitas.
6. Penambahan fitur ambil lokasi untuk pendampingan skripsi dan tugas akhir. Fitur ambil lokasi diperlukan untuk memastikan proses pendampingan skripsi dan tugas akhir dilaksanakan di lingkungan PSLD sesuai syarat yang telah ditetapkan oleh PSLD.
7. Penambahan fitur cetak *logbook* dari pendampingan tugas akhir dan skripsi untuk aktor admin.
8. Penambahan fungsional untuk melakukan pelaporan untuk aktor mahasiswa disabilitas dan *volunteer*.
9. Penambahan fungsional untuk melihat pelaporan masuk serta respon pelaporan masuk untuk aktor admin.

Evaluasi iterasi pertama yang dilakukan memberikan hasil perlu penambahan fungsional maupun fitur untuk mendukung proses pendampingan mahasiswa disabilitas di lingkungan Universitas Brawijaya. Untuk mengakomodir penambahan fungsional maupun fitur dari perangkat lunak pendampingan mahasiswa disabilitas, tahap analisis kebutuhan akan dilanjutkan ke tahap iterasi kedua.

4.5 Iterasi Kedua

Pada iterasi kedua, terdapat beberapa perubahan pada fungsionalitas perangkat lunak pendampingan mahasiswa disabilitas. Perubahan yang dilakukan pada iterasi kedua berdampak pada perubahan beberapa daftar kebutuhan, pemodelan kebutuhan serta analisis data dari perangkat lunak pendampingan mahasiswa disabilitas.

4.5.1 Daftar Kebutuhan Fungsional

4.5.1.1 Kebutuhan Fungsional

Pada kebutuhan fungsional perangkat lunak pendampingan mahasiswa disabilitas pada iterasi kedua, terdapat perubahan yang dilakukan pada beberapa kebutuhan yang ada. Perubahan terjadi pada fungsionalitas yang digunakan oleh *admin*, mahasiswa disabilitas dan mahasiswa *volunteer*. Pada aktor *admin* terdapat penambahan fungsionalitas untuk generate jadwal pendampingan, lihat pengajuan pendampingan tugas akhir, lihat detail pengajuan pendampingan tugas akhir, pilih pendamping pengganti, lihat pelaporan, lihat detail pelaporan masuk dan balas laporan. Pada sisi mahasiswa disabilitas, terdapat penambahan fungsionalitas antara lain lihat pelaporan, lihat detail pelaporan, balas laporan, tulis laporan, ajukan pendampingan tugas akhir, lihat detail pendampingan tugas akhir dan konfirmasi *logbook*. Sedangkan pada sisi mahasiswa *volunteer*,

terdapat penambahan fungsionalitas antara lain lihat pelaporan, lihat detail pelaporan, balas pelaporan, tulis pelaporan, lihat detail pendampingan tugas akhir dan ajukan *logbook*.

A. Kebutuhan Fungsional Admin

Pada iterasi kedua, aktor *admin* mendapat penambahan fungsionalitas. Penambahan fungsionalitas tersebut antara lain terdapat fungsionalitas untuk generate jadwal pendampingan, lihat pengajuan pendampingan tugas akhir, lihat detail pengajuan pendampingan tugas akhir, pilih pendamping pengganti, lihat pelaporan masuk, lihat detail pelaporan masuk dan balas laporan. Kebutuhan fungsional aktor *admin* pada iterasi kedua dapat dilihat pada Tabel 4.32.

Tabel 4.32 Kebutuhan Fungsional Iterasi Kedua Admin

No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
1	DIMENT_1_003	Lihat pendaftaran.	Sistem mampu menyediakan fungsi untuk melihat pendaftaran oleh mahasiswa.
2	DIMENT_1_004	Lihat detail pendaftaran.	Sistem mampu menyediakan fungsi untuk melihat detail pendaftaran.
3	DIMENT_1_006	Tambah jadwal pendampingan.	Sistem mampu menyediakan fungsi tambah jadwal pendampingan mahasiswa disabilitas.
4	DIMENT_1_007	Hapus jadwal pendampingan.	Sistem mampu menyediakan fungsi hapus jadwal pendampingan mahasiswa disabilitas.
5	DIMENT_1_008	Lihat pengajuan izin.	Sistem mampu menyediakan fungsi untuk melihat pengajuan izin pendampingan.
6	DIMENT_1_009	Lihat pengajuan pendampingan skripsi.	Sistem mampu menyediakan fungsi untuk melihat pengajuan pendampingan skripsi.
7	DIMENT_1_010	Lihat detail pengajuan pendampingan skripsi.	Sistem mampu menyediakan fungsi untuk melihat detail pengajuan pendampingan skripsi.
8	DIMENT_1_011	Pilih pendamping skripsi.	Sistem mampu menyediakan fungsi pilih pendamping skripsi untuk mahasiswa disabilitas.



No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
9	DIMENT_1_012	Lihat jadwal Mahasiswa.	Sistem mampu menyediakan fungsi untuk melihat jadwal mahasiswa.
10	DIMENT_1_013	Ubah jadwal Mahasiswa.	Sistem mampu menyediakan fungsi untuk ubah jadwal mahasiswa.
11	DIMENT_1_024	Generate jadwal pendampingan.	Sistem mampu menyediakan fungsi untuk generate jadwal pendampingan.
12	DIMENT_1_025	Lihat pengajuan pendampingan tugas akhir.	Sistem mampu menyediakan fungsi untuk melihat pengajuan pendampingan tugas akhir.
13	DIMENT_1_026	Lihat detail pengajuan pendampingan tugas akhir.	Sistem mampu menyediakan fungsi untuk melihat detail pengajuan pendampingan tugas akhir.
14	DIMENT_1_027	Pilih pendamping pengganti.	Sistem mampu menyediakan fungsi untuk memilih pendamping pengganti untuk jadwal pendampingan yang telah diajukan izin.
15	DIMENT_1_028	Lihat pelaporan.	Sistem mampu menyediakan fungsi untuk melihat pelaporan mahasiswa.
16	DIMENT_1_029	Lihat detail pelaporan.	Sistem mampu menyediakan fungsi untuk melihat detail pelaporan masuk dari mahasiswa disabilitas dan mahasiswa <i>volunteer</i> .
17	DIMENT_1_030	Balas laporan.	Sistem mampu menyediakan fungsi untuk membalas laporan yang masuk.
18	DIMENT_1_036	<i>Logout</i> .	Sistem mampu menyediakan fungsi logout.
19	DIMENT_1_037	Lihat pengajuan pendampingan perkuliahan.	Sistem mampu menyediakan fungsi untuk melihat pengajuan pendampingan perkuliahan yang dilakukan mahasiswa disabilitas.
20	DIMENT_1_038	Lihat detail	Sistem mampu menyediakan



No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
		pengajuan pendampingan perkuliahan.	fungsi untuk melihat detail pengajuan pendampingan perkuliahan yang dilakukan mahasiswa disabilitas.
21	DIMENT_1_039	Terima pengajuan pendampingan perkuliahan.	Sistem mampu menyediakan fungsi untuk terima pengajuan pendampingan perkuliahan yang diajukan mahasiswa disabilitas.
22	DIMENT_1_040	<i>Login Admin</i>	Sistem mampu menyediakan fungsi <i>login</i> untuk <i>admin</i> .
23	DIMENT_1_042	Pilih Pendamping Tugas Akhir	Sistem mampu menyediakan fungsi pilih pendamping tugas akhir untuk mahasiswa disabilitas.

B. Kebutuhan Fungsional Mahasiswa Disabilitas

Pada iterasi kedua, aktor mahasiswa disabilitas mendapatkan penambahan kebutuhan fungsional. Penambahan kebutuhan fungsional untuk mahasiswa disabilitas antara lain lihat detail pelaporan, balas laporan, tulis pelaporan, ajukan pendampingan tugas akhir, lihat detail pendampingan tugas akhir dan konfirmasi *logbook*. Kebutuhan fungsional mahasiswa disabilitas pada iterasi kedua dapat dilihat pada Tabel 4.33.

Tabel 4.33 Kebutuhan Fungsional Iterasi Kedua Mahasiswa Disabilitas

No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
1	DIMENT_1_014	Tambah jadwal.	Sistem mampu menyediakan fungsi tambah jadwal perkuliahan mahasiswa disabilitas.
2	DIMENT_1_015	Hapus jadwal.	Sistem mampu menyediakan fungsi hapus jadwal perkuliahan mahasiswa disabilitas.
3	DIMENT_1_016	Lihat pendampingan perkuliahan.	Sistem mampu menyediakan fungsi melihat jadwal pendampingan mahasiswa disabilitas.
4	DIMENT_1_017	Lihat detail jadwal pendampingan.	Sistem mampu menyediakan fungsi melihat detail jadwal pendampingan mahasiswa disabilitas.



No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
5	DIMENT_1_018	Ubah profil.	Sistem mampu menyediakan fungsi untuk mengubah profil mahasiswa disabilitas.
6	DIMENT_1_019	Izin pendampingan.	Sistem mampu menyediakan fungsi untuk mengajukan izin pendampingan.
7	DIMENT_1_020	Ajukan pendampingan skripsi.	Sistem mampu menyediakan fungsi untuk mengajukan pendampingan skripsi.
8	DIMENT_1_021	Lihat detail pendampingan skripsi.	Sistem mampu menyediakan fungsi untuk melihat detail pendampingan skripsi.
9	DIMENT_1_022	Tampilkan <i>qr code</i> .	Sistem mampu menyediakan fungsi untuk menampilkan <i>qr code</i> pendampingan.
10	DIMENT_1_028	Lihat pelaporan.	Sistem mampu menyediakan fungsi untuk melihat pelaporan mahasiswa.
11	DIMENT_1_029	Lihat detail pelaporan.	Sistem mampu menyediakan fungsi untuk melihat detail pelaporan yang telah ditulis.
12	DIMENT_1_030	Balas laporan.	Sistem mampu menyediakan fungsi untuk membalas laporan.
13	DIMENT_1_031	Tulis pelaporan.	Sistem menyediakan fungsi untuk menulis pelaporan untuk mahasiswa disabilitas.
14	DIMENT_1_032	Ajukan pendampingan tugas akhir.	Sistem mampu menyediakan fungsi untuk mengajukan pendampingan tugas akhir.
15	DIMENT_1_033	Lihat detail pendampingan tugas akhir.	Sistem mampu menyediakan fungsi untuk melihat detail pendampingan tugas akhir.
16	DIMENT_1_034	Konfirmasi <i>logbook</i> .	Sistem mampu menyediakan fungsi konfirmasi <i>logbook</i> untuk pendampingan skripsi dan pendampingan tugas akhir.
17	DIMENT_1_036	Logout.	Sistem mampu menyediakan fungsi logout.



No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
18	DIMENT_1_041	Lihat jadwal.	Sistem mampu menyediakan fungsi untuk melihat jadwal dari mahasiswa.

C. Kebutuhan Fungsional Mahasiswa *Volunteer*

Pada iterasi kedua, aktor mahasiswa *volunteer* mendapatkan penambahan kebutuhan fungsional. Penambahan kebutuhan fungsional pada iterasi kedua antara lain lihat detail pelaporan, balas laporan, tulis pelaporan, lihat detail pendampingan tugas akhir dan ajukan *logbook*. Kebutuhan fungsional untuk aktor mahasiswa *volunteer* pada iterasi kedua dapat dilihat pada Tabel 4.34.

Tabel 4.34 Kebutuhan Fungsional Iterasi Kedua Mahasiswa *Volunteer*

No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
1	DIMENT_1_014	Tambah jadwal.	Sistem mampu menyediakan fungsi tambah jadwal perkuliahan mahasiswa disabilitas.
2	DIMENT_1_015	Hapus jadwal.	Sistem mampu menyediakan fungsi hapus jadwal perkuliahan mahasiswa disabilitas.
3	DIMENT_1_016	Lihat pendampingan perkuliahan.	Sistem mampu menyediakan fungsi melihat jadwal pendampingan mahasiswa disabilitas.
4	DIMENT_1_017	Lihat detail jadwal pendampingan.	Sistem mampu menyediakan fungsi melihat detail jadwal pendampingan mahasiswa disabilitas.
5	DIMENT_1_018	Ubah profil.	Sistem mampu menyediakan fungsi untuk mengubah profil mahasiswa disabilitas.
6	DIMENT_1_019	Izin pendampingan.	Sistem mampu menyediakan fungsi untuk mengajukan izin pendampingan.
7	DIMENT_1_021	Lihat detail pendampingan skripsi.	Sistem mampu menyediakan fungsi untuk melihat detail pendampingan skripsi.
8	DIMENT_1_023	Mulai Pendampingan.	Sistem mampu menyediakan fungsi untuk memulai pendampingan.



No	Kode Kebutuhan	Nama Use Case	Deskripsi Kebutuhan
9	DIMENT_1_028	Lihat pelaporan.	Sistem mampu menyediakan fungsi untuk melihat pelaporan mahasiswa.
10	DIMENT_1_029	Lihat detail pelaporan.	Sistem mampu menyediakan fungsi untuk melihat detail pelaporan yang telah ditulis.
11	DIMENT_1_030	Balas laporan.	Sistem mampu menyediakan fungsi untuk membalas laporan.
12	DIMENT_1_031	Tulis pelaporan.	Sistem menyediakan fungsi untuk menulis pelaporan untuk mahasiswa disabilitas.
13	DIMENT_1_033	Lihat detail pendampingan tugas akhir.	Sistem mampu menyediakan fungsi untuk melihat detail pendampingan tugas akhir.
14	DIMENT_1_035	Ajukan <i>logbook</i> .	Sistem mampu menyediakan fungsi ajukan <i>logbook</i> untuk pendampingan skripsi dan pendampingan tugas akhir.
15	DIMENT_1_036	Logout	Sistem mampu menyediakan fungsi logout.
18	DIMENT_1_041	Lihat jadwal.	Sistem mampu menyediakan fungsi untuk melihat jadwal dari mahasiswa.

4.5.1.2 Spesifikasi Kebutuhan

Pada iterasi kedua, terdapat penambahan kebutuhan fungsional pada perangkat lunak pendampingan mahasiswa disabilitas. Penambahan kebutuhan fungsional berdampak kepada penambahan spesifikasi kebutuhan. Penjelasan mengenai spesifikasi kebutuhan dari masing – masing kebutuhan fungsional dari perangkat lunak pendampingan mahasiswa dapat di lihat pada Tabel 4.35.

Tabel 4.35 Spesifikasi Kebutuhan

No	Kode Kebutuhan	Kode Spesifikasi	Deskripsi
1	DIMENT_1_004	DIMENT_1_004_01	Data pendaftaran yang ditampilkan meliputi foto profil, nama, fakultas, semester, tanggal lahir, angkatan, nomor <i>handphone</i> , email serta berkas pendaftaran.



No	Kode Kebutuhan	Kode Spesifikasi	Deskripsi
2	DIMENT_1_006	DIMENT_1_006_01	Data masukan yang diperlukan meliputi jadwal pilihan, tanggal pendampingan serta pilihan pendamping.
3	DIMENT_1_008	DIMENT_1_008_01	Data yang ditampilkan meliputi nama mahasiswa, hari, nama mata kuliah, ruangan, jam mulai, jam selesai, alasan serta pilihan pengganti.
4	DIMENT_1_008	DIMENT_1_008_02	Pilihan pendamping pengganti yang ditampilkan adalah mahasiswa yang tidak memiliki jadwal pada jam pendampingan.
5	DIMENT_1_010	DIMENT_1_010_01	Data yang ditampilkan meliputi nama mahasiswa, judul skripsi, nama pembimbing, waktu mulai, waktu selesai serta status pengajuan.
6	DIMENT_1_010	DIMENT_1_010_02	Pilihan pendamping yang ditampilkan adalah mahasiswa yang belum memiliki pendampingan skripsi.
7	DIMENT_1_013	DIMENT_1_013_01	Data yang bisa diubah meliputi keterangan jadwal, hari, ruangan, jam mulai dan jam selesai.
8	DIMENT_1_014	DIMENT_1_014_01	Data yang dimasukan meliputi keterangan jadwal, hari, ruangan, jam mulai dan jam selesai.
9	DIMENT_1_017	DIMENT_1_017_01	Data yang ditampilkan meliputi foto mahasiswa, nama mahasiswa, hari, nama mata kuliah, ruangan jam mulai dan jam selesai.
10	DIMENT_1_018	DIMENT_1_018_01	Data yang dapat diubah meliputi nama, tanggal lahir, angkatan, semester, nomor <i>handphone</i> serta alamat email.
11	DIMENT_1_020	DIMENT_1_020_01	Data yang dimasukan meliputi judul skripsi, nama pembimbing, waktu mulai, waktu selesai serta



No	Kode Kebutuhan	Kode Spesifikasi	Deskripsi
			dokumen pendukung.
12	DIMENT_1_021	DIMENT_1_021_01	Data yang ditampilkan meliputi waktu, judul skripsi, dosen pembimbing, fakultas serta nomor <i>handphone</i> .
13	DIMENT_1_021	DIMENT_1_021_01	Data <i>logbook</i> yang ditampilkan meliputi tanggal pendampingan, jam, keterangan serta status pendampingan skripsi.
14	DIMENT_1_024	DIMENT_1_024_01	Data yang dimasukan meliputi tanggal mulai dan tanggal selesai.
15	DIMENT_1_024	DIMENT_1_024_02	Data tanggal mulai dan tanggal selesai berformat "bulan-tanggal-tahun".
16	DIMENT_1_026	DIMENT_1_026_01	Data yang ditampilkan meliputi nama mahasiswa, judul tugas akhir, waktu mulai, waktu selesai serta status pengajuan.
17	DIMENT_1_026	DIMENT_1_026_02	Pilihan pendamping yang ditampilkan adalah mahasiswa yang tidak memiliki jadwal pendampingan tugas akhir dengan mahasiswa lain.
18	DIMENT_1_027	DIMENT_1_027_01	Pilihan pendamping yang ditampilkan adalah mahasiswa selain mahasiswa yang mengajukan izin serta mahasiswa yang tidak memiliki jadwal pada jadwal pendampingan tersebut.
19	DIMENT_1_029	DIMENT_1_029_01	Data yang ditampilkan meliputi judul laporan, nama pelapor, status, tanggal pengajuan, isi laporan, lampiran serta balasan.
20	DIMENT_1_030	DIMENT_1_030_01	Data yang dimasukan meliputi isi balasan serta dokumen pendukung.
21	DIMENT_1_031	DIMENT_1_031_01	Data yang dimasukan meliputi judul laporan, isi laporan serta dokumen pendukung.



No	Kode Kebutuhan	Kode Spesifikasi	Deskripsi
22	DIMENT_1_032	DIMENT_1_032_01	Data yang dimasukan meliputi judul tugas akhir, waktu mulai, waktu selesai serta dokumen pendukung.
23	DIMENT_1_033	DIMENT_1_033_01	Data yang ditampilkan meliputi waktu, judul tugas akhir, fakultas nomor <i>handphone</i> , serta <i>logbook</i> pendampingan
24	DIMENT_1_033	DIMENT_1_033_02	Data <i>logbook</i> yang ditampilkan meliputi tanggal pendampingan, jam, keterangan serta status pendampingan tugas akhir.
25	DIMENT_1_035	DIMENT_1_035_01	Data yang dimasukan meliputi tanggal pendampingan, jam mulai, jam selesai serta keterangan pendampingan.
26	DIMENT_1_035	DIMENT_1_035_02	Sistem mencatat lokasi pengajuan <i>logbook</i> dan menyimpan lokasi ke <i>database</i> .
27	DIMENT_1_038	DIMENT_1_038_01	Data yang ditampilkan meliputi tanggal pengajuan, alat bantu serta berkas pendukung.
28	DIMENT_1_040	DIMENT_1_040_01	Data yang dimasukan meliputi <i>email</i> dan <i>password</i> .

4.5.2 Pemodelan Kebutuhan

Pada iterasi kedua, terdapat penambahan kebutuhan fungsional pada aktor *admin*, mahasiswa disabilitas dan mahasiswa *volunteer*. Penambahan kebutuhan fungsional tersebut berdampak kepada perubahan pemodelan kebutuhan dari perangkat lunak pendampingan mahasiswa disabilitas. Pada pemodelan kebutuhan, akan ada perubahan pada *use case diagram* dan *case scenario*.

4.5.2.1 Use Case Diagram

Pada iterasi kedua, terdapat perubahan pada *use case diagram*. Pada iterasi pertama, perangkat lunak pendampingan mahasiswa disabilitas memiliki 24 *use case*, sedangkan pada iterasi kedua, terdapat penambahan 12 *use case* sehingga *use case* dari perangkat lunak pendampingan mahasiswa disabilitas berjumlah 36. *Use case diagram* dari perangkat lunak pendampingan mahasiswa disabilitas dapat dilihat dalam Gambar 4.3 berikut.



67



4.5.2.2 Use Case Scenario

Pada iterasi kedua, terdapat penambahan *use case* dimana pada iterasi pertama perangkat lunak pendampingan mahasiswa disabilitas memiliki *use case* berjumlah 24, sedangkan pada iterasi kedua jumlah *use case* yaitu 33. Penambahan *use case* ini berdampak pada penambahan *use case scenario* untuk menjelaskan skenario dari kebutuhan fungsional yang baru ditambahkan. *Use case scenario* iterasi kedua dari perangkat lunak pendampingan mahasiswa disabilitas akan dijelaskan pada Tabel 4.36 sampai Tabel 4.53.

A. Use Case Scenario Login Admin

Pada fungsionalitas *login admin*, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas login memiliki fungsi untuk melakukan autentikasi dari *admin*. Skenario yang dijalankan adalah perangkat lunak melakukan pengecekan terhadap masukan yang diberikan antara lain *email* dan *password*. Kemudian jika data ditemukan maka akan melakukan autentikasi terhadap *admin* dan halaman *dashboard* terbuka. *Use case scenario login admin* dapat dilihat pada Tabel 4.36.

Tabel 4.36 Use Case Scenario Login Admin

Nama Use Case	<i>Login Admin</i> (DIMENT_1_040).
Objective	<i>Admin</i> dapat melakukan <i>login</i> pada perangkat lunak.
Aktor	<i>Admin</i> .
Pre-condition	1. <i>Admin</i> membuka halaman <i>login admin</i>. 2. Halaman <i>login admin</i> ditampilkan.
Main Flow	1. <i>Admin</i> mengisi <i>email</i> dan <i>password</i>. 2. <i>Admin</i> menekan tombol <i>login</i>. 3. Sistem melakukan validasi apakah data masukan yang diberikan sesuai dengan data yang tersimpan pada <i>database</i>. 4. Sistem memulai sesi <i>login</i> dari <i>admin</i>.
Alternative Flow	1. Perangkat lunak menampilkan informasi "Please fill out this field" jika terdapat <i>field</i> yang kosong. 2. Perangkat lunak menampilkan informasi "Kami tidak menemukan akun terkait" jika data yang dimasukan salah.
Post-condition	<i>Admin</i> terautentikasi dan sistem menampilkan halaman <i>dashboard</i> .

B. Use Case Scenario Generate Jadwal Pendampingan

Pada fungsionalitas generate jadwal pendampingan, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas generate jadwal pendampingan memiliki fungsi untuk melakukan penjadwalan pendampingan



otomatis pada rentang waktu yang telah ditentukan. Skenario yang dijalankan adalah *admin* membuka menu jadwal pendampingan dan menekan tombol generate jadwal pendampingan, kemudian *admin* akan memilih waktu mulai dan waktu selesai dan menekan tombol generate. Perangkat lunak akan melakukan pencocokan jadwal antara mahasiswa disabilitas dan mahasiswa *volunteer* dan menampilkan hasil pencocokan jadwal tersebut. Kemudian *admin* menekan tombol simpan generate jadwal. *Use case scenario* generate jadwal pendampingan dapat dilihat pada Tabel 4.37.

Tabel 4.37 Use Case Scenario Generate Jadwal Pendampingan

Nama <i>Use Case</i>	Generate jadwal pendampingan (DIMENT_1_024).
<i>Objective</i>	<i>Admin</i> dapat melakukan penjadwalan pendampingan mahasiswa disabilitas secara otomatis pada rentang waktu yang ditentukan.
Aktor	<i>Admin</i> .
<i>Pre-condition</i>	1. <i>Admin</i> membuka halaman jadwal pendampingan. 2. Sistem mengambil data jadwal pendampingan dari <i>database</i>. 2. Halaman jadwal pendampingan ditampilkan.
<i>Main Flow</i>	1. <i>Admin</i> menekan tombol generate jadwal. 2. Sistem menampilkan <i>form</i> pengisian waktu mulai dan selesai. 3. <i>Admin</i> mengisi waktu mulai dan selesai. 4. <i>Admin</i> menekan tombol generate. 5. Sistem melakukan validasi apabila terdapat jadwal pendampingan diantara waktu yang dipilih. 6. Sistem mengambil data jadwal mahasiswa disabilitas dan <i>volunteer</i> dari <i>database</i>. 7. Sistem melakukan pencocokan antara jadwal mahasiswa disabilitas dan mahasiswa <i>volunteer</i>. 8. Sistem menyimpan data hasil <i>generate</i> ke <i>database</i>.
<i>Alternative Flow</i>	1. Perangkat lunak menampilkan informasi "Telah terdapat jadwal pendampingan pada tanggal yang dipilih" jika terdapat jadwal pendampingan pada rentang waktu yang dipilih <i>admin</i>. 2. Sistem menampilkan informasi "<i>Please fill out this field</i>" apabila terdapat data kosong.
<i>Post-condition</i>	Hasil generate jadwal pendampingan disimpan ke <i>database</i> dan halaman jadwal pendampingan ditampilkan.



C. Use Case Scenario Lihat Pengajuan Pendampingan Tugas Akhir

Pada fungsionalitas lihat pengajuan pendampingan tugas akhir, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas lihat pengajuan pendampingan tugas akhir memiliki fungsi untuk melihat daftar pengajuan pendampingan tugas akhir yang dilakukan masing – masing mahasiswa disabilitas. Skenario yang dijalankan adalah *admin* membuka menu pengajuan pendampingan tugas akhir. Setelah itu daftar pengajuan pendampingan tugas akhir akan ditampilkan. *Use case scenario* lihat pengajuan pendampingan tugas akhir dapat dilihat pada Tabel 4.38.

Tabel 4.38 Use Case Scenario Lihat Pengajuan Pendampingan Tugas Akhir

Nama Use Case	Lihat pengajuan pendampingan tugas akhir (DIMENT_1_025).
Objective	<i>Admin</i> dapat melihat pengajuan pendampingan tugas akhir yang dilakukan oleh mahasiswa disabilitas.
Aktor	<i>Admin</i> .
Pre-condition	1. <i>Admin</i> telah terautentikasi.
Main Flow	1. <i>Admin</i> membuka menu pengajuan pendampingan tugas akhir. 2. Sistem mengambil data pendampingan tugas akhir dengan status diajukan dari <i>database</i> .
Alternative Flow	
Post-condition	Halaman pengajuan pendampingan tugas akhir ditampilkan.

D. Use Case Scenario Lihat Detail Pengajuan Pendampingan Tugas Akhir

Pada fungsionalitas lihat detail pengajuan pendampingan tugas akhir, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas lihat detail pengajuan pendampingan tugas akhir memiliki fungsi untuk melihat detail pengajuan pendampingan tugas akhir yang dilakukan masing – masing mahasiswa disabilitas. Skenario yang dijalankan adalah *admin* membuka menu pengajuan pendampingan tugas akhir, kemudian memilih pengajuan yang ada dan menekan tombol detail. Setelah itu detail pengajuan pendampingan tugas akhir akan ditampilkan. *Use case scenario* lihat detail pengajuan pendampingan tugas akhir dapat dilihat pada Tabel 4.39.

Tabel 4.39 Use Case Scenario Lihat Detail Pengajuan Pendampingan Tugas Akhir

Nama Use Case	Lihat detail pengajuan pendampingan tugas akhir (DIMENT_1_026).
Objective	<i>Admin</i> dapat melihat detail pengajuan pendampingan tugas akhir yang dilakukan oleh mahasiswa disabilitas.
Aktor	<i>Admin</i> .



<i>Pre-condition</i>	1. <i>Admin</i> telah terautentikasi. 2. Halaman daftar pengajuan pendampingan telah ditampilkan
<i>Main Flow</i>	1. <i>Admin</i> menekan tombol lihat detail pengajuan pada pengajuan yang dipilih. 2. Sistem mengambil data detail pengajuan dari <i>database</i> sesuai dengan pengajuan pendampingan yang dipilih.
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Halaman detail pengajuan pendampingan tugas akhir ditampilkan.

E. Use Case Scenario Pilih Pendamping Pengganti

Pada fungsionalitas pilih pendamping pengganti, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas pilih pendamping pengganti memiliki fungsi untuk *admin* dapat memilih pendamping pengganti untuk jadwal pendampingan yang diajukan izin oleh mahasiswa. Skenario yang dijalankan adalah *admin* membuka menu pengajuan izin, kemudian memilih pengajuan yang ada dan menekan tombol detail. Setelah itu *admin* dapat memilih pendamping pengganti pada jadwal pendampingan tersebut. *Use case scenario* pilih pendamping pengganti dapat dilihat pada Tabel 4.40.

Tabel 4.40 Use Case Scenario Pilih Pendamping Pengganti

Nama <i>Use Case</i>	Pilih pendamping pengganti (DIMENT_1_027).
<i>Objective</i>	<i>Admin</i> dapat memilih pendamping pengganti pada jadwal pendampingan yang dilakukan izin yang dilakukan oleh mahasiswa.
Aktor	<i>Admin</i> .
<i>Pre-condition</i>	1. <i>Admin</i> telah terautentikasi. 2. Halaman pengajuan izin telah ditampilkan.
<i>Main Flow</i>	1. <i>Admin</i> menekan tombol detail pada pengajuan izin yang dipilih. 2. Sistem menampilkan detail jadwal yang dilakukan pengajuan izin serta menampilkan daftar mahasiswa <i>volunteer</i> yang sesuai untuk menjadi pendamping pengganti. 3. <i>Admin</i> memilih pendamping pengganti. 4. <i>Admin</i> menekan tombol simpan. 5. Sistem menyimpan perubahan pendamping dari jadwal pendampingan ke <i>database</i> .
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Pilihan pendamping pengganti berhasil dilakukan dan



perubahan disimpan ke *database*.

F. Use Case Scenario Lihat Pelaporan

Pada fungsionalitas lihat pelaporan masuk, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*, mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas lihat pelaporan masuk memiliki fungsi untuk aktor dapat melihat pelaporan yang diajukan oleh mahasiswa. Skenario yang dijalankan adalah aktor membuka pelaporan, kemudian daftar pelaporan masuk akan ditampilkan. *Use case scenario* lihat pelaporan dapat dilihat pada Tabel 4.41.

Tabel 4.41 Use Case Scenario Lihat Pelaporan

Nama Use Case	Lihat pelaporan (DIMENT_1_028).
Objective	Aktor dapat melihat pelaporan yang dikirim oleh mahasiswa.
Aktor	1. <i>Admin</i> . 2. Mahasiswa Disabilitas. 3. Mahasiswa <i>Volunteer</i> .
Pre-condition	1. Aktor telah terautentikasi.
Main Flow	1. Aktor menekan tombol pelaporan. 2. Sistem mengambil data pelaporan yang pernah diajukan di <i>database</i> .
Alternative Flow	-
Post-condition	Halaman pelaporan ditampilkan.

G. Use Case Scenario Lihat Detail Pelaporan

Pada fungsionalitas lihat detail pelaporan masuk, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*, mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas lihat detail pelaporan masuk memiliki fungsi untuk aktor dapat melihat detail pelaporan yang diajukan. Skenario yang dijalankan adalah aktor membuka pelaporan, kemudian memilih pelaporan yang ingin dilihat dengan cara menekan pelaporan tersebut. *Use case scenario* lihat detail pelaporan masuk dapat dilihat pada Tabel 4.42.

Tabel 4.42 Use Case Scenario Lihat Detail Pelaporan

Nama Use Case	Lihat detail pelaporan (DIMENT_1_029).
Objective	Aktor dapat melihat detail pelaporan yang dikirim oleh mahasiswa.
Aktor	1. <i>Admin</i> . 2. Mahasiswa Disabilitas. 3. Mahasiswa <i>Volunteer</i> .



<i>Pre-condition</i>	1. Aktor telah terautentikasi. 2. Halaman pelaporan telah ditampilkan.
<i>Main Flow</i>	1. Aktor menekan tautan pada pelaporan yang dipilih. 2. Sistem mengambil data detail pelaporan yang dipilih dari <i>database</i>.
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Detail pelaporan ditampilkan.

H. Use Case Scenario Balas Laporan

Pada fungsionalitas balas laporan, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*, mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas balas pelaporan memiliki fungsi untuk aktor dapat membalas laporan yang telah diajukan. Skenario yang dijalankan adalah aktor membuka pelaporan, kemudian memilih pelaporan yang ingin dibalas dengan cara menekan pelaporan tersebut, kemudian menuliskan balasan pada kolom respon dan mengunggah lampiran pendukung, kemudian menekan tombol balas. *Use case scenario* balas laporan dapat dilihat pada Tabel 4.43.

Tabel 4.43 Use Case Scenario Balas Laporan

Nama Use Case	Balas laporan (DIMENT_1_030).
<i>Objective</i>	Aktor dapat membalas laporan yang telah diajukan.
Aktor	1. <i>Admin</i>. 2. Mahasiswa disabilitas. 3. Mahasiswa <i>volunteer</i>.
<i>Pre-condition</i>	1. Aktor telah terautentikasi. 2. Halaman detail pelaporan telah ditampilkan.
<i>Main Flow</i>	1. Aktor mengisi balasan pada kolom respon. 2. Aktor mengunggah lampiran berkas. 3. Aktor menekan tombol balas. 4. Sistem melakukan validasi masukan yang diberikan aktor. 5. Sistem menyimpan data balasan pelaporan ke <i>database</i>.
<i>Alternative Flow</i>	1. Sistem menampilkan informasi " <i>Please fill out this field</i> " apabila terdapat data kosong.
<i>Post-condition</i>	Balasan berhasil dikirim dan disimpan ke <i>database</i> .

I. Use Case Scenario Tulis Laporan

Pada fungsionalitas tulis laporan, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas tulis laporan memiliki fungsi untuk aktor dapat menulis untuk diajukan ke *admin*. Skenario yang dijalankan adalah aktor membuka pelaporan,



kemudian menuliskan laporan pada kolom judul, laporan dan mengunggah lampiran pendukung, kemudian menekan tombol laporkan. *Use case scenario* tulis laporan dapat dilihat pada Tabel 4.44.

Tabel 4.44 Use Case Scenario Tulis Laporan

Nama Use Case	Tulis laporan (DIMENT_1_031).
Objective	Aktor dapat menulis laporan yang akan diajukan.
Aktor	1. Mahasiswa disabilitas. 2. Mahasiswa volunteer.
Pre-condition	1. Aktor telah terautentikasi. 2. Halaman pelaporan telah ditampilkan.
Main Flow	1. Aktor mengisi judul pada kolom judul. 2. Aktor mengisi laporan pada kolom laporan. 3. Aktor mengunggah lampiran berkas. 4. Aktor menekan tombol laporkan. 5. Sistem melakukan validasi masukan yang diberikan aktor. 6. Sistem menyimpan laporan yang diberikan aktor ke database.
Alternative Flow	1. Sistem menampilkan informasi "Please fill out this field" apabila terdapat data kosong.
Post-condition	Laporan berhasil dikirim dan disimpan ke database.

J. Use Case Scenario Ajukan Pendampingan Tugas Akhir

Pada fungsionalitas ajukan pendampingan tugas akhir, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas. Fungsionalitas ajukan pendampingan tugas akhir memiliki fungsi agar mahasiswa disabilitas dapat melakukan pengajuan pendampingan untuk proses tugas akhir yang dilaksanakannya. Skenario yang dijalankan adalah mahasiswa disabilitas mengisi data judul tugas akhir, waktu pengerjaan, mengunggah dokumen pengajuan pendampingan tugas akhir dan menekan tombol ajukan. *Use case scenario* ajukan pendampingan tugas akhir dapat dilihat pada Tabel 4.45.

Tabel 4.45 Use Case Scenario Ajukan Pendampingan Tugas Akhir

Nama Use Case	Ajukan pendampingan tugas akhir (DIMENT_1_032).
Objective	Mahasiswa disabilitas dapat mengajukan pendampingan tugas akhir.
Aktor	Mahasiswa disabilitas.
Pre-condition	1. Mahasiswa disabilitas membuka menu pendampingan tugas akhir. 2. Mahasiswa disabilitas menekan tombol ajukan.



	3. Halaman ajukan pendampingan tugas akhir ditampilkan.
<i>Main Flow</i>	1. Mahasiswa disabilitas mengisi data judul tugas akhir, waktu pengerjaan. 2. Mahasiswa disabilitas mengunggah dokumen permohonan pengajuan pendampingan tugas akhir. 3. Mahasiswa disabilitas menekan tombol ajukan. 4. Sistem melakukan validasi masukan yang diberikan aktor. 5. Sistem menyimpan data pengajuan pendampingan tugas akhir ke <i>database</i>.
<i>Alternative Flow</i>	1. Sistem menampilkan informasi “ <i>Please fill out this field</i> ” jika terdapat <i>field</i> yang kosong.
<i>Post-condition</i>	Sistem menampilkan pesan “Pengajuan pendampingan tugas akhir berhasil”.

K. Use Case Scenario Lihat Detail Pendampingan Tugas Akhir

Pada fungsionalitas lihat detail pendampingan tugas akhir, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas ajukan pendampingan tugas akhir memiliki fungsi agar mahasiswa disabilitas dan *volunteer* dapat melihat pendampingan tugas akhir yang akan dilaksanakan. Skenario yang dijalankan adalah mahasiswa disabilitas maupun *volunteer* membuka menu pendampingan tugas akhir dan sistem akan menampilkan detail pendampingan tugas akhir. *Use case scenario* detail pendampingan tugas akhir dapat dilihat pada Tabel 4.46.

Tabel 4.46 Use Case Scenario Lihat Detail Pendampingan Tugas Akhir

Nama Use Case	Lihat detail pendampingan tugas akhir (DIMENT_1_033).
<i>Objective</i>	Mahasiswa disabilitas maupun <i>volunteer</i> dapat melihat detail pendampingan tugas akhir yang akan dilaksanakan.
Aktor	1. Mahasiswa disabilitas. 2. Mahasiswa <i>volunteer</i>.
<i>Pre-condition</i>	1. Aktor telah terautentikasi.
<i>Main Flow</i>	1. Aktor menekan tombol menu pendampingan tugas akhir. 2. Sistem mengambil data pendampingan tugas akhir dari <i>database</i> sesuai dengan aktor yang terautentikasi.
<i>Alternative Flow</i>	1. Sistem menampilkan pesan “Saat ini kamu belum memiliki pendampingan tugas akhir.” apabila belum ada pendampingan tugas akhir.
<i>Post-condition</i>	Halaman detail pendampingan tugas akhir ditampilkan.



L. Use Case Scenario Konfirmasi *Logbook*

Pada fungsionalitas konfirmasi *logbook*, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas. Fungsionalitas konfirmasi *logbook* memiliki fungsi agar mahasiswa disabilitas dapat melakukan konfirmasi *logbook* pendampingan tugas akhir maupun skripsi yang diajukan oleh mahasiswa *volunteer*. Skenario yang dijalankan adalah mahasiswa disabilitas membuka menu pendampingan tugas akhir maupun skripsi, kemudian menekan tombol konfirmasi pada *logbook* yang dipilih. *Use case scenario* konfirmasi *logbook* dapat dilihat pada Tabel 4.47.

Tabel 4.47 Use Case Scenario Konfirmasi *Logbook*

Nama Use Case	Konfirmasi <i>logbook</i> (DIMENT_1_034).
Objective	Mahasiswa disabilitas dapat melakukan konfirmasi <i>logbook</i> pendampingan tugas akhir maupun skripsi yang diajukan oleh mahasiswa <i>volunteer</i> .
Aktor	Mahasiswa disabilitas.
Pre-condition	1. Mahasiswa disabilitas telah terautentikasi. 2. Halaman detail pendampingan telah ditampilkan.
Main Flow	1. Mahasiswa disabilitas menekan tombol konfirmasi pada <i>logbook</i> yang dipilih. 2. Sistem menyimpan perubahan status dari <i>logbook</i> yang dipilih ke <i>database</i> .
Alternative Flow	
Post-condition	<i>Logbook</i> berhasil dikonfirmasi dan sistem menampilkan informasi “ <i>Logbook</i> berhasil dikonfirmasi”.

M. Use Case Scenario Ajukan *Logbook*

Pada fungsionalitas ajukan *logbook*, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa *volunteer*. Fungsionalitas ajukan *logbook* memiliki fungsi agar mahasiswa *volunteer* dapat melakukan pengajuan *logbook* pendampingan tugas akhir maupun skripsi yang telah dilaksanakan. Skenario yang dijalankan adalah mahasiswa *volunteer* membuka menu pendampingan tugas akhir maupun skripsi, kemudian menekan tombol isi *logbook*, kemudian mengisi data waktu mulai, waktu selesai dan keterangan, kemudian menekan tombol ajukan. *Use case scenario* ajukan *logbook* dapat dilihat pada Tabel 4.48.

Tabel 4.48 Use Case Scenario Ajukan *Logbook*

Nama Use Case	Ajukan <i>logbook</i> (DIMENT_1_035).
Objective	Mahasiswa <i>volunteer</i> dapat melakukan pengajuan <i>logbook</i> pendampingan tugas akhir maupun skripsi telah dilaksanakan.
Aktor	Mahasiswa <i>volunteer</i> .

<i>Pre-condition</i>	1. Mahasiswa <i>volunteer</i> telah terautentikasi. 2. Halaman detail pendampingan telah ditampilkan.
<i>Main Flow</i>	1. Mahasiswa <i>volunteer</i> menekan tombol isi <i>logbook</i>. 2. Sistem menampilkan <i>form</i> pengisian <i>logbook</i>. 3. Mahasiswa <i>volunteer</i> mengisi data waktu mulai, waktu selesai dan keterangan. 4. Mahasiswa <i>volunteer</i> menekan tombol ajukan. 5. Sistem mengambil data lokasi dari mahasiswa <i>volunteer</i>. 6. Sistem menyimpan data pengajuan <i>logbook</i> pendampingan ke <i>database</i>.
<i>Alternative Flow</i>	1. Sistem menampilkan informasi “<i>Please fill out this field</i>” jika terdapat <i>field</i> yang kosong.
<i>Post-condition</i>	<i>Logbook</i> berhasil diajukan dan sistem menampilkan informasi “ <i>Logbook</i> berhasil diajukan”.

N. Use Case Scenario Pilih Pendamping Tugas Akhir

Pada fungsionalitas pilih pendamping tugas akhir, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas pilih pendamping tugas akhir memiliki fungsi untuk memilih pendamping tugas akhir dari pengajuan pendampingan tugas akhir yang dilakukan masing – masing mahasiswa disabilitas. Skenario yang dijalankan adalah *admin* membuka menu pengajuan pendampingan tugas akhir, kemudian memilih pengajuan yang ada dan menekan tombol selengkapnya. Setelah itu halaman detail pengajuan pendampingan tugas akhir ditampilkan dan *admin* dapat memilih pendamping tugas akhir untuk mahasiswa disabilitas yang mengajukan pendampingan tugas akhir. *Use case scenario* pilih pendamping tugas akhir dapat dilihat pada Tabel 4.49.

Tabel 4.49 Use Case Scenario Pilih Pendamping Tugas Akhir

Nama Use Case	Pilih pendamping tugas akhir (DIMENT_1_042).
<i>Objective</i>	<i>Admin</i> dapat memilih pendamping untuk pengajuan pendampingan tugas akhir yang dilakukan oleh mahasiswa disabilitas.
Aktor	<i>Admin</i> .
<i>Pre-condition</i>	1. <i>Admin</i> telah terautentikasi. 2. Halaman pengajuan pendampingan tugas akhir ditampilkan.
<i>Main Flow</i>	1. <i>Admin</i> memilih pendamping untuk pengajuan pendampingan tugas akhir. 2. <i>Admin</i> menekan tombol simpan. 3. Sistem menyimpan data pendamping tugas akhir yang



	dipilih ke <i>database</i> .
<i>Alternative Flow</i>	
<i>Post-condition</i>	Pendamping tugas akhir dipilih dan disimpan ke <i>database</i> .

O. Use Case Scenario Lihat Pengajuan Pendampingan Perkuliahan

Pada fungsionalitas lihat pengajuan pendampingan perkuliahan, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas lihat pengajuan perkuliahan memiliki fungsi untuk melihat pengajuan pendampingan perkuliahan yang dilakukan masing – masing mahasiswa disabilitas. Skenario yang dijalankan adalah *admin* membuka menu pengajuan, kemudian memilih pendampingan perkuliahan. Setelah itu daftar pengajuan pendampingan perkuliahan akan ditampilkan. *Use case scenario* lihat detail pengajuan pendampingan tugas akhir dapat dilihat pada Tabel 4.50.

Tabel 4.50 Use Case Scenario Lihat Pengajuan Pendampingan Perkuliahan

Nama Use Case	Lihat pengajuan pendampingan perkuliahan (DIMENT_1_037).
<i>Objective</i>	<i>Admin</i> dapat melihat daftar pengajuan pendampingan perkuliahan yang dilakukan oleh mahasiswa disabilitas.
Aktor	<i>Admin</i> .
<i>Pre-condition</i>	1. <i>Admin</i> telah terautentikasi.
<i>Main Flow</i>	1. <i>Admin</i> membuka menu pengajuan dengan menekan tombol tutor -> pengajuan pada navigasi. 2. Sistem mengambil data pengajuan pendampingan dari <i>database</i>. 3. <i>Admin</i> membuka tab Perkuliahan. 4. Sistem melakukan penyaringan untuk menampilkan pengajuan perkuliahan.
<i>Alternative Flow</i>	
<i>Post-condition</i>	Daftar pengajuan pendampingan perkuliahan ditampilkan.

P. Use Case Scenario Lihat Detail Pengajuan Pendampingan Perkuliahan

Pada fungsionalitas lihat detail pengajuan pendampingan perkuliahan, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas lihat detail pengajuan perkuliahan memiliki fungsi untuk melihat detail pengajuan pendampingan perkuliahan yang dilakukan masing – masing mahasiswa disabilitas. Skenario yang dijalankan adalah *admin* membuka menu pengajuan, kemudian memilih pendampingan perkuliahan, kemudian memilih pengajuan yang ada dan menekan tombol detail. Setelah itu detail pengajuan pendampingan perkuliahan akan ditampilkan. *Use case scenario* lihat detail pengajuan pendampingan tugas akhir dapat dilihat pada Tabel 4.51.



Tabel 4.51 Use Case Scenario Lihat Detail Pengajuan Pendampingan Perkuliahan

Nama Use Case	Lihat detail pengajuan pendampingan perkuliahan (DIMENT_1_038).
Objective	Admin dapat melihat detail pengajuan pendampingan perkuliahan yang dilakukan oleh mahasiswa disabilitas.
Aktor	Admin.
Pre-condition	1. Admin telah terautentikasi. 2. Halaman daftar pengajuan pendampingan telah ditampilkan.
Main Flow	1. Admin menekan tombol lihat detail pengajuan pada pengajuan yang dipilih. 2. Sistem mengambil data detail pengajuan pendampingan perkuliahan yang dipilih dari database.
Alternative Flow	
Post-condition	Halaman detail pengajuan pendampingan perkuliahan ditampilkan.

Q. Use Case Scenario Terima Pengajuan Pendampingan Perkuliahan

Pada fungsionalitas terima pengajuan pendampingan perkuliahan, aktor yang dapat menjalankan fungsionalitas ini adalah *admin*. Fungsionalitas terima pengajuan perkuliahan memiliki fungsi untuk konfirmasi pengajuan pendampingan perkuliahan yang telah dilakukan mahasiswa disabilitas. Skenario yang dijalankan adalah *admin* membuka menu pengajuan, kemudian memilih pendampingan perkuliahan, kemudian memilih pengajuan yang ada dan menekan tombol detail. Setelah detail pengajuan pendampingan perkuliahan ditampilkan, *admin* menekan tombol konfirmasi. Use case scenario terima pengajuan pendampingan tugas akhir dapat dilihat pada Tabel 4.52.

Tabel 4.52 Use Case Scenario Terima Pengajuan Pendampingan Perkuliahan

Nama Use Case	Terima pengajuan pendampingan perkuliahan (DIMENT_1_039).
Objective	Admin dapat melakukan konfirmasi pengajuan pendampingan perkuliahan yang dilakukan oleh mahasiswa disabilitas.
Aktor	Admin.
Pre-condition	1. Admin telah terautentikasi. 2. Halaman detail pengajuan pendampingan telah ditampilkan



<i>Main Flow</i>	1. <i>Admin</i> menekan tombol konfirmasi untuk menerima pengajuan pendampingan perkuliahan. 2. Sistem menampilkan kotak dialog untuk melakukan konfirmasi dari aksi yang dilakukan <i>admin</i>. 3. <i>Admin</i> menekan tombol “Ya!” pada kotak konfirmasi. 4. Sistem melakukan perubahan status pengajuan pendampingan perkuliahan dan menyimpan perubahan ke <i>database</i>.
<i>Alternative Flow</i>	
<i>Post-condition</i>	Konfirmasi pengajuan pendampingan perkuliahan berhasil dilakukan dan sistem menampilkan informasi “Anda berhasil melakukan konfirmasi”.

R. Use Case Scenario Lihat Jadwal

Pada fungsionalitas lihat jadwal mahasiswa, aktor yang dapat menjalankan fungsionalitas ini adalah mahasiswa disabilitas dan mahasiswa *volunteer*. Fungsionalitas lihat jadwal memiliki fungsi untuk melihat jadwal yang dimiliki masing – masing mahasiswa. Skenario yang dijalankan adalah aktor membuka menu jadwal dan halaman jadwal ditampilkan. *Use case scenario* lihat jadwal dapat dilihat pada Tabel 4.53.

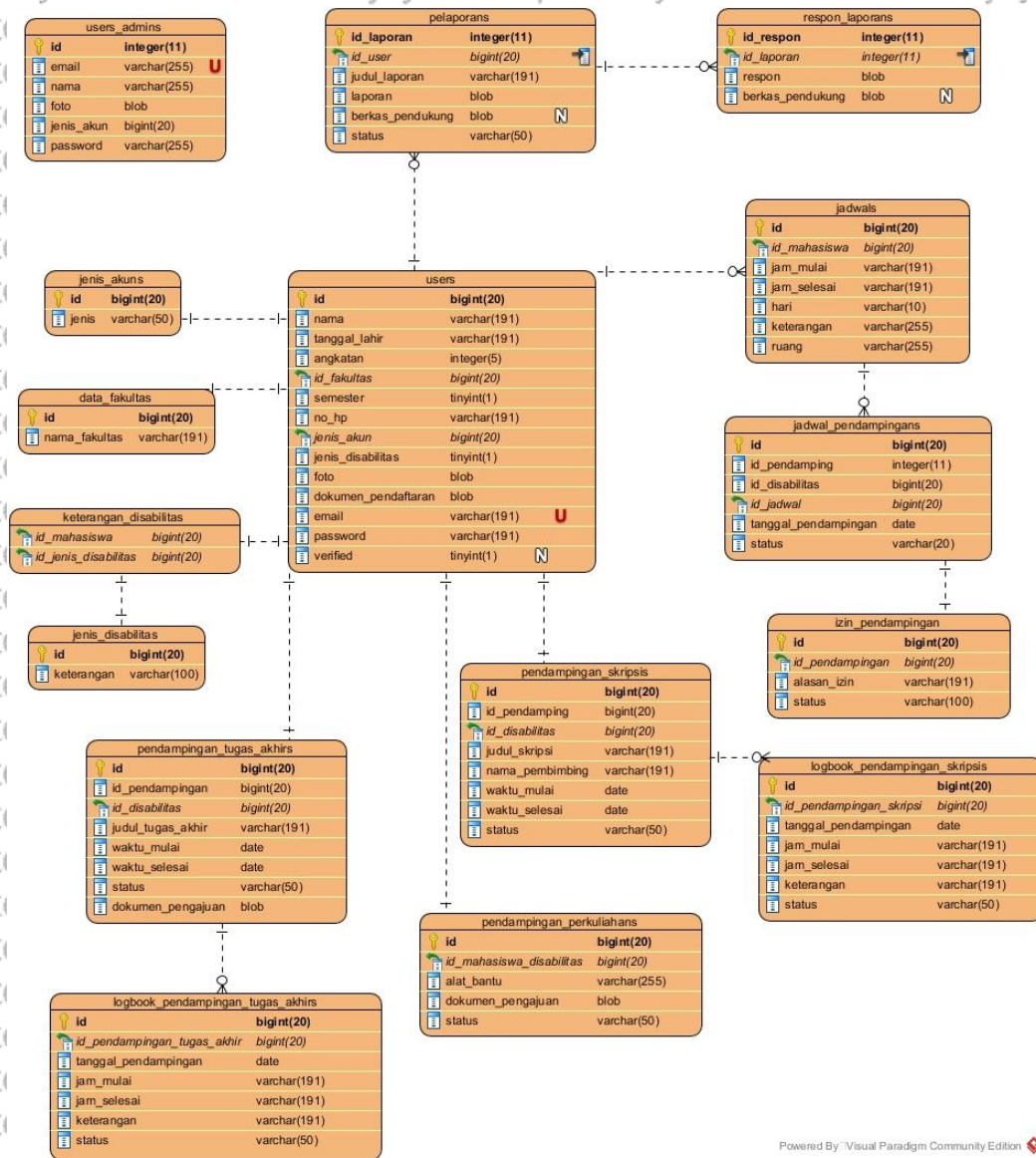
Tabel 4.53 Use Case Scenario Lihat Jadwal

Nama <i>Use Case</i>	Lihat jadwal (DIMENT_1_041).
<i>Objective</i>	Aktor dapat melihat jadwal dari masing – masing mahasiswa.
Aktor	1. Mahasiswa Disabilitas. 2. Mahasiswa <i>Volunteer</i>.
<i>Pre-condition</i>	1. Aktor telah terautentikasi.
<i>Main Flow</i>	1. Aktor membuka menu jadwal. 2. Sistem mengambil data jadwal mahasiswa dari <i>database</i> sesuai dengan aktor yang terautentikasi.
<i>Alternative Flow</i>	-
<i>Post-condition</i>	Halaman jadwal ditampilkan.

4.5.3 Analisis Data

Setelah kebutuhan perangkat lunak pendampingan mahasiswa disabilitas pada iterasi kedua didapatkan, langkah selanjutnya yang harus dilakukan adalah melakukan perbaikan analisis data. Perbaikan analisis data diperlukan untuk menunjang basis data hasil dari penambahan kebutuhan fungsional yang terjadi pada iterasi kedua. Pada iterasi kedua ini, terdapat 16 entitas dikarenakan

adanya penambahan entitas baru yaitu jenis_akun, pelaporans, respon_laporans, pendampingan_tugas_akhirs, logbook_pendampingan_tugas_akhirs, pendampingan_perkuliahans, data_fakultas, jenis_disabilitas, keterangan_disabilitas dan users_adminin. Analisis data iterasi kedua dari perangkat lunak pendampingan mahasiswa disabilitas dapat dilihat dalam Gambar 4.4.



Gambar 4.7 Entity Relationship Diagram Iterasi Kedua Perangkat Lunak Pendampingan Mahasiswa Disabilitas

4.5.4 Evaluasi

Pada iterasi kedua dari analisis kebutuhan perangkat lunak pendampingan mahasiswa disabilitas, telah dilakukan perubahan pada daftar kebutuhan dari perangkat lunak pendampingan mahasiswa disabilitas. Pada aktor *admin*



terdapat penambahan fungsionalitas untuk *generate* jadwal pendampingan, lihat pengajuan pendampingan tugas akhir, lihat detail pengajuan pendampingan tugas akhir, pilih pendamping pengganti, lihat pelaporan masuk, lihat detail pelaporan masuk dan balas laporan. Pada sisi mahasiswa disabilitas, terdapat penambahan fungsionalitas antara lain lihat detail pelaporan, balas laporan, tulis laporan, ajukan pendampingan tugas akhir, lihat detail pendampingan tugas akhir dan konfirmasi *logbook*. Sedangkan pada sisi mahasiswa *volunteer*, terdapat penambahan fungsionalitas antara lain lihat detail pelaporan, balas pelaporan, tulis pelaporan, lihat detail pendampingan tugas akhir dan ajukan *logbook*.

Setelah dilakukannya perubahan pada daftar kebutuhan perangkat lunak pendampingan mahasiswa disabilitas maka dilakukan perubahan kembali dari *use case diagram* yang ada. Perubahan ini dilakukan untuk menyesuaikan *use case diagram* dengan daftar kebutuhan yang ada. Setelah dilakukan perubahan pada *use case diagram*, maka dilakukan penambahan *use case scenario* untuk masing – masing kebutuhan baru yang ditambahkan pada iterasi kedua ini. Perubahan juga terjadi pada analisis data dimana adanya penambahan lima entitas untuk menunjang penambahan kebutuhan fungsional. Setelah dilakukannya iterasi kedua maka daftar kebutuhan perangkat lunak pendampingan mahasiswa disabilitas telah lengkap dan iterasi dapat diakhiri.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Perangkat Lunak

Pada bab perancangan, akan dilakukan perancangan sebagai landasan dalam pengembangan perangkat lunak pendampingan mahasiswa disabilitas. Dengan melakukan perancangan, akan diketahui bagaimana perangkat lunak akan dikembangkan sehingga dapat diketahui bagaimana hasil akhir dari suatu perangkat lunak baik itu dari sisi tampilan antarmuka, algoritma maupun dari sisi *backend*. Perancangan perangkat lunak dilakukan dengan menggunakan konsep perancangan berbasis objek (*Object oriented design*). Perancangan berbasis objek dilakukan dengan membuat pemodelan *sequence diagram*, pemodelan *class diagram*, perancangan algoritma dan perancangan antarmuka.

5.1.1 Perancangan Arsitektur

Perancangan arsitektur dilakukan untuk memberikan rancangan terhadap modul yang akan dikembangkan. Perancangan arsitektur akan dilakukan dengan melakukan pemodelan *sequence diagram*, pemodelan *class diagram* dan perancangan algoritma.

5.1.1.1 Pemodelan Sequence Diagram

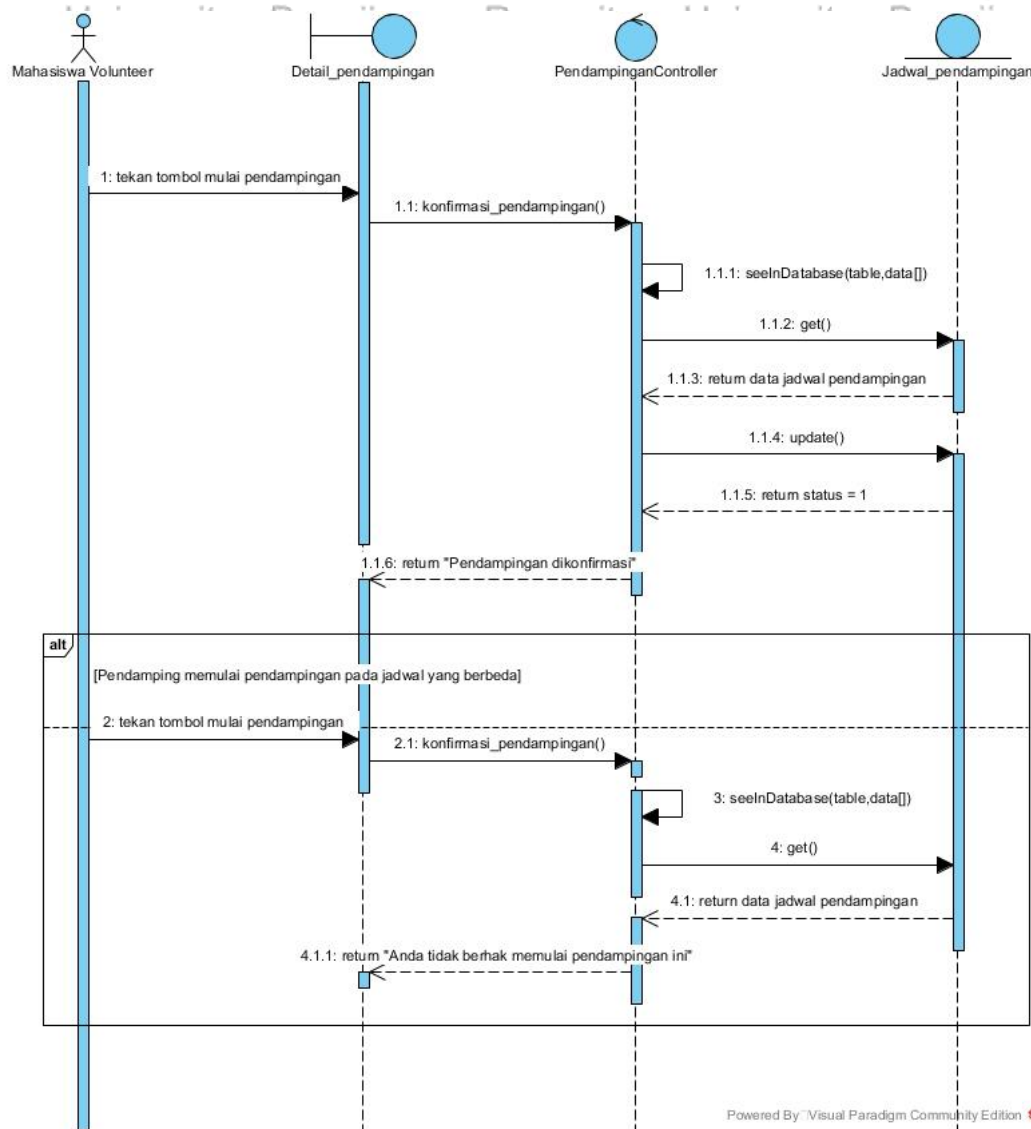
Sequence diagram akan digunakan dalam pengembangan perangkat lunak pendampingan mahasiswa disabilitas. *Sequence diagram* perlu digunakan karena kemampuannya dalam menggambarkan bagaimana interaksi yang dilakukan untuk mengirim pesan antar objeknya. Dengan menggunakan *sequence diagram*, akan mempermudah dalam memahami bagaimana suatu fungsionalitas bekerja dimulai dari aksi yang dipicu oleh aktor, kemudian perangkat lunak menjalankan suatu proses berdasar aksi yang diberikan aktor, sampai perangkat lunak menghasilkan keluaran yang ditampilkan kepada aktor. Selain itu, *sequence diagram* juga mampu menggambarkan bagaimana kondisi alternatif yang akan terjadi sesuai dengan aksi yang diberikan oleh aktor.

A. Sequence Diagram Mulai Pendampingan

Pada fungsionalitas mulai pendampingan, Fungsionalitas dijalankan oleh mahasiswa *volunteer*. Proses diawali dengan mahasiswa *volunteer* menekan tombol mulai pendampingan pada halaman detail_pendampingan. Ketika menekan tombol mulai pendampingan, maka akan membuat *method* konfirmasi_pendampingan() berjalan. Pada PendampinganController akan dilakukan proses cek apakah data pendampingan sesuai dengan *database* melalui dimana proses cek akan dilakukan oleh *seeInDatabase()*. *Method* *seeInDatabase()* akan memanggil *method* *get()* pada model Jadwal_pendampingan yang digunakan untuk mengambil data jadwal pendampingan. Jika data pendampingan sesuai maka PendampinganController akan memanggil *method* *update()* pada model Jadwal_pendampingan untuk melakukan perubahan status dari pendampingan yang akan dijalankan. Model



Jadwal_pendampingan akan mengembalikan status = 1 sebagai penanda untuk PendampinganController apabila proses *update* berhasil dilakukan dan PendampinganController akan menampilkan pesan "Pendampingan dikonfirmasi". Proses mulai pendampingan dapat dilihat dalam Gambar 5.2.



Powered By: Visual Paradigm Community Edition

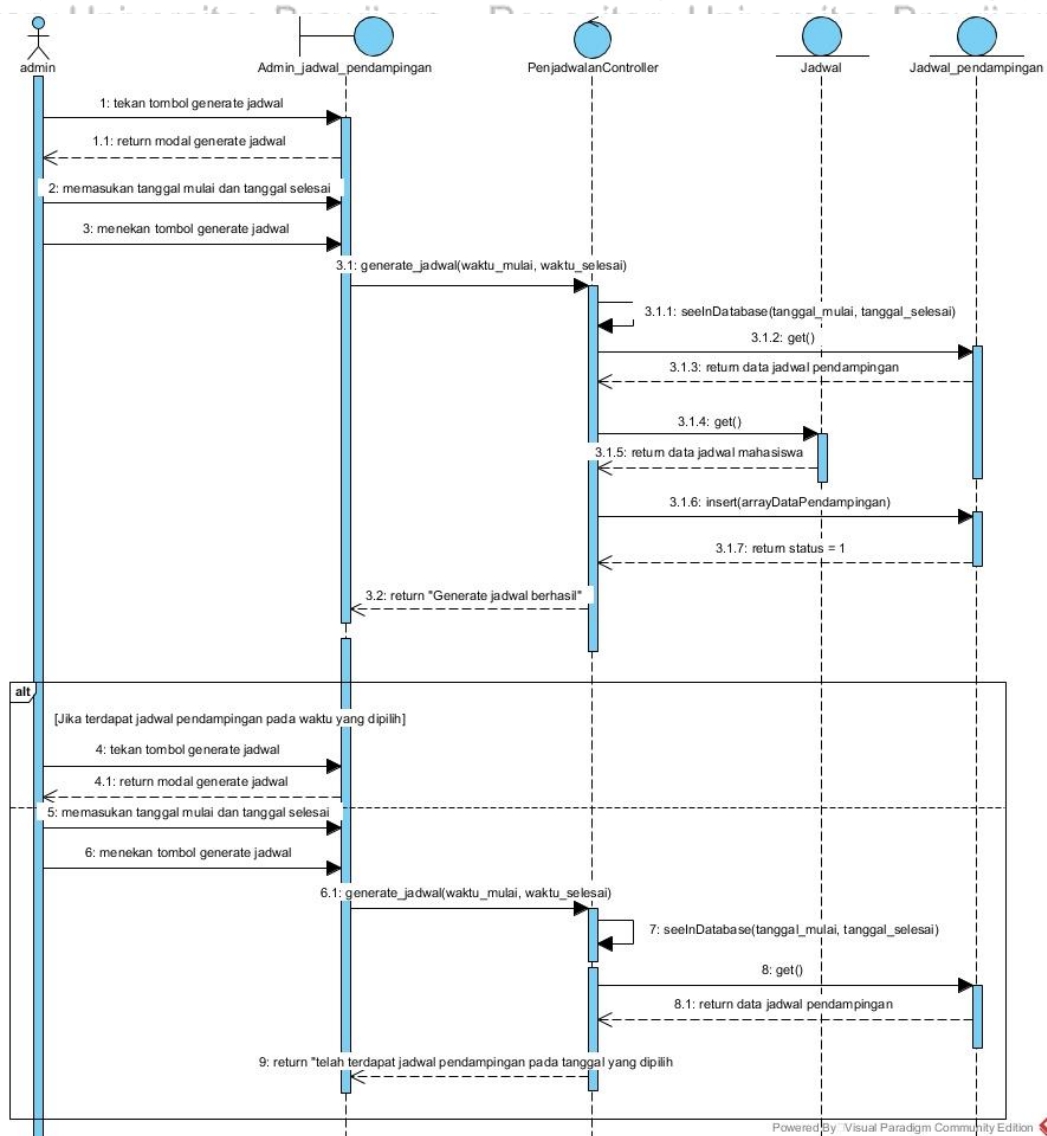
Gambar 5.1 Sequence Diagram Mulai Pendampingan

B. Sequence Diagram Generate Jadwal Pendampingan

Pada fungsionalitas generate jadwal, fungsionalitas dijalankan oleh admin. Proses diawali dengan ditampilkannya halaman Jadwal_pendampingan_admin dan admin menekan tombol generate jadwal. Ketika admin menekan tombol generate jadwal, maka akan ditampilkan modal generate jadwal pada halaman Admin_jadwal_pendampingan. Kemudian admin akan memasukkan tanggal mulai dan tanggal selesai dan menekan tombol generate. Tombol generate akan memanggil fungsi *generate_jadwal()* pada *controller* PenjadwalanController. *Method* *generate_jadwal()* akan memanggil *method* *seeInDatabase()* untuk melakukan proses cek apakah telah terdapat jadwal pendampingan pada tanggal



yang dipilih dengan menjalankan *method* *get()* pada model Jadwal_pendampingan. Model Jadwal_pendampingan akan mengembalikan data jadwal pendampingan ke PenjadwalanController apabila didapatkan data jadwal pendampingan pada tanggal yang dipilih. Jika data yang dikembalikan kosong maka menandakan bahwa tidak ada jadwal pendampingan pada tanggal yang dipilih. Proses dilanjutkan dengan memanggil *method* *get()* dari model Jadwal untuk mengambil data jadwal mahasiswa. Model Jadwal akan mengembalikan data jadwal mahasiswa ke PenjadwalanController dimana data tersebut akan diolah untuk mencari pendamping setiap jadwal perkuliahan mahasiswa disabilitas. Setelah proses pencarian pendamping selesai maka data akan disimpan ke *database* menggunakan *method* *insert()* pada model Jadwal_pendampingan dan model Jadwal_pendampingan akan mengembalikan status = 1 sebagai penanda untuk PenjadwalanController apabila proses simpan data ke *database* berhasil. PenjadwalanController akan mengembalikan informasi "Generate jadwal berhasil" ke *view* Admin_jadwal_pendampingan. Proses generate jadwal pendampingan dapat dilihat dalam Gambar 5.3.

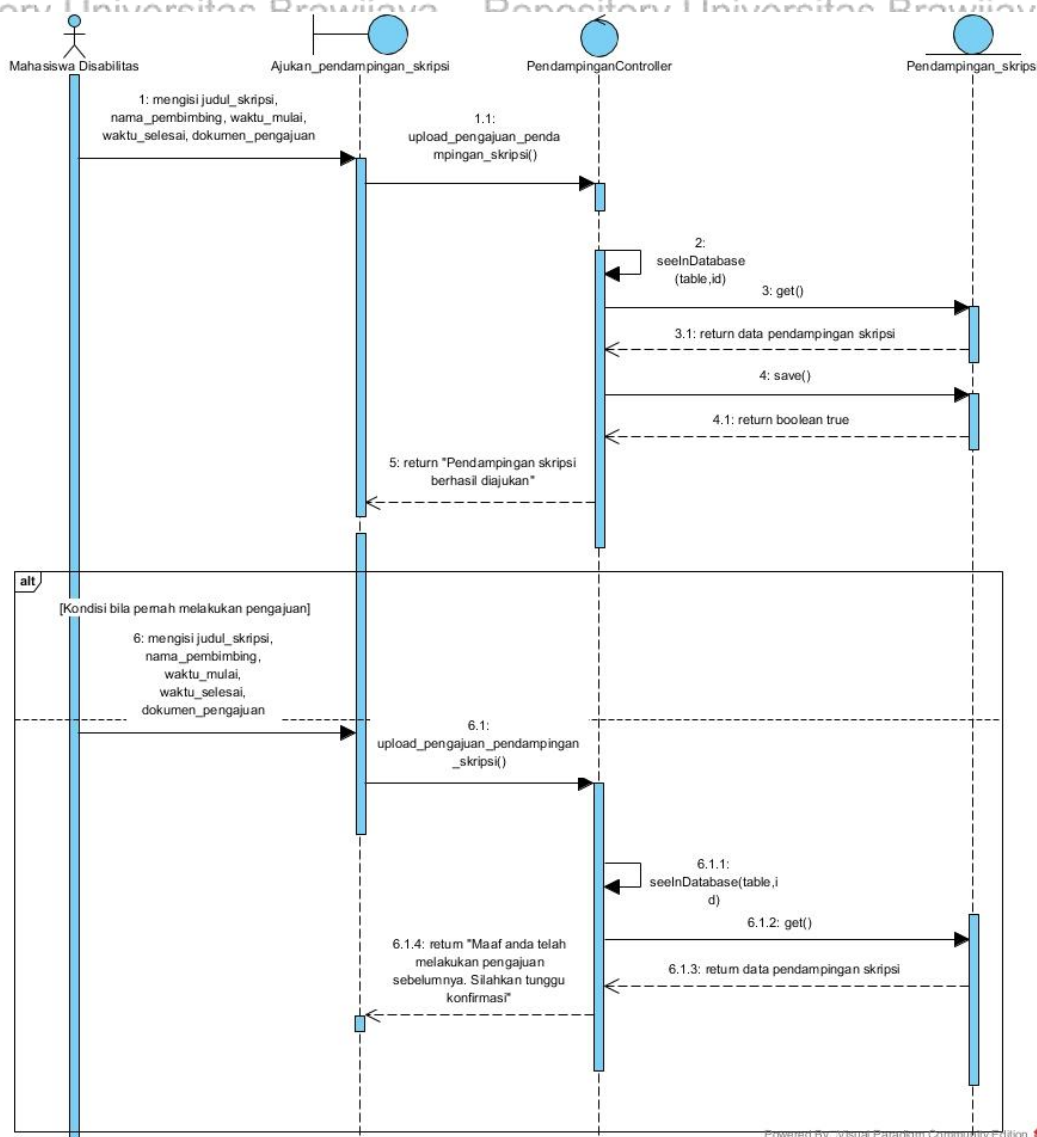


Gambar 5.2 Sequence Diagram Generate Jadwal Pendampingan

C. Sequence Diagram Pengajuan Pendampingan Skripsi

Pada fungsionalitas pengajuan pendampingan skripsi, fungsionalitas dijalankan oleh mahasiswa disabilitas. Proses diawali dengan memanggil *method* *seeInDatabase()* untuk melakukan cek apakah mahasiswa disabilitas sudah pernah melakukan pengajuan pendampingan skripsi sebelumnya. *Method* *seeInDatabase()* akan menjalankan *method* *get()* pada model *Pendampingan_skripsi* dimana akan mengembalikan data pendampingan skripsi. Apabila data yang dikembalikan kosong maka mahasiswa belum pernah melakukan pengajuan sebelumnya. Setelah itu *PendampinganController* akan menyimpan data pengajuan pendampingan yang telah dilakukan dengan memanggil *method* *save()* pada model *Pendampingan_skripsi*. Model *Pendampingan_skripsi* akan mengembalikan kembalian *boolean true* yang menandakan proses simpan data berhasil dan *PendampinganController* akan

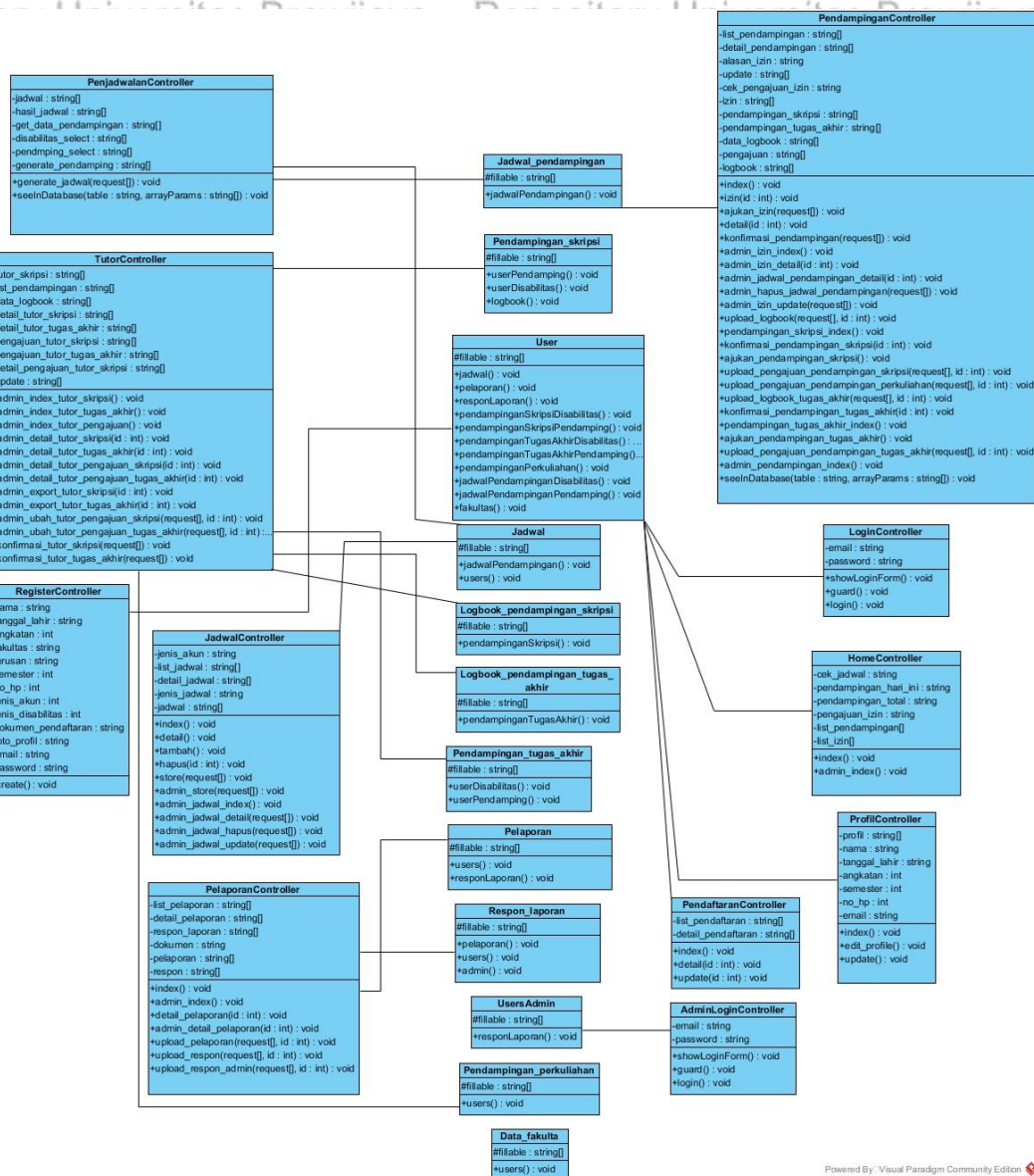
menampilkan informasi “Pendampingan skripsi berhasil diajukan”. Proses pengajuan pendampingan skripsi dapat dilihat dalam Gambar 5.4.



Gambar 5.3 Sequence Diagram Pengajuan Pendampingan Skripsi

5.1.1.2 Pemodelan Class Diagram

Class diagram merepresentasikan hubungan antar masing – masing kelas yang terdapat pada perangkat lunak pendampingan mahasiswa disabilitas. Dengan membuat *class diagram* akan membantu proses pengembangan pada tahap implementasi. Perangkat lunak pendampingan mahasiswa disabilitas ini menggunakan konsep pemrograman berbasis objek dalam proses pengembangannya. Pada *class diagram* perangkat lunak pendampingan mahasiswa disabilitas ini, terdapat tiga jenis kelas yaitu kelas *model*, *view* dan *controller*. Hubungan antar kelas dari perangkat lunak pendampingan mahasiswa disabilitas dapat dilihat pada *class diagram* yang direpresentasikan dalam Gambar 5.5.



Gambar 5.4 Class Diagram Perangkat Lunak Pendampingan Mahasiswa Disabilitas

A. Perancangan Class PendampinganController

PendampinganController merupakan *class* bertipe *controller*. *Class* ini memiliki fungsi utama dalam fitur pendampingan yang dilakukan oleh mahasiswa. Pada PendampinganController terdapat 12 atribut antara lain jenis_akun, alasan_izin, cek pengajuan izin bertipe data *string* dan atribut list_pendampingan, detail_pendampingan, update, izin, pendampingan_skripsi, pendampingan_tugas_akhir, data_logbook, pengajuan, logbook bertipe *array string*. Pada PendampinganController juga terdapat 23 *method* yaitu index(), izin(), ajukan_izin() detail(), konfirmasi_pendampingan(), admin_izin_index(), admin_izin_detail(), admin_jadwal_pendampingan(), admin_hapus_jadwal_pendampingan(), admin_izin_update(), upload_logbook(), pendampingan_skripsi_index(), konfirmasi_pendampingan_skripsi(),



ajukan_pendampingan_skripsi()), upload_pengajuan_pendampingan_skripsi()),
 upload_logbook_tugas_akhir(), konfirmasi_pendampingan_tugas_akhir()),
 pendampingan_tugas_akhir_index()), ajukan_pendampingan_tugas_akhir()),
 upload_pengajuan_pendampingan_tugas_akhir()),
 admin_pendampingan_index()), upload_pengajuan_pendampingan_perkuliahan
 dan seeliteDatabase(). PendampinganController memiliki relasi dengan modal
 Jadwal_pendampingan. Class PendampinganController dapat dijelaskan dalam
 Gambar 5.6.

PendampinganController
-jenis_akun : string -list_pendampingan : string[] -detail_pendampingan : string[] -alasan_izin : string -update : string[] -cek_pengajuan_izin : string -izin : string[] -pendampingan_skripsi : string[] -pendampingan_tugas_akhir : string[] -data_logbook : string[] -pengajuan : string[] -logbook : string[]
+index() : void +izin(id : int) : void +ajukan_izin(request[]) : void +detail(id : int) : void +konfirmasi_pendampingan(request[]) : void +admin_izin_index() : void +admin_izin_detail(id : int) : void +admin_jadwal_pendampingan_detail(id : int) : void +admin_hapus_jadwal_pendampingan(request[]) : void +admin_izin_update(request[]) : void +upload_logbook(request[], id : int) : void +pendampingan_skripsi_index() : void +konfirmasi_pendampingan_skripsi(id : int) : void +ajukan_pendampingan_skripsi() : void +upload_pengajuan_pendampingan_skripsi(request[], id : int) : void +upload_pengajuan_pendampingan_perkuliahan(request[], id : int) : void +upload_logbook_tugas_akhir(request[], id : int) : void +konfirmasi_pendampingan_tugas_akhir(id : int) : void +pendampingan_tugas_akhir_index() : void +ajukan_pendampingan_tugas_akhir() : void +upload_pengajuan_pendampingan_tugas_akhir(request[], id : int) : void +admin_pendampingan_index() : void +seeliteDatabase(table : string, arrayParams : string[]) : void

Gambar 5.5 Class PendampinganController

B. Perancangan Class PenjadwalanController

PenjadwalanController merupakan *class* bertipe *controller*. *Class* ini memiliki fungsi utama dalam fitur penjadwalan kegiatan pendampingan dari mahasiswa. Pada PenjadwalanController terdapat enam atribut antara lain jadwal, hasil_jadwal, get_data_pendampingan, disabilitas_select, pendamping_select, generate_jadwal bertipe *array string*. Kemudian terdapat pula sebuah *method* yaitu generate_jadwal() dan seeliteDatabase() dimana *method* tersebut bertipe *void*. *Class* PenjadwalanController memiliki relasi dengan beberapa *class* lain



model Jadwal_pendampingan dan jadwal. *Class* PenjadwalanController dapat dilihat dalam Gambar 5.7.

PenjadwalanController
-jadwal : string[] -hasil_jadwal : string[] -get_data_pendampingan : string[] -disabilitas_select : string[] -pendmping_select : string[] -generate_pendamping : string[] +generate_jadwal(request[]) : void +seeInDatabase(table : string, arrayParams : string[]) : void

Gambar 5.6 *Class* PenjadwalanController

C. Perancangan *Class* TutorController

TutorController merupakan *class* bertipe *controller*. *Class* ini memiliki fungsi utama dalam mengontrol kegiatan pendampingan skripsi dan pendampingan tugas akhir. Pada TutorController terdapat sembilan atribut antara lain tutor_skripsi, list_pendampingan, data_logbook, detail_tutor_skripsi, detail_tutor_tugas_akhir, pengajuan_tutor_skripsi, pengajuan_tutor_tugas_akhir, detail_pengajuan_tutor_skripsi dan update yang keseluruhan atribut bertipe *array string*. Pada TutorController juga terdapat 13 *method* antara lain admin_index_tutor_skripsi(), admin_index_tutor_tugas_akhir(), admin_index_tutor_pengajuan(), admin_detail_tutor_skripsi(), admin_detail_tutor_tugas_akhir(), admin_detail_tutor_pengajuan_skripsi(), admin_detail_tutor_pengajuan_tugas_akhir(), admin_export_tutor_skripsi(), admin_export_tutor_tugas_akhir(), admin_ubah_tutor_pengajuan_skripsi(), admin_ubah_tutor_pengajuan_tugas_akhir(), konfirmasi_tutor_skripsi(), konfirmasi_tutor_tugas_akhir(). *Class* TutorController juga memiliki relasi dengan *class* lain *model* Pendampingan_skripsi, pendampingan_tugas_akhir, logbook_pendampingan_skripsi, logbook_pendampingan_tugas_akhir. *Class* TutorController dapat dilihat dalam Gambar 5.8.



TutorController
-tutor_skripsi : string[] -list_pendampingan : string[] -data_logbook : string[] -detail_tutor_skripsi : string[] -detail_tutor_tugas_akhir : string[] -pengajuan_tutor_skripsi : string[] -pengajuan_tutor_tugas_akhir : string[] -detail_pengajuan_tutor_skripsi : string[] -update : string[]
+admin_index_tutor_skripsi() : void +admin_index_tutor_tugas_akhir() : void +admin_index_tutor_pengajuan() : void +admin_detail_tutor_skripsi(id : int) : void +admin_detail_tutor_tugas_akhir(id : int) : void +admin_detail_tutor_pengajuan_skripsi(id : int) : void +admin_detail_tutor_pengajuan_tugas_akhir(id : int) : void +admin_export_tutor_skripsi(id : int) : void +admin_export_tutor_tugas_akhir(id : int) : void +admin_ubah_tutor_pengajuan_skripsi(request[], id : int) : void +admin_ubah_tutor_pengajuan_tugas_akhir(request[], id : int) : void +konfirmasi_tutor_skripsi(request[]) : void +konfirmasi_tutor_tugas_akhir(request[]) : void

Gambar 5.7 Class TutorController

D. Perancangan Class LoginController

LoginController merupakan class bertipe *controller*. Class ini memiliki fungsi dalam mengatur *login* pengguna. Pada LoginController terdapat dua atribut yaitu *email* dan *password* yang bertipe *string*. Kemudian terdapat pula method *showLoginForm()*, *guard()* dan *login()* bertipe *void*. LoginController memiliki relasi dengan class lainnya yaitu Users. LoginController dapat dilihat dalam Gambar 5.9.

LoginController
-email : string -password : string
+showLoginForm() : void +guard() : void +login() : void

Gambar 5.8 Class LoginController

E. Perancangan Class RegisterController

RegisterController merupakan class bertipe *controller*. Class ini memiliki fungsi utama dalam mengatur proses registrasi yang dilakukan mahasiswa disabilitas dan *volunteer*. Pada RegisterController terdapat 13 atribut antara lain nama, tanggal, fakultas, jurusan, dokumen pendaftaran, email dan password yang masing – masing bertipe *string*. Kemudian terdapat juga atribut angkatan, semester, no_hp, jenis_akun dan jenis_disabilitas yang masing – masing bertipe *integer*. Pada RegisterController terdapat satu method yaitu method *create()*



bertipe `void`. `RegisterController` memiliki relasi dengan `class` lain yaitu `register` dan `Users`. `Class RegisterController` dapat dilihat dalam Gambar 5.10.

RegisterController
-nama : string
-tanggal_lahir : string
-angkatan : int
-fakultas : string
-jurusan : string
-semester : int
-no_hp : int
-jenis_akun : int
-jenis_disabilitas : int
-dokumen_pendaftaran : string
-foto_profil : string
-email : string
-password : string
+create() : void

Gambar 5.9 Class RegisterController

F. Perancangan Class HomeController

`HomeController` merupakan `class` bertipe `controller`. `Class` ini memiliki fungsi utama untuk mengatur halaman awal dari perangkat lunak pendampingan mahasiswa disabilitas. Pada `HomeController` terdapat tujuh atribut yaitu `jenis_akun`, `cek_jadwal`, `pendampingan_hari_ini`, `pendampingan_total`, `pengajuan_izin` bertipe `string` serta `list_pendampingan`, `list_izin` bertipe `array`. Pada `HomeController` ini terdapat juga dua `method` yaitu `index()` dan `admin_index` bertipe `void`. `HomeController` memiliki relasi dengan `class model` `users`. `Class HomeController` dapat dilihat dalam Gambar 5.11.

HomeController
-jenis_akun : string
-cek_jadwal : string
-pendampingan_hari_ini : string
-pendampingan_total : string
-pengajuan_izin : string
-list_pendampingan[]
-list_izin[]
+index() : void
+admin_index() : void

Gambar 5.10 Class HomeController

G. Perancangan Class JadwalController

`JadwalController` merupakan `class` bertipe `controller`. `Class` ini memiliki fungsi untuk mengatur fungsi jadwal dari kegiatan mahasiswa. Pada `JadwalController` terdapat lima atribut antara lain `jenis_akun` dan `jenis_jadwal` dimana masing-masing atribut bertipe `string` serta `list_jadwal`, `detail_jadwal`, `jadwal` bertipe



string array. Pada *JadwalController* terdapat sepuluh *method* yaitu *index()*, *detail()*, *tambah()*, *hapus()*, *store()*, *admin_store()*, *admin_jadwal_index()*, *admin_jadwal_detail()*, *admin_jadwal_hapus()* dan *admin_jadwal_update()* dimana masing – masing *method* bertipe *void*. *Class JadwalController* memiliki relasi dengan *class model* *Jadwal*. *Class JadwalController* dapat dilihat dalam Gambar 5.12.

JadwalController
-jenis_akun : string -list_jadwal : string[] -detail_jadwal : string[] -jenis_jadwal : string -jadwal : string[]
+index() : void +detail() : void +tambah() : void +hapus(id : int) : void +store(request[]) : void +admin_store(request[]) : void +admin_jadwal_index() : void +admin_jadwal_detail(request[]) : void +admin_jadwal_hapus(request[]) : void +admin_jadwal_update(request[]) : void

Gambar 5.11 Class JadwalController

H. Perancangan Class ProfilController

ProfilController merupakan *class* bertipe *controller*. *Class* ini memiliki fungsi untuk mengatur profil dari pengguna. Pada *ProfilController* terdapat delapan atribut. Terdapat atribut bertipe *string* yaitu *jenis_akun*, *nama*, *tanggal_lahir* dan *email*. Pada *ProfilController* juga terdapat atribut bertipe *integer* yaitu *angkatan*, *semester* dan *no_hp*. Pada *ProfilController* terdapat juga atribut bertipe *string* dengan *modifier array* yaitu *profil*. Selain atribut, terdapat juga tiga *method* yaitu *method* *index()*, *edit_profile()* dan *update()* yang masing – masing bertipe *void*. *Class ProfilController* memiliki relasi dengan *class* *profil*, *edit_profil* dan *Users*. *Class ProfilController* dapat dilihat dalam Gambar 5.13.



ProfilController
-jenis_akun : string
-profil : string[]
-nama : string
-tanggal_lahir : string
-angkatan : int
-semester : int
-no_hp : int
-email : string
+index() : void
+edit_profile() : void
+update() : void

Gambar 5.12 Class ProfilController

I. Perancangan Class AdminLoginController

AdminLoginController merupakan class bertipe *controller*. Class ini memiliki fungsi dalam mengatur *login* admin. Pada AdminLoginController terdapat dua atribut yaitu *email* dan *password* yang bertipe *string*. Kemudian terdapat pula method *showLoginForm()*, *guard()* dan *login()* bertipe *void*. AdminLoginController memiliki relasi dengan class lainnya yaitu *UsersAdmin*. AdminLoginController dapat dilihat dalam Gambar 5.14.

AdminLoginController
-email : string
-password : string
+showLoginForm() : void
+guard() : void
+login() : void

Gambar 5.13 Class AdminLoginController

J. Perancangan Class PendaftaranController

PendaftaranController merupakan class bertipe *controller*. Class ini memiliki fungsi dalam mengatur pengelolaan pendaftaran oleh admin. Pada PendaftaranController terdapat dua atribut yaitu *list_pendaftaran* dan *detail_pendaftaran* yang bertipe *string*. Kemudian terdapat pula method *index()*, *detail()* dan *update()* bertipe *void*. PendaftaranController memiliki relasi dengan class lainnya yaitu *Users*. PendaftaranController dapat dilihat dalam Gambar 5.15.



PendaftaranController
-list_pendaftaran : string[]
-detail_pendaftaran : string[]
+index() : void
+detail(id : int) : void
+update(id : int) : void

Gambar 5.14 Class PendaftaranController

K. Perancangan Class PelaporanController

PelaporanController merupakan class bertipe *controller*. Class ini memiliki fungsi dalam mengatur proses pelaporan yang dilakukan oleh mahasiswa dan admin. Pada PelaporanController terdapat enam atribut *list_pelaporan*, *detail_pelaporan*, *respon_laporan*, *pelaporan* dan *respon* yang bertipe *string array* dan terdapat pula dokumen bertipe *string*. Kemudian terdapat method *index()*, *admin_index()*, *detail_pelaporan()*, *admin_detail_pelaporan()*, *upload_pelaporan()*, *upload_respon()* dan *upload_respon_admin()* bertipe *void*. PelaporanController memiliki relasi dengan class lainnya yaitu *model* Pelaporan dan Respon_laporan. PelaporanController dapat dilihat dalam Gambar 5.16.

PelaporanController
-list_pelaporan : string[]
-detail_pelaporan : string[]
-respon_laporan : string[]
-dokumen : string
-pelaporan : string[]
-respon : string[]
+index() : void
+admin_index() : void
+detail_pelaporan(id : int) : void
+admin_detail_pelaporan(id : int) : void
+upload_pelaporan(request[], id : int) : void
+upload_respon(request[], id : int) : void
+upload_respon_admin(request[], id : int) : void

Gambar 5.15 Class PelaporanController

L. Perancangan Class Jadwal_pendampingan

Jadwal_pendampingan merupakan class bertipe *model*. Class ini memiliki fungsi dalam akses data jadwal pendampingan pada *database*. Pada Jadwal_pendampingan terdapat satu atribut yaitu *fillable* bertipe *string*. Kemudian terdapat pula *jadwalPendampingan()* bertipe *void*. Jadwal_pendampingan memiliki relasi dengan class lainnya yaitu *PenjadwalanController* dan *PendampinganController*. Jadwal_pendampingan dapat dilihat dalam Gambar 5.17.



Jadwal_pendampingan
#fillable : string[]
+jadwalPendampingan() : void

Gambar 5.16 Class Jadwal_pendampingan

L. Perancangan Class Pendampingan_skripsi

Pendampingan_skripsi merupakan class bertipe *model*. Class ini memiliki fungsi dalam akses data pendampingan skripsi pada *database*. Pada Pendampingan_skripsi terdapat satu atribut yaitu *fillable* bertipe *string*. Kemudian terdapat pula *method* *userPendamping()*, *userDisabilitas()* dan *logbook()* bertipe *void*. Pendampingan_skripsi memiliki relasi dengan class lainnya yaitu *TutorController*. Pendampingan_skripsi dapat dilihat dalam Gambar 5.18.

Pendampingan_skripsi
#fillable : string[]
+userPendamping() : void
+userDisabilitas() : void
+logbook() : void

Gambar 5.17 Class Pendampingan_skripsi

M. Perancangan Class User

User merupakan class bertipe *model*. Class ini memiliki fungsi dalam akses data mahasiswa pada *database*. Pada User terdapat satu atribut yaitu *fillable* bertipe *string*. Kemudian terdapat pula *method* *jadwal()*, *pelaporan()*, *responLaporan()*, *pendampinganSkripsiDisabilitas()*, *pendampinganSkripsiPendamping()*, *pendampinganTugasAkhirDisabilitas()*, *pendampinganTugasAkhirPendamping()*, *pendampinganPerkuliahan()*, *jadwalPendampinganDisabilitas()*, *jadwalPendampinganPendamping()* dan *fakultas()* bertipe *void*. User memiliki relasi dengan class lainnya yaitu *TutorController*, *LoginController*, *HomeController*, *ProfilController* dan *PendaftaranController*. User dapat dilihat dalam Gambar 5.19.



User
#fillable : string[]
+jadwal() : void
+pelaporan() : void
+responLaporan() : void
+pendampinganSkripsiDisabilitas() : void
+pendampinganSkripsiPendamping() : void
+pendampinganTugasAkhirDisabilitas() : ...
+pendampinganTugasAkhirPendamping()...
+pendampinganPerkuliahan() : void
+jadwalPendampinganDisabilitas() : void
+jadwalPendampinganPendamping() : void
+fakultas() : void

Gambar 5.18 Class User

N. Perancangan Class Jadwal

Jadwal merupakan class bertipe *model*. Jadwal ini memiliki fungsi dalam akses data jadwal mahasiswa pada *database*. Pada Jadwal terdapat satu atribut yaitu *fillable* bertipe *string*. Kemudian terdapat pula *method* *jadwalPendampingan()* dan *users()* bertipe *void*. Jadwal memiliki relasi dengan class lainnya yaitu *JadwalController* dan *PenjadwalanController*. Jadwal dapat dilihat dalam Gambar 5.20.

Jadwal
#fillable : string[]
+jadwalPendampingan() : void
+users() : void

Gambar 5.19 Class Jadwal

O. Perancangan Class Logbook_pendampingan_skripsi

Logbook_pendampingan_skripsi merupakan class bertipe *model*. *Logbook_pendampingan_skripsi* ini memiliki fungsi dalam akses data *logbook* pendampingan skripsi mahasiswa pada *database*. Pada *Logbook_pendampingan_skripsi* terdapat satu atribut yaitu *fillable* bertipe *string*. Kemudian terdapat pula *method* *pendampinganSkripsi()* bertipe *void*. *Logbook_pendampingan_skripsi* memiliki relasi dengan class lainnya yaitu *TutorController*. *Logbook_pendampingan_skripsi* dapat dilihat dalam Gambar 5.21.



Logbook_pendampingan_skripsi
#fillable : string[]
+pendampinganSkripsi() : void

Gambar 5.20 Class Logbook_pendampingan_skripsi

P. Perancangan Class Logbook_pendampingan_tugas_akhir

Logbook_pendampingan_tugas_akhir merupakan class bertipe *model*. Logbook_pendampingan_tugas_akhir ini memiliki fungsi dalam akses data logbook pendampingan tugas akhir mahasiswa pada *database*. Pada Logbook_pendampingan_tugas_akhir terdapat satu atribut yaitu fillable bertipe *string*. Kemudian terdapat pula *method* pendampinganTugasAkhir() bertipe *void*. Logbook_pendampingan_tugas_akhir memiliki relasi dengan class lainnya yaitu TutorController. Logbook_pendampingan_tugas_akhir dapat dilihat dalam Gambar 5.22.

Logbook_pendampingan_tugas_akhir
#fillable : string[]
+pendampinganTugasAkhir() : void

Gambar 5.21 Class Logbook_pendampingan_tugas_akhir

Q. Perancangan Class Pendampingan_tugas_akhir

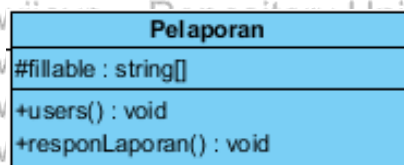
Pendampingan_tugas_akhir merupakan class bertipe *model*. Pendampingan_tugas_akhir ini memiliki fungsi dalam akses data pendampingan tugas akhir mahasiswa pada *database*. Pada Pendampingan_tugas_akhir terdapat satu atribut yaitu fillable bertipe *string*. Kemudian terdapat pula *method* userDisabilitas() dan userPendamping() bertipe *void*. Pendampingan_tugas_akhir memiliki relasi dengan class lainnya yaitu TutorController. Pendampingan_tugas_akhir dapat dilihat dalam Gambar 5.23.

Pendampingan_tugas_akhir
#fillable : string[]
+userDisabilitas() : void
+userPendamping() : void

Gambar 5.22 Class Pendampingan_tugas_akhir

R. Perancangan Class Pelaporan

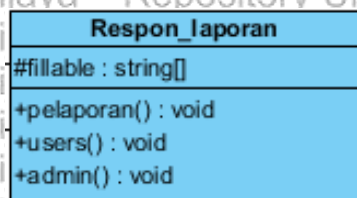
Pelaporan merupakan class bertipe *model*. Pelaporan ini memiliki fungsi dalam akses data pelaporan yang dilakukan mahasiswa pada *database*. Pada Pelaporan terdapat satu atribut yaitu fillable bertipe *string*. Kemudian terdapat pula *method* users() dan responLaporan() bertipe *void*. Laporan memiliki relasi dengan class lainnya yaitu PelaporanController. Pelaporan dapat dilihat dalam Gambar 5.24.



Gambar 5.23 Class Pelaporan

S. Perancangan Class Respon_laporan

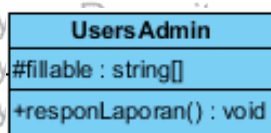
Respon_laporan merupakan class bertipe *model*. Respon_laporan ini memiliki fungsi dalam akses data respon pelaporan yang dilakukan mahasiswa dan admin pada *database*. Pada Respon_laporan terdapat satu atribut yaitu *fillable* bertipe *string*. Kemudian terdapat pula *method* *pelaporan()*, *users()* dan *admin()* bertipe *void*. Respon_laporan memiliki relasi dengan *class* lainnya yaitu *PelaporanController*. Respon_laporan dapat dilihat dalam Gambar 5.25.



Gambar 5.24 Class Respon_laporan

T. Perancangan Class UsersAdmin

UsersAdmin merupakan class bertipe *model*. UsersAdmin ini memiliki fungsi dalam akses data admin pada *database*. Pada UsersAdmin terdapat satu atribut yaitu *fillable* bertipe *string*. Kemudian terdapat pula *method* *responLaporan()* bertipe *void*. UsersAdmin memiliki relasi dengan *class* lainnya yaitu *AdminLoginController*. UsersAdmin dapat dilihat dalam Gambar 5.26.



Gambar 5.25 Class UsersAdmin

U. Perancangan Class Pendampingan_perkuliahan

Pendampingan_perkuliahan merupakan class bertipe *model*. Pendampingan_perkuliahan ini memiliki fungsi dalam akses data pengajuan pendampingan perkuliahan pada *database*. Pada Pendampingan_perkuliahan terdapat satu atribut yaitu *fillable* bertipe *string*. Kemudian terdapat pula *method* *users()* bertipe *void*. Pendampingan_perkuliahan memiliki relasi dengan *class* lainnya yaitu *TutorController*. Pendampingan_perkuliahan dapat dilihat dalam Gambar 5.27.



Pendampingan_perkuliahan
#fillable : string[]
+users() : void

Gambar 5.26 Class Pendampingan_perkuliahan

V. Perancangan Class Data_fakulta

Data_fakulta merupakan class bertipe *model*. Data_fakulta ini memiliki fungsi dalam akses data fakultas pada *database*. Pada Data_fakulta terdapat satu atribut yaitu *fillable* bertipe *string*. Kemudian terdapat pula *method* *users()* bertipe *void*. Data_fakulta dapat dilihat dalam Gambar 5.28.

Data_fakulta
#fillable : string[]
+users() : void

Gambar 5.27 Class Data_fakulta

5.1.1.3 Perancangan Algoritme

Perancangan algoritme akan digunakan dalam pengembangan perangkat lunak pendampingan mahasiswa disabilitas. Dalam melakukan perancangan algoritme ini akan digunakan *pseudocode*. *Pseudocode* penting untuk digunakan karena mampu mendeskripsikan algoritma secara lebih natural sehingga mampu dipahami secara baik oleh manusia. *Pseudocode* akan memberikan penjelasan algoritma secara runtut dengan menggunakan bahasa yang lebih baku walaupun menyerupai bahasa pemrograman.

1. Pseudocode Pengajuan Pendampingan Skripsi

Pada Tabel 5.1 terdapat *pseudocode* dari pengajuan pendampingan skripsi. Proses pengajuan skripsi dimulai dengan melakukan deklarasi variabel *judul_skripsi*, *nama_pembimbing*, *waktu_mulai*, *waktu_selesai* dan *dokumen_pengajuan*. Kemudian dilanjutkan dengan proses cek apakah mahasiswa pernah melakukan pengajuan pendampingan skripsi dengan memanggil *method* *seeInDatabase()* Kemudian dilanjutkan dengan proses seleksi hasil dari pemanggilan *method* *seeInDatabase()*. Jika hasil menunjukkan terdapat data pengajuan maka akan menampilkan pesan error, jika tidak maka akan melakukan proses simpan data pengajuan pendampingan skripsi. Alur pengajuan pendampingan skripsi dapat dilihat pada Tabel 5.1.

Tabel 5.1 Pseudocode Pengajuan Pendampingan Skripsi

1	Begin
2	Declaration <i>judul_skripsi</i> , <i>nama_pembimbing</i> ,
3	<i>waktu_mulai</i> , <i>waktu_selesai</i> , <i>dokumen_pengajuan</i>
4	Instance pengajuan from class <i>Pendampingan_skripsi</i>
5	Call method <i>seeInDatabase()</i>
6	If (output of <i>seeInDatabase()</i> == true)



```

7      Print "Maaf anda telah melakukan pengajuan
      sebelumnya. Silahkan tunggu konfirmasi"
8      Else
9          If(input is null)
10         Print "Isi semua data terlebih dahulu"
11     Else
12         Do save data pengajuan pendampingan skripsi to
          Database.
13         Print "Pendampingan skripsi berhasil diajukan"
14     Endif
16 Endif
16 End

```

2. *Pseudocode Generate Jadwal Pendampingan*

Pada Tabel 5.2 terdapat *pseudocode* dari generate jadwal pendampingan. Proses generate jadwal pendampingan dimulai dengan melakukan deklarasi variabel `tanggal_mulai` dan `tanggal_selesai`. Kemudian akan dilakukan pemanggilan *method* `seeInDatabase()` untuk melakukan cek apakah terdapat jadwal pendampingan pada tanggal yang dipilih. Apabila hasil dari *method* `seeInDatabase()` menunjukkan terdapat jadwal pendampingan pada tanggal pilihan, maka akan mengembalikan pesan error, jika hasil dari *method* `seeInDatabase()` menandakan belum ada jadwal pendampingan pada tanggal pilihan maka proses dilakukan dengan memanggil *method* `do_generate()` untuk melakukan proses pencocokan antara jadwal mahasiswa disabilitas dan mahasiswa *voulnteer*. Hasil dari *method* `do_generate()` selanjutnya disimpan ke *database* dan mengembalikan pesan jika generate jadwal berhasil. Proses generate jadwal pendampingan dapat dilihat pada Tabel 5.2.

Tabel 5.2 *Pseudocode* Generate Jadwal Pendampingan

```

1      Begin
2      Declaration tanggal_mulai, tanggal_selesai
3      Call method seeInDatabase()
4      If(output of seeInDatabase() == true)
5          Print "telah terdapat jadwal pendampingan pada
          tanggal yang dipilih."
6      Else
7          Call method do_generate()
8          Inisialization data from method do_generate() to
          variable hasil_jadwal
9          Do save hasil_jadwal to Database.
10         Print "Generate jadwal berhasil"
11     Endif
12 End

```

3. *Pseudocode Mulai Pendampingan*

Pada Tabel 5.3 terdapat *pseudocode* dari mulai pendampingan. Proses mulai pendampingan diawali dengan melakukan proses cek apakah terdapat `id_pendampingan` atau tidak. Kemudian dilanjutkan dengan deklarasi dan inisialisasi beberapa variabel yaitu `id_pendampingan`, `berhasil`, `gagal` dan



id_pendamping. Kemudian dilakukan proses cek menggunakan *method* *seeInDatabase()* untuk mengetahui apakah data pendampingan sesuai atau tidak. Jika sesuai maka akan dilanjutkan dengan *update* status pendampingan menjadi dikonfirmasi dan mengembalikan pesan berhasil. Jika gagal maka akan dilakukan proses mengembalikan pesan gagal. Proses mulai pendampingan dapat dilihat pada Tabel 5.3.

Tabel 5.3 Pseudocode Mulai Pendampingan

1	Begin
2	If (POST[id_pendampingan] is exist)
3	Inisialization id_pendampingan, status, gagal, berhasil, id_pendamping
4	Call method seeInDatabase()
5	Inisialization cek_data with output from seeInDatabase()
6	If (cek data is true)
7	Update status jadwal_pendampingan where id = id_pendampingan and id_pendamping = id_pendamping
8	Print 1
9	Else
10	Print 0
11	Endif
12	Else
13	Print 0
14	Endif
15	End

5.1.2 Perancangan Antarmuka

Perancangan antarmuka akan digunakan dalam pengembangan perangkat lunak pendampingan mahasiswa disabilitas. Perancangan antarmuka sangat penting untuk dilakukan sebagai dasar dalam membangun tampilan dari perangkat lunak pendampingan mahasiswa disabilitas ini. Dengan membuat perancangan antarmuka, akan diketahui bagaimana hasil akhir dari tampilan yang harus dikembangkan sehingga dapat memudahkan proses pengembangan tampilan perangkat lunak. Selain itu penting dilakukannya perancangan antarmuka mengingat tampilan adalah komponen utama dari perangkat lunak yang terlihat pertama kali oleh pengguna dan menjadi media dalam penyampaian informasi kepada pengguna dari perangkat lunak pendampingan mahasiswa disabilitas ini. Perancangan antarmuka akan dijelaskan dalam subbab 5.1.5.1 sampai 5.1.5.3.

5.1.2.1 Perancangan Antarmuka Pengajuan Pendampingan Skripsi

Gambar 5.28 Perancangan Antarmuka Pengajuan Pendampingan Skripsi

Perancangan antarmuka pengajuan skripsi dapat dilihat dalam Gambar 5.15. Antarmuka dari pengajuan skripsi ini akan ditampilkan pada pengguna dengan status mahasiswa disabilitas. Terdapat delapan komponen utama dalam antarmuka pengajuan skripsi antara lain tombol ajukan, masukan judul skripsi, masukan nama pembimbing, pilihan waktu mulai, pilihan waktu selesai, masukan *upload* dokumen, tombol pilih dokumen dan tombol ajukan pendampingan. Penjelasan detail mengenai delapan komponen dari pengajuan pendampingan skripsi ini dapat dilihat pada Tabel 5.4.

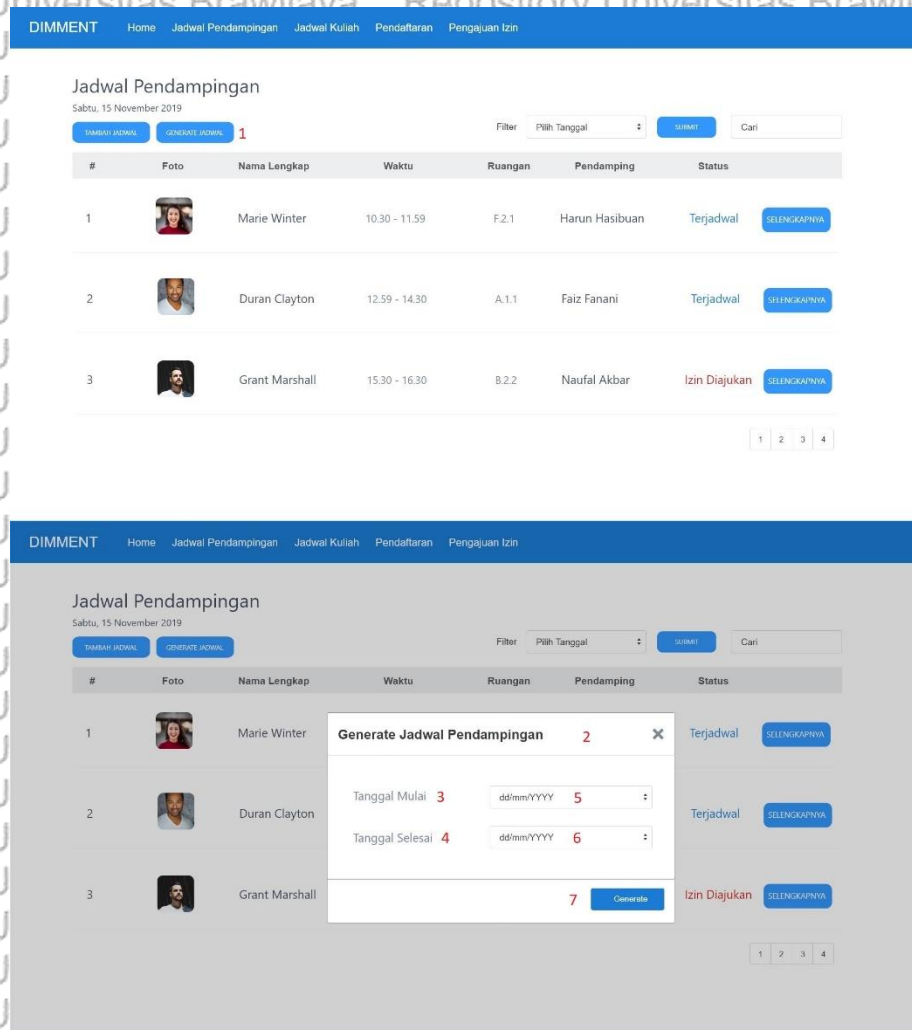
Tabel 5.4 Komponen Perancangan Antarmuka Pengajuan Pendampingan Skripsi

No.	Nama Komponen	Penjelasan
1	Tombol Ajukan	Tombol ajukan berfungsi untuk membuka halaman pengisian pengajuan pendampingan skripsi. Tombol ini hanya muncul jika mahasiswa disabilitas belum pernah melakukan pengajuan pendampingan skripsi.
2	Masukan Judul Skripsi	Masukan judul skripsi berfungsi sebagai tempat untuk mengisi judul skripsi yang akan diajukan untuk didampingi.
3	Masukan Nama Pembimbing	Masukan nama pembimbing berfungsi sebagai tempat untuk mengisi nama pembimbing yang membimbing mahasiswa disabilitas.
4	Masukan Waktu	Masukan waktu mulai berfungsi sebagai tempat untuk mengisi waktu mulai pendampingan yang



No.	Nama Komponen	Penjelasan
	Mulai.	diajukan.
5	Masukan Waktu Selesai.	Masukan waktu selesai berfungsi sebagai tempat untuk mengisi waktu selesai pendampingan yang diajukan.
6	Masukan Dokumen Upload.	Masukan <i>upload</i> dokumen berfungsi sebagai tempat untuk menampilkan nama dokumen yang akan di unggah.
7	Tombol Pilih.	Tombol pilih berfungsi untuk memilih dokumen yang akan di unggah untuk pengajuan pendampingan skripsi.
8	Tombol Ajukan Pendampingan.	Tombol ajukan pendampingan berfungsi untuk mengajukan pendampingan skripsi.

5.1.2.2 Perancangan Antarmuka Generate Jadwal Pendampingan



Gambar 5.29 Perancangan Antarmuka Generate Jadwal Pendampingan

Perancangan antarmuka generate jadwal pendampingan dapat dilihat dalam Gambar 5.16. Antarmuka dari generate jadwal pendampingan akan ditampilkan kepada aktor dengan status sebagai *admin*. Pada perancangan antarmuka generate jadwal pendampingan, terdapat sebelas komponen utama yang digunakan untuk menjalankan fungsionalitas generate jadwal pendampingan antara lain tombol generate jadwal, generate dan simpan generate, sebuah *modal pop up* yang akan menampilkan input tanggal mulai dan tanggal selesai dan sebuah tabel untuk menampilkan hasil generate jadwal yang dilakukan pada batas tanggal yang dipilih. Penjelasan detail mengenai perancangan antarmuka generate jadwal pendampingan dapat dilihat pada tabel 5.5.

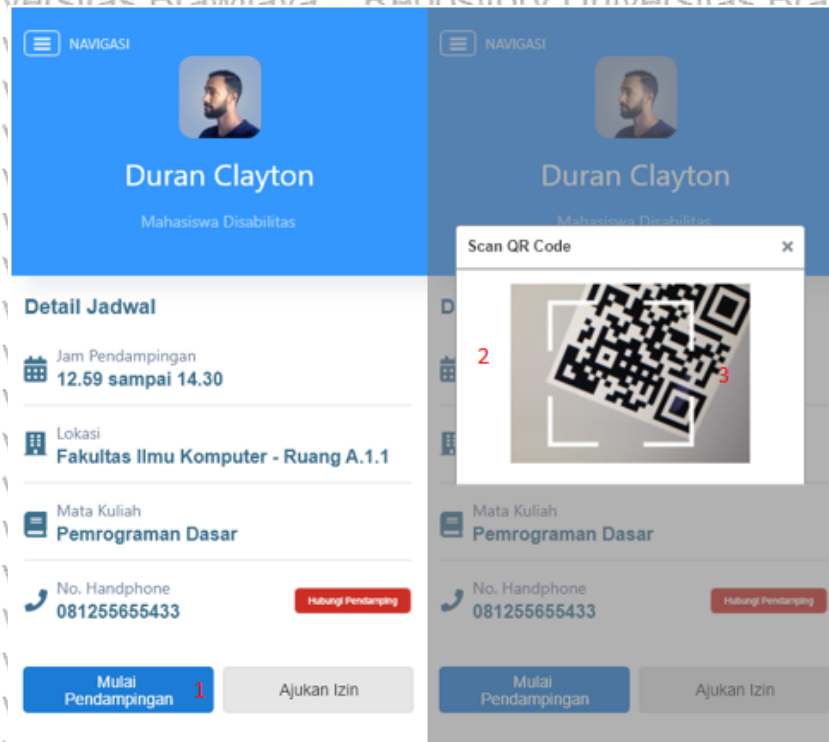
Tabel 5.5 Komponen Perancangan Antarmuka Generate Jadwal Pendampingan

No.	Nama Komponen	Penjelasan
1	Tombol Generate Jadwal.	Tombol generate jadwal berfungsi untuk memulai proses generate jadwal pendampingan



No.	Nama Komponen	Penjelasan
1		dimana aksi yang dijalankan ketika menekan tombol ini adalah membuka <i>modal pop up</i> yang berisi masukan tanggal mulai dan tanggal selesai.
2	Modal Generate Jadwal Pendampingan.	Modal generate jadwal pendampingan adalah modal yang akan muncul apabila tombol generate jadwal pendampingan di tekan. Modal ini berisi masukan tanggal mulai, tanggal selesai, label tanggal mulai, label tanggal selesai dan tombol generate.
3	Label Tanggal Mulai.	Label tanggal mulai merupakan teks yang menampilkan informasi pendukung untuk data masukan pada masukan tanggal mulai.
4	Label Tanggal Selesai.	Label tanggal selesai merupakan teks yang menampilkan informasi pendukung untuk data masukan pada masukan tanggal selesai.
5	Masukan Tanggal Mulai.	Masukan tanggal mulai berfungsi sebagai tempat untuk mengisi tanggal mulai dari jadwal pendampingan yang akan di <i>generate</i> .
6	Masukan Tanggal Selesai.	Masukan tanggal selesai berfungsi sebagai tempat untuk mengisi tanggal selesai dari jadwal pendampingan yang akan di <i>generate</i> .
7	Tombol Generate.	Tombol generate berfungsi sebagai tombol yang akan menjalankan fungsi generate jadwal sesuai dengan masukan tanggal mulai dan tanggal selesai.

5.1.2.3 Perancangan Antarmuka Mulai Pendampingan



Gambar 5.30 Perancangan Antarmuka Mulai Pendampingan

Perancangan antarmuka mulai pendampingan dapat dilihat dalam Gambar 5.17. Antarmuka dari mulai pendampingan ini akan ditampilkan pada pengguna dengan status sebagai mahasiswa *volunteer*. Terdapat 3 komponen utama yang digunakan untuk memulai pendampingan ini. Ketiga komponen tersebut adalah tombol mulai pendampingan, modal *scan QR code* dan video *preview scan QR code*. Penjelasan detail mengenai ketiga komponen ini dapat dilihat pada tabel 5.6.

Tabel 5.6 Komponen Perancangan Antarmuka Mulai Pendampingan

No.	Nama Komponen	Penjelasan
1	Tombol Mulai Pendampingan.	Tombol mulai pendampingan berfungsi untuk membuka modal <i>scan QR code</i> .
2	Modal <i>scan QR code</i> .	Modal <i>scan QR code</i> adalah modal yang akan muncul apabila tombol mulai pendampingan ditekan. Di dalam modal ini akan terdapat video <i>preview scan QR code</i> yang akan menampilkan proses <i>scan</i> yang dilakukan oleh kamera perangkat yang digunakan oleh mahasiswa <i>volunteer</i> .
3	Video <i>preview scan QR code</i> .	Video <i>preview scan QR code</i> berfungsi untuk menampilkan proses <i>scan QR code</i> yang dilakukan oleh kamera perangkat dari



No.	Nama Komponen	Penjelasan
		mahasiswa <i>volunteer</i> secara <i>real time</i> sehingga dapat membantu mahasiswa <i>volunteer</i> untuk menyelaraskan posisi kamera dengan <i>QR code</i> .

5.2 Implementasi Sistem

Pada bab implementasi, akan dilakukan proses pembuatan dari perangkat lunak pendampingan mahasiswa disabilitas menjadi perangkat lunak yang dapat digunakan oleh pengguna. Proses implementasi ini dilakukan berdasarkan rancangan perangkat lunak yang telah dibuat sebelumnya pada tahap perancangan. Implementasi dari perangkat lunak pendampingan mahasiswa disabilitas dilakukan menggunakan pendekatan berbasis objek (*Object Oriented Programming*).

5.2.1 Spesifikasi Sistem

Pada subbab ini akan dijelaskan bagaimana spesifikasi sistem yang digunakan dalam proses implementasi dari perangkat lunak pendampingan mahasiswa disabilitas. Spesifikasi sistem akan dijabarkan menjadi spesifikasi perangkat lunak yang digunakan dan spesifikasi perangkat keras yang digunakan selama proses implementasi perangkat lunak pendampingan mahasiswa disabilitas.

5.2.1.1 Spesifikasi Perangkat Lunak

Pada implementasi perangkat lunak pendampingan mahasiswa disabilitas, terdapat beberapa perangkat lunak yang digunakan untuk membantu proses implementasi yang dilakukan. Perangkat lunak yang digunakan terbagi menjadi sistem operasi, editor pemrograman, editor perancangan, *framework*, *server* lokal, *database* dan *browser*. Untuk penjelasan lebih lengkap mengenai spesifikasi perangkat lunak yang digunakan dalam tahap implementasi dapat dilihat pada Tabel 5.7.

Tabel 5.7 Spesifikasi Perangkat Lunak

No	Nama Komponen	Spesifikasi
1	Sistem Operasi	Windows 10 Pro Versi 1809 64 Bit
2	Editor Perancangan	Visual Paradigm Versi 15.1
3	Editor Pemrograman	Visual Studio Code Versi 1.40.2
4	Bahasa Pemrograman dan <i>Framework</i>	• Laravel • Bootstrap • Datatable • Fontawesome • PHP • HTML



No	Nama Komponen	Spesifikasi
		• CSS • JQuery • SQL
5	DBMS	MySQL
6	Database Server	PhpMyAdmin
7	Web Server	Xampp
8	Browser	Mozilla Firefox

5.2.1.2 Spesifikasi Perangkat Keras

Pada implementasi perangkat lunak pendampingan disabilitas, digunakan perangkat keras sebagai media untuk melakukan implementasi. Spesifikasi lengkap perangkat keras yang digunakan dalam proses implementasi perangkat lunak pendampingan mahasiswa disabilitas dapat dilihat pada Tabel 5.8.

Tabel 5.8 Spesifikasi Perangkat Keras

No	Nama Komponen	Spesifikasi
1	<i>Device</i>	Notebook Asus A456UF
2	<i>Processor</i>	Intel® Core™ i5-6200U CPU @ 2.30 GHz
3	Memori	8192MB
4	<i>Display</i>	Intel® HD Graphics 520

5.2.2 Implementasi Kelas

Pada bagian implementasi kelas, akan dijelaskan mengenai kelas – kelas yang telah diimplementasikan pada perangkat lunak pendampingan mahasiswa disabilitas. Pada bagian ini akan dijelaskan nama kelas, tipe kelas beserta nama file dari kelas yang telah ada. Penjelasan lebih lengkap mengenai implementasi kelas dapat dilihat dalam Tabel 5.9.

Tabel 5.9 Implementasi Kelas

No	Nama Kelas	Nama File	Tipe Kelas
1	LoginController	LoginController.php	<i>Controller</i>
2	AdminLoginController	AdminLoginController.php	<i>Controller</i>
3	RegisterController	RegisterController.php	<i>Controller</i>
4	HomeController	HomeController.php	<i>Controller</i>
5	JadwalController	JadwalController.php	<i>Controller</i>
6	PendaftaranController	PendaftaranController.php	<i>Controller</i>



No	Nama Kelas	Nama File	Tipe Kelas
7	PelaporanController	PelaporanController.php	Controller
8	PendampinganController	PendampinganController.php	Controller
9	PenjadwalanController	PenjadwalanController.php	Controller
10	ProfileController	ProfileController.php	Controller
11	TutorController	TutorController.php	Controller
12	Izin_pendampingan	izin_pendampingan.php	Model
13	Data_fakulta	Data_fakulta.php	Model
14	Jenis_akun	Jenis_akun.php	Model
15	UsersAdmin	UsersAdmin.php	Model
16	Jadwal_pendampingan	Jadwal_pendampingan.php	Model
17	Jadwal	Jadwal.php	Model
18	Logbook_pendampingan_skripsi	Logbook_pendampingan_skripsi.php	Model
19	Pelaporan	Pelaporan.php	Model
20	Logbook_pendampingan_tugas_akhir	Logbook_pendampingan_tugas_akhir.php	Model
21	Pendampingan_skripsi	Pendampingan_skripsi.php	Model
22	Pendampingan_tugas_akhir	Pendampingan_tugas_akhir.php	Model
23	Pendampingan	Pendampingan.php	Model
24	Respon_laporan	Respon_laporan.php	Model
25	User	User.php	Model
26	File HTML	login.blade.php	View
27	File HTML	adminLogin.blade.php	View
28	File HTML	register.blade.php	View
29	File HTML	app_admin.blade.php	View
30	File HTML	app_sidebar.blade.php	View



No	Nama Kelas	Nama File	Tipe Kelas
31	File HTML	metadata.blade.php	View
32	File HTML	navbar.blade.php	View
33	File HTML	sidebar.blade.php	View
34	File HTML	admin_home.blade.php	View
35	File HTML	admin_izin_pendampingan_detail.blade.php	View
36	File HTML	admin_izin_pendampingan.blade.php	View
37	File HTML	admin_pendaftaran_detail.blade.php	View
38	File HTML	admin_pendaftaran.blade.php	View
39	File HTML	admin_jadwal_pendampingan.blade.php	View
40	File HTML	admin_jadwal_pendampingan_detail.blade.php	View
41	File HTML	admin_detail_tutor_pengajuan_skripsi.blade.php	View
42	File HTML	admin_detail_tutor_pengajuan_tugas_akhir.blade.php	View
43	File HTML	admin_detail_tutor_skripsi.blade.php	View
44	File HTML	admin_detail_tutor_tugas_akhir.blade.php	View
45	File HTML	admin_index_tutor_pengajuan.blade.php	View
46	File HTML	admin_index_tutor_skripsi.blade.php	View
47	File HTML	admin_index_tutor_tugas_akhir.blade.php	View
48	File HTML	admin_pelaporan_detail.blade.php	View
49	File HTML	admin_pelaporan.blade.php	View
50	File HTML	detail_jadwal.blade.php	View
51	File HTML	detail_pendampingan.blade.php	View
52	File HTML	edit_profile.blade.php	View
53	File HTML	home.blade.php	View
54	File HTML	izin_pendampingan.blade.php	View
55	File HTML	jadwal_pendampingan.blade.php	View
56	File HTML	jadwal.blade.php	View



No	Nama Kelas	Nama File	Tipe Kelas
57	File HTML	profile.blade.php	View
58	File HTML	tambah_jadwal.blade.php	View
59	File HTML	ajukan_pendampingan_skripsi.blade.php	View
60	File HTML	ajukan_pendampingan_tugas_akhir.blade.php	View
61	File HTML	logbook.blade.php	View
62	File HTML	logbook_tugas_akhir.blade.php	View
63	File HTML	logbook.blade.php	View
64	File HTML	pelaporan_detail.blade.php	View
65	File HTML	pelaporan.blade.php	View
66	File HTML	pendampingan_skripsi.blade.php	View
67	File HTML	pendampingan_tugas_akhir.blade.php	View

5.2.3 Implementasi Algoritme

Pada subbab implementasi algoritme, akan dijabarkan mengenai kode program yang diimplementasikan pada perangkat lunak pendampingan mahasiswa disabilitas. Implementasi algoritme ini dilakukan berdasarkan perancangan algoritme yang telah dibuat sebelumnya dalam bentuk *pseudocode*. Pada subbab ini akan dijelaskan tiga algoritme yang telah diimplementasikan yaitu pengajuan_pendampingan(), generate() dan konfirmasi_pendampingan().

5.2.3.1 Implementasi Method generate_jadwal()

Method generate_jadwal() adalah *method* yang berfungsi untuk melakukan proses generate jadwal pendampingan secara otomatis. Method generate_jadwal() diperlukan agar *admin* dapat melakukan penjadwalan pendampingan secara cepat dengan melakukan pencocokan antara jadwal mahasiswa disabilitas dan mahasiswa *volunteer*. Implementasi kode program dari *method* generate_jadwal() selengkapnya dapat dilihat pada Tabel 5.10.

Tabel 5.10 Kode Program Method generate_jadwal()

No	Kode Program
1	public function generate_jadwal(Request \$request)
2	{
3	\$tanggal_mulai = \$request->get('waktu_mulai');
4	\$tanggal_selesai = \$request->get('waktu_selesai');
5	\$get_data_pendampingan = \$this->seeInDatabase
	('App\Jadwal_pendampingan',
	array(['tanggal_pendampingan', '>='],



```

        $tanggal_mulai], ['tanggal_pendampingan',
        '!=', $tanggal_selesai));
6
7      if ($get_data_pendampingan > 0) {
8        return redirect('admin_jadwal_pendampingan/') -
9        >with('error', 'telah terdapat jadwal
        pendampingan pada tanggal yang dipilih.');
```

Pada Tabel 5.10 terdapat kode program untuk memulai proses generate jadwal. Pada baris 3-4 dilakukan inisialisasi beberapa variabel, kemudian pada baris 5 dilakukan cek apakah telah terdapat jadwal pendampingan pada tanggal yang dipilih menggunakan *method* *seeInDatabase()*. Apabila telah terdapat jadwal pendampingan maka akan mengembalikan pesan error pada baris 7, apabila tidak ada jadwal pendampingan pada tanggal yang dipilih, maka akan dilanjutkan dengan memanggil *method* *do_generate()* pada baris 9. Hasil dari *method* *do_generate()* akan disimpan ke *database* pada baris 10. Kemudian pada baris 11 akan dilakukan pengembalian pesan yang menandakan jika generate jadwal berhasil.

5.2.3.2 Implementasi Method konfirmasi_pendampingan()

Method *konfirmasi_pendampingan()* adalah *method* yang berfungsi untuk melakukan proses konfirmasi pendampingan. *Method* *konfirmasi_pendampingan()* diperlukan agar mahasiswa *volunteer* dapat melakukan konfirmasi jika pendampingan telah dimulai dan proses pendampingan bisa dimulai. Implementasi kode program dari *method* *konfirmasi_pendampingan()* selengkapnya dapat dilihat pada Tabel 5.11.

Tabel 5.11 Kode Program *Method* *konfirmasi_pendampingan()*

No	Kode Program
1	public function konfirmasi_pendampingan(Request \$request)
2	{
3	if (\$request->has('id_pendampingan')) {
4	\$id_pendampingan=\$request->id_pendampingan;
5	\$status = "Pendampingan Dimulai";
6	\$gagal = "0";
7	\$berhasil = "1";
8	\$id_pendamping = Auth::guard('user')->user()
	->id;
	\$cek_data = \$this->seeInDatabase('App\Jadwal



	pendampingan', ['id' => \$id_pendampingan, 'id_pendamping' => \$id_pendamping]);
9	if (\$cek_data == "true") {
10	Jadwal_pendampingan::where('id', \$id_pendampingan)->where(['id_pendamping' => \$id_pendamping])->update(['status' => \$status]);
11	return response()->json(['status' => \$berhasil]);
12	} else {
13	return response()->json(['status' => \$gagal]);
14	}
15	} else {
16	return response()->json(['status' => \$gagal]);
17	}
18	}

Pada Tabel 5.11 terdapat kode program untuk memulai proses pendampingan. Proses dimulai dengan inisialisasi beberapa variabel antara lain `id_pendampingan`, `status`, `gagal`, `berhasil` dan `id_pendampingan`. Kemudian dilakukan cek apakah data pendampingan tersimpan di *database* menggunakan *method* `seelInDatabase()` pada baris 8. Kemudian pada baris 9 akan dilakukan seleksi sesuai dengan hasil yang dikembalikan oleh *method* `seelInDatabase()`. Kemudian dilanjutkan dengan melakukan *update* status pendampingan pada baris 10 dan mengeluarkan status berhasil pada baris 11. Jika data pendampingan tidak ditemukan maka akan mengeluarkan status gagal pada baris 13 dan baris 14 akan mengeluarkan status gagal apabila tidak ditemukan `id_pendampingan`.

5.2.3.3 Implementasi Method `upload_pengajuan_pendampingan_skripsi()`

Method `upload_pengajuan_pendampingan_skripsi()` merupakan *method* yang berfungsi untuk melakukan pengajuan pendampingan skripsi yang dilakukan oleh mahasiswa disabilitas. Implementasi kode program untuk *method* `upload_pengajuan_pendampingan_skripsi()` dapat dilihat pada Tabel 5.12.

Tabel 5.12 Kode Program *Method* `upload_pengajuan_pendampingan_skripsi()`

No	Kode Program
1	public function <code>upload_pengajuan_pendampingan_skripsi</code>
	(Request \$request, \$id) {
2	\$validationArray = [
	'judul_skripsi' => 'required',
	'nama_pembimbing' => 'required',
	'waktu_mulai' => 'required',
	'waktu_selesai' => 'required',
	'dokumen_pengajuan' => 'required',



```

},
3      $validator = Validator::make($request->all(),
      $validationArray);
4      $dokumen = $request->file('dokumen_pengajuan');
5      $dokumenSaveAsName = time() .
      Auth::guard('user')->user()->email .
      "-dokumen_pengajuan_skripsi.";
      $dokumen->getClientOriginalExtension();
6      $upload_path = 'dokumen_pengajuan_skripsi/';
7      $dokumen_url = $upload_path . $dokumenSaveAsName;
8      $dokumen->move($upload_path, $dokumenSaveAsName);
9
      $pengajuan = new Pendampingan_skripsi([
          'id_disabilitas' => $id,
          'judul_skripsi' => $request->get
          ('judul_skripsi'),
          'nama_pembimbing' => $request->get
          ('nama_pembimbing'),
          'waktu_mulai' => $request->get
          ('waktu_mulai'),
          'waktu_selesai' => $request->get
          ('waktu_selesai'),
          'status' => 'diajukan',
          'dokumen_pengajuan' => $dokumen_url,
      ]);
10     $cek_pengajuan = $this->seeInDatabase
      ('App\Pendampingan_skripsi',
      ['id_disabilitas' => Auth::guard('user')->user()-
      >id]);
11     if ($cek_pengajuan >= 1) {
12         return response()->json(['status' => 'error',
          'message' => 'Maaf anda telah melakukan
          pengajuan sebelumnya.
          Silahkan tunggu konfirmasi']);
13     } else {
14         if ($validator->fails()) {
15             return response()->json(['status' =>
          'error', 'message' => 'Isi semua data
          terlebih dahulu']);
16         } else {
17             $pengajuan->save();
18             return response()->json(['status' =>
          'success', 'message' => 'Pendampingan
          skripsi berhasil diajukan']);
19         }
20     }
21 }
22 }

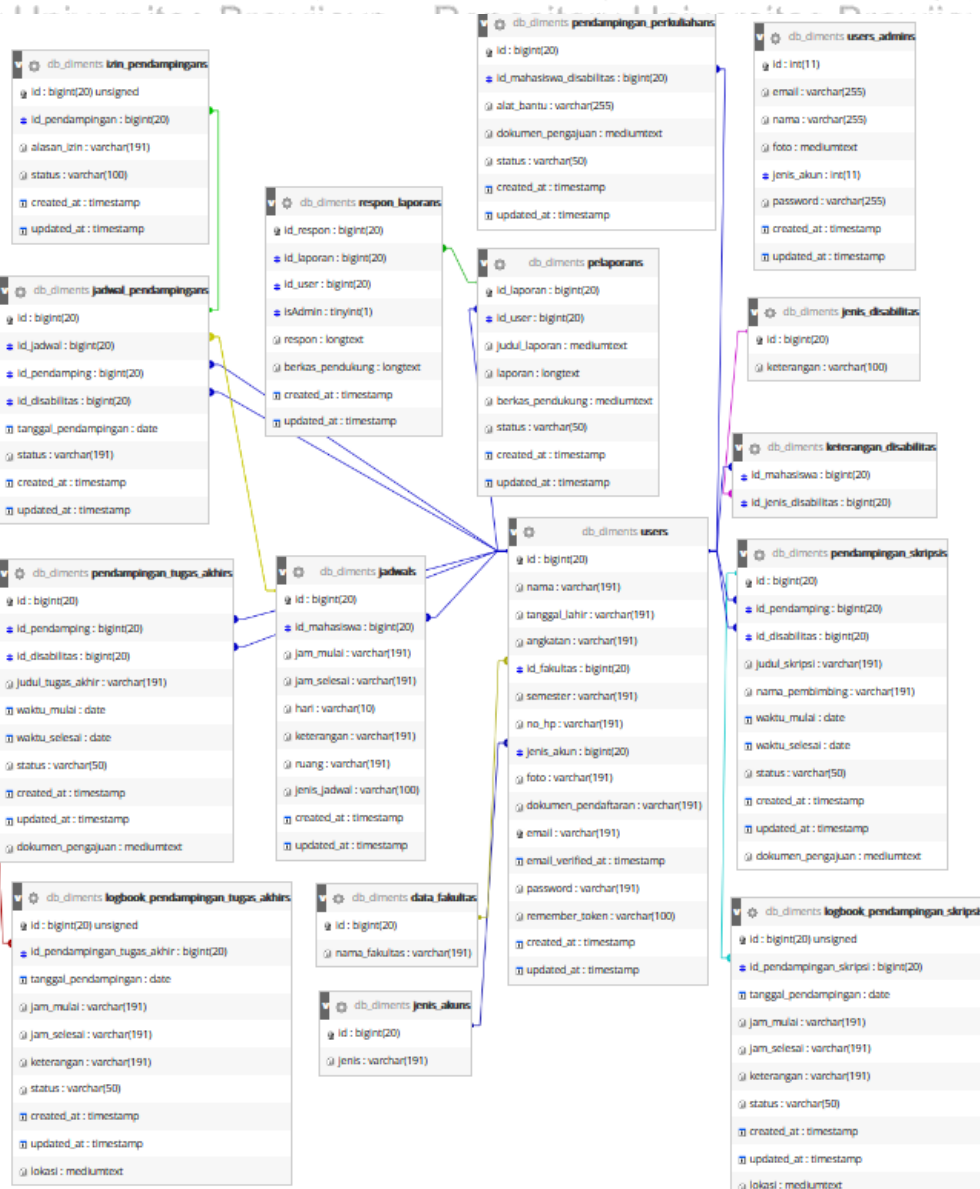
```



Kode program dari *method* `upload_pengajuan_pendampingan_skripsi()` dimulai dengan deklarasi fungsi pada baris 1. Kemudian dilakukan proses validasi masukan yang dilakukan mahasiswa disabilitas. Pada baris 4-7 dilakukan proses pemindahan serta penggantian nama dari dokumen pengajuan yang diunggah oleh mahasiswa disabilitas. Data masukan yang diberikan mahasiswa disabilitas akan diinisialisasi ke dalam variabel pada baris 8. Pada baris 10 akan dilakukan cek apakah pengajuan pendampingan skripsi telah dilakukan sebelumnya, dengan memanggil *method* `seelnDatabase()`. Pada baris 11 akan dilakukan seleksi apakah pengajuan telah dilakukan sebelumnya, jika iya maka akan mengembalikan pesan error pada baris 12 sedangkan jika belum maka dilanjutkan dengan cek apakah terdapat data kosong atau tidak pada baris 14. Jika terdapat data kosong maka akan mengembalikan pesan error. Jika tidak terdapat data kosong maka data pada variabel di baris 8 akan disimpan menggunakan *method* `save()` pada baris 13 dan dilanjutkan dengan mengembalikan pesan sukses telah berhasil melakukan pengajuan pendampingan skripsi.

5.2.4 Implementasi Basis Data

Implementasi basis data dilakukan dengan pembuatan basis data perangkat lunak pendampingan mahasiswa disabilitas dengan nama `db_diment`. Pada implementasi basis data yang telah dilakukan didapatkan hasil yaitu terdapat 16 tabel pada basis data yang terdiri dari `data_fakultas`, `izin_pendampingans`, `jadwals`, `jadwal_pendampingans`, `jenis_akuns`, `jenis_disabilitas`, `keterangan_disabilitas`, `logbook_pendampingan_skripsi`, `logbook_pendampingan_tugas_akhirs`, `pelaporans`, `pendampingan_perkuliahans`, `pendampingan_skripsi`, `pendampingan_tugas_akhirs`, `users` dan `users_admins`. Terdapat relasi antara tabel pada basis data yang telah diimplementasikan. Hasil implementasi basis data dapat dilihat dalam Gambar 5.32.



Gambar 5.31 Implementasi Basis Data

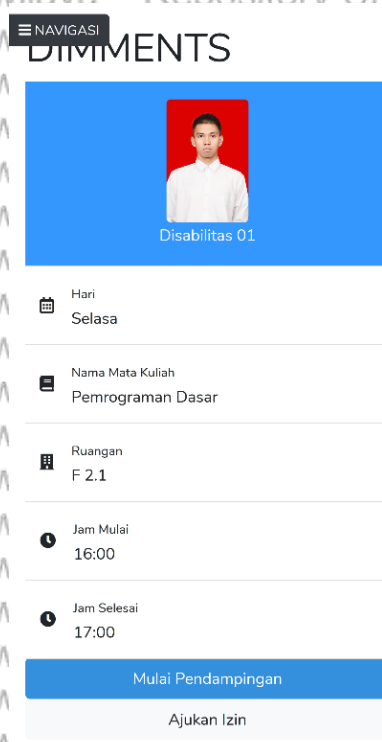
5.2.5 Implementasi Antarmuka

Pada implementasi antarmuka, akan dijabarkan bagaimana hasil implementasi antarmuka yang dilakukan pada perangkat lunak pendampingan mahasiswa disabilitas. Implementasi antarmuka ini dilakukan dengan acuan perancangan antarmuka yang telah dilakukan sebelumnya. Pada pengembangan perangkat lunak pendampingan mahasiswa disabilitas, digunakan beberapa *framework* untuk membantu proses implementasi antarmuka perangkat lunak pendampingan mahasiswa disabilitas ini. *Framework* yang digunakan adalah bootstrap dan jquery. Pada subbab ini pula dijabarkan beberapa antarmuka yang telah diimplementasikan antara lain halaman pengajuan pendampingan, generate jadwal dan detail pendampingan.



5.2.5.1 Implementasi Antarmuka Halaman Detail Pendampingan

Pada halaman detail pendampingan, terdapat tombol mulai pendampingan. Tombol mulai pendampingan ini akan memanggil fungsi *scan qr code* sehingga mahasiswa *volunteer* dapat melakukan pemindaian pada *qr code* pendampingan yang tersedia pada akun mahasiswa disabilitas. Ketika proses pemindaian berhasil maka akan menjalankan fungsi *konfirmasi_pendampingan()* untuk memulai proses pendampingan. Selain tombol mulai pendampingan, terdapat tombol ajukan izin untuk memulai proses pengajuan izin pendampingan dan beberapa informasi yang ditampilkan terkait dengan pendampingan seperti hari pendampingan, nama mata kuliah, ruangan, jam mulai dan selesai. Hasil implementasi antarmuka halaman detail pendampingan dapat dilihat dalam Gambar 5.33.



Gambar 5.32 Implementasi Antarmuka Halaman Detail Pendampingan

5.2.5.2 Implementasi Antarmuka Halaman Pengajuan Pendampingan Skripsi

Halaman pengajuan pendampingan skripsi akan digunakan oleh mahasiswa disabilitas untuk melakukan proses pengajuan pendampingan skripsi yang akan dilakukannya. Pada halaman pengajuan pendampingan skripsi, terdapat lima kolom masukan yang disediakan. Kolom masukan yang disediakan antara lain masukan judul skripsi, nama pembimbing, waktu mulai, waktu selesai dan dokumen pendaftaran. Kemudian terdapat tombol ajukan pendampingan dimana tombol ini berfungsi untuk memulai proses unggah data pengajuan yang dimasukan oleh mahasiswa disabilitas ke *database*. Hasil implementasi



antarmuka dari halaman pengajuan pendampingan skripsi dapat dilihat dalam Gambar 5.34.

NAVIGASI

IMPLEMENTATIONS

Ajukan Pendampingan Skripsi

Judul Skripsi

Nama Pembimbing

Waktu Mulai

mm / dd / yyyy

Waktu Selesai

mm / dd / yyyy

Dokumen Pendaftaran

Pilih File

Browse

Ajukan Pendampingan

Gambar 5.33 Implementasi Antarmuka Halaman Pengajuan Pendampingan Skripsi

5.2.5.3 Implementasi Antarmuka Generate Jadwal Pendampingan

Antarmuka generate jadwal pendampingan digunakan oleh aktor admin untuk melakukan generate jadwal pendampingan mahasiswa disabilitas. Antarmuka generate jadwal pendampingan terdapat pada halaman jadwal pendampingan admin. Pada antarmuka generate jadwal pendampingan terdapat dua masukan yaitu waktu mulai dan waktu selesai dimana masukan ini akan digunakan oleh admin untuk memasukan pilihan tanggal mulai dan selesai dari jadwal yang akan di generate. Selain masukan waktu mulai dan waktu selesai, terdapat tombol generate jadwal yang berfungsi untuk memanggil fungsi yang akan menjalankan mekanisme pencocokan jadwal antara mahasiswa *volunteer* dan mahasiswa disabilitas. Hasil dari antarmuka generate jadwal dapat dilihat dalam Gambar 5.35.



DIMMENTS

[Home](#)
[Jadwal Pendampingan](#)
[Jadwal Kuliah](#)
[Pendaftaran](#)
[Pengajuan Izin](#)

Logout

Jadwal Pendampingan

Tambah Jadwal Pendampingan

Generate Jadwal

Show 10 entries

Search:

#	Foto	Nama	Tanggal Pendampingan	Jam	Status	Action
1		Disabilitas 01	2019-12-24	10:00 - 11:30	dijadwalkan	Selengkapnya

Showing 1 to 1 of 1 entries

Previous

1

Next

DIMMENTS

[Home](#)
[Jadwal Pendampingan](#)
[Jadwal Kuliah](#)
[Pendaftaran](#)
[Pengajuan Izin](#)

Logout

Jadwal Pendampingan

Tambah Jadwal Pendampingan

Generate Jadwal

Show 10 entries

Search:

#	Foto	Nama	Tanggal Pendampingan	Jam	Status	Action
1		Disabilitas 01	2019-12-24	10:00 - 11:30	dijadwalkan	Selengkapnya

Showing 1 to 1 of 1 entries

Previous

1

Next

Generate Jadwal

Waktu Mulai

mm / dd / yyyy

Waktu Selesai

mm / dd / yyyy

Generate Jadwal

Gambar 5.34 Implementasi Antarmuka Generate Jadwal

BAB 6 PENGUJIAN

Pada bab pengujian, akan dilaksanakan proses uji terhadap perangkat lunak yang telah dikembangkan pada tahap implementasi. Tahap pengujian bertujuan untuk mengetahui fungsi – fungsi yang terdapat pada perangkat lunak dapat bekerja sesuai dengan kebutuhan yang telah didefinisikan. Dalam melakukan proses uji perangkat lunak pendampingan mahasiswa disabilitas ini, akan dilakukan menggunakan tiga metode uji antara lain pengujian unit, pengujian validasi dan pengujian kompatibilitas.

6.1 Pengujian Unit

Pengujian unit adalah pengujian yang dilakukan dengan menguji alur dari perangkat lunak. Pengujian ini dilakukan dengan menguji seluruh jalur independen yang didapatkan dari *pseudocode* perangkat lunak. Pada pengujian unit ini, pengujian akan dilakukan terhadap tiga *method* yaitu *generate_jadwal()*, *konfirmasi_pendampingan()* dan *upload_pengajuan_pendampingan_skripsi()*.

6.1.1 Pengujian Unit pada *Method generate_jadwal()*

Method generate_jadwal() berfungsi untuk melakukan generate jadwal pendampingan yang akan digunakan oleh aktor admin. *Method generate_jadwal()* memiliki *pseudocode* seperti pada Tabel 6.1.

1. *Pseudocode method generate_jadwal()*

Tabel 6.1 *Pseudocode method generate_jadwal()*

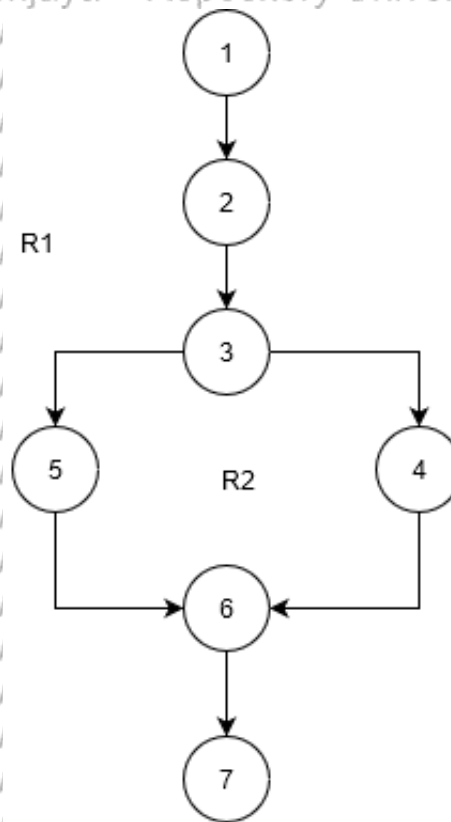
No	Pseudocode	Node
1	Begin	1
2	Declaration tanggal_mulai, tanggal_selesai	2
3	Call method seeInDatabase()	
4	If (output of seeInDatabase() == true)	3
5	Print "telah terdapat jadwal pendampingan pada tanggal yang dipilih."	4
6	Else	5
7	Call method do generate()	
8	Inisialization data from method do generate() to variable hasil_jadwal	
9	Do save hasil_jadwal to Database.	
10	Print "Generate jadwal berhasil"	
11	Endif	6
12	End	7



2. Basic Path Testing

a. Flow Graph

Pseudocode method generate_jadwal() memiliki *flow graph* seperti dalam Gambar 6.1.



Gambar 6.1 Basic Path Pseudocode generate_jadwal()

b. Cyclometric Complexity

- $V(G) = R = 2$
- $V(G) = E - N + 2 = 7 - 7 + 2 = 2$
- $V(G) = P + 1 = 1 + 1 = 2$

c. Independent Path

- Jalur 1 = 1 - 2 - 3 - 4 - 6 - 7
- Jalur 2 = 1 - 2 - 3 - 5 - 6 - 7

Setelah *flow graph* dari *method* generate_jadwal() dibuat, maka dapat ditentukan *cyclometric complexity* untuk menentukan jumlah jalur independen yang menjadi kasus uji untuk *method* generate_jadwal(). Kasus uji dapat ditentukan dengan menentukan jalur independen yang dapat dilihat pada poin c. Penjelasan kasus uji dari jalur independen dapat dilihat pada Tabel 6.2.



Tabel 6.2 Test Case Method generate_jadwal()

No. Jalur	Prosedur Pengujian	Expected Result	Actual Result	Status
Jalur 1	Variabel tanggal_mulai dan tanggal_selesai di inisialisasi, dengan kondisi sudah terdapat jadwal pendampingan yang tersimpan di database di antara tanggal tersebut. Data inputan : - Tanggal_mulai : 04/06/2020 - Tanggal_selesai : 04/25/2020	Jadwal pendampingan gagal di generate dan sistem menampilkan informasi “telah terdapat jadwal pendampingan pada tanggal yang dipilih”	Jadwal pendampingan gagal di generate dan sistem menampilkan informasi “telah terdapat jadwal pendampingan pada tanggal yang dipilih”	Valid
Jalur 2	Variabel tanggal_mulai dan tanggal_selesai di inisialisasi, dengan kondisi belum ada jadwal pendampingan yang tersimpan di database di antara tanggal tersebut. Data inputan : - Tanggal_mulai : 05/01/2020 - Tanggal_selesai : 05/20/2020	Jadwal pendampingan berhasil di generate dan sistem menampilkan informasi “Generate jadwal berhasil”	Jadwal pendampingan berhasil di generate dan sistem menampilkan informasi “Generate jadwal berhasil”	Valid

6.1.2 Pengujian Unit pada Method konfirmasi_pendampingan()

Method konfirmasi_pendampingan() merupakan *method* yang berfungsi kepada mahasiswa yang berstatus sebagai pendamping untuk melakukan konfirmasi kehadiran pendampingannya pada jadwal yang dipilih. Method konfirmasi_pendampingan() memiliki *pseudocode* yang dapat dilihat pada Tabel 6.3.

1. Pseudocode method konfirmasi_pendampingan()

Tabel 6.3 Pseudocode method konfirmasi_pendampingan()

No	Pseudocode	Node
1	Begin	1
2	If (POST[id pendampingan] is exist)	2

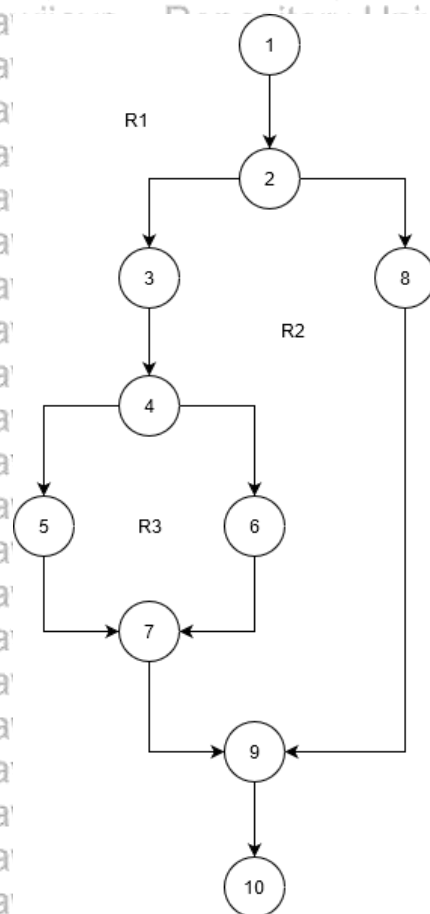


No	Pseudocode	Node
3	Inisialization id pendampingan, status, gagal, berhasil, id_pendamping	3
4	Call method seeInDatabase()	
5	Inisialization cek data with output from seeInDatabase()	
6	If(cek data is true)	4
7	Update status jadwal_pendampingan where id = id_pendampingan and id_pendamping = id_pendamping	5
8	Print 1	
9	Else	6
10	Print 0	
11	Endif	7
12	Else	8
13	Print 0	
14	Endif	9
15	End	10

2. Basic Path Testing

a. Flow Graph

Pseudocode method konfirmasi_pendampingan() memiliki *flow graph* seperti dalam Gambar 6.2.



Gambar 6.2 Basic Path Pseudocode konfirmasi_pendampingan()

b. Cyclomatic Complexity

$$V(G) = R + 1$$

$$V(G) = E - N + 2 = 11 - 10 + 2 = 3$$

$$V(G) = P + 1 = 2 + 1 = 3$$

c. Independent Path

$$\text{Jalur 1} = 1 - 2 - 8 - 9 - 10$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 5 - 7 - 9 - 10$$

$$\text{Jalur 3} = 1 - 2 - 3 - 4 - 5 - 6 - 9 - 10$$

Setelah flow graph dari method konfirmasi_pendampingan() dibuat, maka dapat ditentukan cyclomatic complexity untuk menentukan jumlah jalur independen yang menjadi kasus uji untuk method konfirmasi_pendampingan(). Kasus uji dapat ditentukan dengan menentukan jalur independen yang dapat dilihat pada poin c. Penjelasan kasus uji dari jalur independen dapat dilihat pada Tabel 6.4.



Tabel 6.4 Test Case Method konfirmasi_pendampingan()

No. Jalur	Prosedur Pengujian	Expected Result	Actual Result	Status
Jalur 1	Variabel id_pendamping di inisialisasi, dengan kondisi id_pendampingan tidak diisi. Data inputan : - Id_pendamping : 2	Pendampingan tidak terkonfirmasi dan sistem menampilkan informasi "0".	Pendampingan tidak terkonfirmasi dan sistem menampilkan informasi "0".	Valid.
Jalur 2	Variabel id_pendampingan dan id_pendamping di inisialisasi, dengan kondisi id_pendamping tidak sama dengan data pada jadwal pendampingan di database. Data inputan : - Id_pendampingan : 3 - Id_pendamping : 25	Pendampingan tidak terkonfirmasi dan sistem menampilkan informasi "0".	Pendampingan tidak terkonfirmasi dan sistem menampilkan informasi "0".	Valid.
Jalur 3	Variabel id_pendampingan dan id_pendamping di inisialisasi, dengan kondisi id_pendamping dan id_pendampingan sama dengan data pada jadwal pendampingan di database. Data inputan : - Id_pendampingan : 7 - Id_pendamping : 4	Pendampingan terkonfirmasi dan sistem menampilkan informasi "1".	Pendampingan terkonfirmasi dan sistem menampilkan informasi "1".	Valid.

6.1.3 Pengujian Unit pada Method upload_pengajuan_pendampingan_skripsi()

Method upload_pengajuan_pendampingan_skripsi() merupakan method yang berfungsi agar mahasiswa disabilitas dapat mengajukan pendampingan skripsinya. Pengajuan pendampingan skripsi dapat dilakukan apabila mahasiswa disabilitas belum pernah mengajukan pendampingan skripsi sebelumnya. Pseudocode method upload_pengajuan_pendampingan_skripsi() dapat dilihat pada Tabel 6.5.

1. Pseudocode method upload_pengajuan_pendampingan_skripsi()

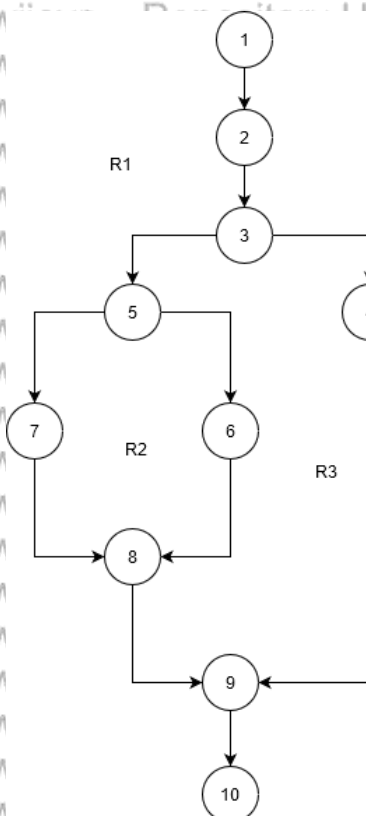
Tabel 6.5 *Pseudocode method upload_pengajuan_pendampingan_skripsi()*

No	Pseudocode	Node
1	Begin	1
2	Declaration judul_skripsi, nama_pembimbing, waktu_mulai, waktu_selesai, dokumen_pengajuan	2
3	Instance pengajuan from class Pendampingan_skripsi	
4	Call method seeInDatabase()	
5	If(output of seeInDatabase() == true)	3
6	Print "Maaf anda telah melakukan pengajuan sebelumnya. Silahkan tunggu konfirmasi"	4
7	Else	5
8	If(input is null)	
9	Print "Isi semua data terlebih dahulu"	6
10	Else	7
11	Do save data pengajuan pendampingan skripsi to Database.	
12	Print "Pendampingan skripsi berhasil diajukan"	
13	Endif	8
14	Endif	9
15	End	10

2. Basic Path Testing

a. Flow Graph

Pseudocode method upload_pengajuan_pendampingan_skripsi() memiliki *flow graph* seperti dalam Gambar 6.3.



Gambar 6.3 Basic Path Pseudocode upload_pengajuan_pendampingan_skripsi()

b. Cyclometric Complexity

- $V(G) = R = 3$
- $V(G) = E - N + 2 = 11 - 10 + 2 = 3$
- $V(G) = P + 1 = 2 + 1 = 3$

c. Independent Path

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

Setelah *flow graph* dari *method* `upload_pengajuan_pendampingan_skripsi()` dibuat, maka dapat ditentukan *cyclometric complexity* untuk menentukan jumlah jalur independen yang menjadi kasus uji untuk *method* `upload_pengajuan_pendampingan_skripsi()`. Kasus uji dapat ditentukan dengan menentukan jalur independen yang dapat dilihat pada poin c. Penjelasan kasus uji dari jalur independen dapat dilihat pada Tabel 6.6.

Tabel 6.6 Test Case Method upload_pengajuan_pendampingan_skripsi()

No. Jalur	Prosedur Pengujian	Expected Result	Actual Result	Status
Jalur	Variabel judul_skripsi,	Pengajuan	Pengajuan	Valid.



No. Jalur	Prosedur Pengujian	Expected Result	Actual Result	Status
1	nama_pembimbing, waktu_mulai, waktu_selesai, dokumen_pengajuan diinisialisasi. Pengujian dilakukan terhadap user yang telah mengajukan pendampingan skripsi sebelumnya. Data input : - judul_skripsi : aplikasi pencarian lowongan kerja berbasis web - nama_pembimbing : Aditya S.Kom - waktu_mulai : 2019-08-12 - waktu_selesai : 2020-05-10 - dokumen_pengajuan : pengajuan_pendampingan/fachril@gmail.com.png	pendampingan skripsi gagal dan sistem menampilkan informasi "Maaf anda telah melakukan pengajuan sebelumnya. Silahkan tunggu konfirmasi".	pendampingan skripsi gagal dan sistem menampilkan informasi "Maaf anda telah melakukan pengajuan sebelumnya. Silahkan tunggu konfirmasi".	
Jalur 2	Variabel judul_skripsi, waktu_mulai, waktu_selesai, dokumen_pengajuan diinisialisasi. Pengujian dilakukan	Pengajuan pendampingan skripsi gagal dan sistem menampilkan informasi "Isi semua data	Pengajuan pendampingan skripsi gagal dan sistem menampilkan informasi "Isi	Valid.



No. Jalur	Prosedur Pengujian	Expected Result	Actual Result	Status
	terhadap user yang belum melakukan pengajuan pendampingan. Variabel nama_pembimbing tidak diinisialisasi. Data input : - judul_skripsi : aplikasi pencarian volunteer berbasis android - waktu_mulai : 2019-06-12 - waktu_selesai : 2019-12-10 - dokumen_pengajuan : pengajuan_pendampingan/fano@gmail.com.png	terlebih dahulu".	semua data terlebih dahulu".	
Jalur 3	Variabel judul_skripsi, nama_pembimbing, waktu_mulai, waktu_selesai, dokumen_pengajuan diinisialisasi. Pengujian dilakukan terhadap user yang belum pernah mengajukan pendampingan skripsi sebelumnya.	Pengajuan pendampingan skripsi berhasil, sistem menampilkan informasi "Pendampingan skripsi berhasil diajukan".	Pengajuan pendampingan skripsi berhasil, sistem menampilkan informasi "Pendampingan skripsi berhasil diajukan".	Valid



No. Jalur	Prosedur Pengujian	Expected Result	Actual Result	Status
	Data input : - judul_skrripsi : aplikasi pembelajaran Bahasa jawa berbasis android - nama_pembi mbing : Aditya S.Kom - waktu_mulai : 2019-08-12 - waktu_selesai : 2020-05-10 - dokumen_pengajuan : pengajuan_pendampingan/ faizfanani@gmail.com.png			

6.2 Pengujian Validasi

Pengujian validasi diperlukan untuk memastikan fungsionalitas yang telah di implementasikan telah sesuai dengan *use case scenario* yang telah di definisikan tanpa melihat sumber kode dari perangkat lunak. Pengujian validasi dilakukan dengan melakukan percobaan pada setiap fungsi dan melihat keluaran yang dihasilkan.

6.2.1 Pengujian Validasi pada Fungsional Login

Pengujian pada fungsional login dilakukan dengan memasukan email dan password pada halaman login. Terdapat tiga skenario uji yang dapat dilihat pada Tabel 6.7.

Tabel 6.7 Skenario Uji Validasi Fungsional Login

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa	Login berhasil dan	Login berhasil	Valid.



No	Skenario	Expected	Actual	Status
1	<i>volunteer</i> mengisi email dan password sesuai dengan informasi yang tersimpan pada <i>database</i>. Data masukan : - Email : agenthrn@gmail.com - Password : 12345678	halaman <i>dashboard</i> ditampilkan.	dan halaman <i>dashboard</i> ditampilkan.	
2	Aktor mahasiswa <i>volunteer</i> mengisi username dan password tidak sesuai dengan informasi yang tersimpan pada <i>database</i>. Data masukan : - Email : agenthrn@gmail.com - Password : 12345	Login gagal dan sistem menampilkan informasi "Kami tidak menemukan akun terkait."	Login gagal dan sistem menampilkan informasi "Kami tidak menemukan akun terkait."	Valid.
3	Aktor mahasiswa <i>volunteer</i> mengisi username dan password tidak diisi. Data masukan : Email : agenthrn@gmail.com	Sistem menampilkan informasi " <i>Please fill out this field</i> "	Sistem menampilkan informasi " <i>Please fill out this field</i> "	Valid.

6.2.2 Pengujian Validasi pada Fungsional Login Admin

Pengujian pada fungsional login admin dilakukan dengan memasukkan email dan password pada halaman login admin. Terdapat tiga skenario uji yang dapat dilihat pada Tabel 6.8.



Tabel 6.8 Skenario Uji Validasi Fungsional Login Admin

No	Skenario	Expected	Actual	Status
1	Aktor admin mengisi email dan password sesuai dengan informasi yang tersimpan pada database. Data masukan : - Email : admin@diment.com - Password : 12345678	Login berhasil dan halaman <i>dashboard</i> ditampilkan.	Login berhasil dan halaman <i>dashboard</i> ditampilkan.	Valid.
2	Aktor admin mengisi username dan password tidak sesuai dengan informasi yang tersimpan pada database. Data masukan : - Email : admin@diment.com - Password : 12345	Login gagal dan sistem menampilkan informasi "Kami tidak menemukan akun terkait."	Login gagal dan sistem menampilkan informasi "Kami tidak menemukan akun terkait."	Valid.
3	Aktor admin mengisi username dan password tidak diisi. Data masukan : - Email : admin@diment.com	Sistem menampilkan informasi "Please fill out this field"	Sistem menampilkan informasi "Please fill out this field"	Valid.

6.2.3 Pengujian Validasi pada Fungsional Hapus Jadwal Pendampingan

Pengujian pada fungsional hapus jadwal pendampingan dilakukan dengan memilih jadwal yang akan di hapus dan menekan tombol selengkapnya, kemudian menekan tombol hapus. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.9.



Tabel 6.9 Skenario Uji Validasi Fungsional Hapus Jadwal Pendampingan

No	Skenario	Expected	Actual	Status
1	Aktor admin memilih jadwal pendampingan dan menekan tombol selengkapnya, kemudian menekan tombol hapus.	Jadwal pendampingan berhasil di hapus dan sistem menampilkan informasi “Anda berhasil hapus jadwal pendampingan”.	Jadwal pendampingan berhasil di hapus dan sistem menampilkan informasi “Anda berhasil hapus jadwal pendampingan”.	Valid.

6.2.4 Pengujian Validasi pada Fungsional Pilih Pendamping Pengganti

Pengujian pada fungsional pilih pendamping pengganti dilakukan oleh aktor admin dengan memilih jadwal yang telah diajukan izin kemudian melakukan perubahan pada pilihan pendamping yang tersedia dan menekan tombol simpan. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.10.

Tabel 6.10 Skenario Uji Validasi Fungsional Pilih Pendamping Pengganti

No	Skenario	Expected	Actual	Status
1	Aktor admin memilih jadwal yang telah diajukan izin kemudian melakukan perubahan pada pilihan pendamping yang tersedia dan menekan tombol simpan	Pendamping berhasil diubah dan sistem menampilkan informasi “Anda berhasil mengubah pendamping”.	Pendamping berhasil diubah dan sistem menampilkan informasi “Anda berhasil mengubah pendamping”.	Valid.

6.2.5 Pengujian Validasi pada Fungsional Generate Jadwal Pendampingan

Pengujian pada fungsional ubah pendamping dilakukan oleh aktor admin dengan mengisi waktu mulai dan waktu selesai kemudian menekan tombol generate. Terdapat dua skenario uji yang dapat dilihat pada Tabel 6.11.

Tabel 6.11 Skenario Uji Validasi Fungsional Generate Jadwal Pendampingan

No	Skenario	Expected	Actual	Status
1	Aktor admin mengisi waktu mulai dan waktu selesai kemudian menekan tombol	Generate jadwal berhasil. Halaman jadwal pendampingan ditampilkan dan	Generate jadwal berhasil. Halaman jadwal pendampingan ditampilkan dan	Valid.



No	Skenario	Expected	Actual	Status
1	generate. Data masukan : - Waktu mulai : 03/11/2020 - Waktu selesai : 03/21/2020	sistem menampilkan informasi "Generate jadwal berhasil".	sistem menampilkan informasi "Generate jadwal berhasil".	
2	Aktor admin mengisi waktu mulai dan waktu selesai tidak diisi kemudian menekan tombol generate. Data masukan : - Waktu mulai : 03/11/2020	Sistem menampilkan informasi "Please fill out this field"	Sistem menampilkan informasi "Please fill out this field"	Valid.

6.2.6 Pengujian Validasi pada Fungsional Lihat Pengajuan Izin

Pengujian pada fungsional lihat pengajuan izin dilakukan oleh aktor admin dengan membuka halaman pengajuan izin. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.12.

Tabel 6.12 Skenario Uji Validasi Fungsional Lihat Pengajuan Izin

No	Skenario	Expected	Actual	Status
1	Aktor admin membuka halaman pengajuan izin.	Halaman pengajuan izin ditampilkan.	Halaman pengajuan izin ditampilkan.	Valid.

6.2.7 Pengujian Validasi pada Fungsional Lihat Pengajuan Pendampingan Perkuliahan

Pengujian pada fungsional lihat pengajuan pendampingan perkuliahan dilakukan oleh aktor admin dengan membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan perkuliahan. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.13.

Tabel 6.13 Skenario Uji Validasi Fungsional Lihat Pengajuan Pendampingan Perkuliahan

No	Skenario	Expected	Actual	Status
1	Aktor admin membuka halaman pengajuan pendampingan, kemudian menekan	Daftar pengajuan pendampingan perkuliahan	Daftar pengajuan pendampingan perkuliahan	Valid.



No	Skenario	Expected	Actual	Status
	tombol tab pendampingan perkuliahan.	ditampilkan.	ditampilkan.	

6.2.8 Pengujian Validasi pada Fungsional Lihat Pengajuan Pendampingan Skripsi

Pengujian pada fungsional lihat pengajuan pendampingan skripsi dilakukan oleh aktor admin dengan membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan skripsi. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.14.

Tabel 6.14 Skenario Uji Validasi Fungsional Lihat Pengajuan Pendampingan Skripsi

No	Skenario	Expected	Actual	Status
1	Aktor admin membuka halaman pengajuan pendampingan, kemudian menekan tombol tab pendampingan skripsi.	Daftar pengajuan pendampingan skripsi ditampilkan.	Daftar pengajuan pendampingan skripsi ditampilkan.	Valid.

6.2.9 Pengujian Validasi pada Fungsional Lihat Pengajuan Pendampingan Tugas Akhir

Pengujian pada fungsional lihat pengajuan pendampingan tugas akhir dilakukan oleh aktor admin dengan membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan tugas akhir. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.15.

Tabel 6.15 Skenario Uji Validasi Fungsional Lihat Pengajuan Pendampingan Tugas Akhir

No	Skenario	Expected	Actual	Status
1	Aktor admin membuka halaman pengajuan pendampingan, kemudian menekan tombol tab pendampingan tugas akhir.	Daftar pengajuan pendampingan tugas akhir ditampilkan.	Daftar pengajuan pendampingan tugas akhir ditampilkan.	Valid.



6.2.10 Pengujian Validasi pada Fungsional Lihat Jadwal Mahasiswa

Pengujian pada fungsional lihat jadwal mahasiswa dilakukan oleh aktor admin. Skenario pengujian diawali dengan membuka halaman jadwal. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.16.

Tabel 6.16 Skenario Uji Validasi Fungsional Lihat Jadwal Mahasiswa

No	Skenario	Expected	Actual	Status
1	Aktor admin membuka halaman jadwal.	Halaman jadwal ditampilkan.	Halaman jadwal ditampilkan.	Valid.

6.2.11 Pengujian Validasi pada Fungsional Lihat Pendaftaran

Pengujian pada fungsional lihat pendaftaran dilakukan oleh aktor admin. Skenario pengujian diawali dengan melakukan login dan membuka halaman pendaftaran. Setelah itu daftar pendaftaran mahasiswa ditampilkan. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.17.

Tabel 6.17 Skenario Uji Validasi Fungsional Lihat Pendaftaran

No	Skenario	Expected	Actual	Status
1	Aktor admin melakukan login dilanjutkan dengan membuka halaman pendaftaran.	Daftar pendaftaran mahasiswa ditampilkan.	Daftar pendaftaran mahasiswa ditampilkan.	Valid.

6.2.12 Pengujian Validasi pada Fungsional Lihat Detail Pendaftaran

Pengujian pada fungsional lihat detail pendaftaran dilakukan oleh aktor admin. Skenario pengujian diawali dengan melakukan login dan membuka halaman pendaftaran. Setelah daftar pendaftaran mahasiswa ditampilkan, admin menekan tombol selengkapnya pada pengajuan pendaftaran yang dipilih. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.18.

Tabel 6.18 Skenario Uji Validasi Fungsional Lihat Detail Pendaftaran

No	Skenario	Expected	Actual	Status
1	Aktor admin melakukan login dilanjutkan dengan membuka halaman pendaftaran. Setelah daftar pendaftaran ditampilkan, admin menekan tombol selengkapnya pada pengajuan pendaftaran	Detail pendaftaran mahasiswa yang dipilih ditampilkan.	Detail pendaftaran mahasiswa yang dipilih ditampilkan.	Valid.



No	Skenario	Expected	Actual	Status
	yang dipilih.			

6.2.13 Pengujian Validasi pada Fungsional Ubah Jadwal Mahasiswa

Pengujian pada fungsional ubah jadwal mahasiswa dilakukan oleh aktor admin dengan membuka halaman jadwal, kemudian menekan tombol ubah pada jadwal yang dipilih. Terdapat dua skenario uji yang dapat dilihat pada Tabel 6.19.

Tabel 6.19 Skenario Uji Validasi Fungsional Lihat Jadwal Mahasiswa

No	Skenario	Expected	Actual	Status
1	Aktor admin membuka halaman jadwal, kemudian menekan tombol ubah pada jadwal yang dipilih, kemudian melakukan perubahan pada data yang diinginkan. Data masukan : - Keterangan jadwal : Pelajaran Metpen - Hari : Selasa - Ruangan R.1.1 - Jam Mulai : 10:00 AM - Jam Selesai : 11:00 AM - Jenis Jadwal : Tetap	Data jadwal mahasiswa berhasil diubah dan sistem menampilkan informasi "Jadwal berhasil diubah".	Data jadwal mahasiswa berhasil diubah dan sistem menampilkan informasi "Jadwal berhasil diubah".	Valid.
2	Aktor admin membuka halaman jadwal, kemudian menekan tombol ubah pada jadwal yang dipilih, kemudian melakukan perubahan pada data yang diinginkan dengan mengosongkan data	Sistem menampilkan informasi "Please fill out this field"	Sistem menampilkan informasi "Please fill out this field"	Valid.



No	Skenario	Expected	Actual	Status
	ruangan.			
	Data masukan :			
	- Keterangan			
	- jadwal :			
	- Pelajaran			
	- Metpen			
	- Hari : Selasa			
	- Jam Mulai :			
	10:00 AM			
	- Jam Selesai :			
	11:00 AM			
	- Jenis Jadwal :			
	Tetap			

6.2.14 Pengujian Validasi pada Fungsional Logout

Pengujian pada fungsional lihat detail jadwal mahasiswa dilakukan oleh aktor admin dengan menekan tombol logout. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.20.

Tabel 6.20 Skenario Uji Validasi Fungsional Logout

No	Skenario	Expected	Actual	Status
1	Aktor admin menekan tombol logout, kemudian menekan tombol "Ya" pada pesan konfirmasi melakukan logout.	Logout berhasil dilakukan dan halaman login ditampilkan.	Logout berhasil dilakukan dan halaman login ditampilkan.	Valid.

6.2.15 Pengujian Validasi pada Fungsional Ubah Profil

Pengujian pada fungsional ubah profil dilakukan oleh aktor mahasiswa disabilitas dengan membuka halaman profil, kemudian menekan tombol ubah profil anda, melakukan perubahan pada data profil dan menekan tombol simpan. Terdapat dua skenario uji yang dapat dilihat pada Tabel 6.21.

Tabel 6.21 Skenario Uji Validasi Fungsional Ubah Profil

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa disabilitas / mahasiswa volunteer menekan	Perubahan profil berhasil disimpan ke database dan	Perubahan profil berhasil disimpan ke	Valid.



No	Skenario	Expected	Actual	Status
1	tombol profil, kemudian menekan tombol ubah profil anda, melakukan perubahan pada data profil dan menekan tombol simpan. Data masukan : - Nama : Harun Hasibuan - Tanggal Lahir : 05/17/1998 - Angkatan : 2016 - Semester : 7 - Nomor Handphone : 081336055105 - E-Mail Address : agenthrn@gmail.com	sistem menampilkan informasi "Berhasil Ubah Profil".	database dan sistem menampilkan informasi "Berhasil Ubah Profil".	
2	Aktor mahasiswa disabilitas menekan tombol profil, kemudian menekan tombol ubah profil anda, melakukan perubahan pada data profil dengan data nomor handphone tidak diisi dan menekan tombol simpan. Data masukan : - Nama : Harun Hasibuan - Tanggal Lahir : 05/17/1998 - Angkatan : 2016 - Semester : 7 - E-Mail Address : agenthrn@gmail.com	Sistem menampilkan informasi "Please enter a number"	Sistem menampilkan informasi "Please enter a number"	Valid.



No	Skenario	Expected	Actual	Status
	com			

6.2.16 Pengujian Validasi pada Fungsional Ajukan Pendampingan Skripsi

Pengujian pada fungsional ajukan pendampingan skripsi dilakukan oleh aktor mahasiswa disabilitas dengan membuka halaman pendampingan skripsi, kemudian menekan tombol ajukan, kemudian mengisi informasi yang diperlukan dan menekan tombol ajukan pendampingan. Terdapat tiga skenario uji yang dapat dilihat pada Tabel 6.22.

Tabel 6.22 Skenario Uji Validasi Fungsional Ajukan Pendampingan Skripsi

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa disabilitas membuka halaman pendampingan skripsi, kemudian menekan tombol ajukan, kemudian mengisi informasi yang diperlukan dan menekan tombol ajukan pendampingan. Data masukan : - Judul Skripsi : Aplikasi Pencatatan Kesehatan Karyawan - Nama Pembimbing : Budi S.Pd - Waktu Mulai : 03/11/2020 - Waktu Selesai : 08/11/2020 - Dokumen Pendaftaran dokumen_skripsi.jpg	Pengajuan pendampingan skripsi berhasil disimpan ke <i>database</i> dan sistem menampilkan informasi "Pendampingan skripsi berhasil diajukan".	Pengajuan pendampingan skripsi berhasil disimpan ke <i>database</i> dan sistem menampilkan informasi "Pendampingan skripsi berhasil diajukan".	Valid.
2	Aktor mahasiswa disabilitas membuka	Sistem menampilkan	Sistem menampilkan	Valid.



No	Skenario	Expected	Actual	Status
	halaman pendampingan skripsi, kemudian menekan tombol ajukan, kemudian tidak mengisi informasi judul skripsi dan menekan tombol ajukan pendampingan. Data masukan : - Nama Pembimbing : Budi S.Pd - Waktu Mulai : 03/11/2020 - Waktu Selesai : 08/11/2020 - Dokumen Pendaftaran : dokumen_skripsi.jpg	informasi "Please fill out this field"	informasi "Please fill out this field"	
3	Aktor mahasiswa disabilitas telah mengajukan pendampingan skripsi, kemudian mencoba melakukan pengajuan kembali dengan menekan tombol ajukan pada halaman pendampingan skripsi.	Slstem menampilkan informasi "Maaf anda telah melakukan pengajuan sebelumnya. Silahkan tunggu konfirmasi"	Slstem menampilkan informasi "Maaf anda telah melakukan pengajuan sebelumnya. Silahkan tunggu konfirmasi".	Valid

6.2.17 Pengujian Validasi pada Fungsional Ajukan Pendampingan Tugas Akhir

Pengujian pada fungsional ajukan pendampingan tugas akhir dilakukan oleh aktor mahasiswa disabilitas dengan membuka halaman pendampingan tugas akhir, kemudian menekan tombol ajukan, kemudian mengisi informasi yang diperlukan dan menekan tombol ajukan pendampingan. Terdapat tiga skenario uji yang dapat dilihat pada Tabel 6.23.



Tabel 6.23 Skenario Uji Validasi Fungsional Ajukan Pendampingan Tugas Akhir

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa disabilitas membuka halaman pendampingan tugas akhir, kemudian menekan tombol ajukan, kemudian mengisi informasi yang diperlukan dan menekan tombol ajukan pendampingan. Data masukan : - Judul Tugas Akhir : Pembuatan Aplikasi Informasi Penyakit Berdasarkan Ciri Dengan Metode K-NN. - Waktu Mulai : 03/12/2020 - Waktu Selesai : 04/23/2020 - Dokumen Pendaftaran : dokumen_skripsi.jpg	Pengajuan pendampingan tugas akhir berhasil disimpan ke <i>database</i> dan sistem menampilkan informasi "Pendampingan tugas akhir berhasil diajukan".	Pengajuan pendampingan tugas akhir berhasil berhasil disimpan ke <i>database</i> dan sistem menampilkan informasi "Pendampingan tugas akhir berhasil diajukan".	Valid.
2	Aktor mahasiswa disabilitas membuka halaman pendampingan skripsi, kemudian menekan tombol ajukan, kemudian tidak mengisi informasi judul tugas akhir dan menekan tombol ajukan pendampingan. Data masukan : - Waktu Mulai : 03/12/2020	Sistem menampilkan informasi "<i>Please fill out this field</i>".	Sistem menampilkan informasi "<i>Please fill out this field</i>".	Valid.



No	Skenario	Expected	Actual	Status
	- Waktu Selesai : 04/23/2020 - Dokumen Pendaftaran dokumen_skripsi.jpg			
3	Aktor mahasiswa telah mengajukan pendampingan tugas akhir, kemudian melakukan pengajuan kembali dengan menekan tombol ajukan pada halaman pendampingan tugas akhir.	Sistem menampilkan informasi "Maaf anda telah melakukan pengajuan sebelumnya. Silahkan tunggu konfirmasi"	Sistem menampilkan informasi "Maaf anda telah melakukan pengajuan sebelumnya. Silahkan tunggu konfirmasi"	Valid.

6.2.18 Pengujian Validasi pada Fungsional Lihat Detail Pengajuan Pendampingan Perkuliahan

Pengujian pada fungsional lihat detail pengajuan pendampingan perkuliahan dilakukan oleh aktor admin. Skenario diawali dengan membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan perkuliahan. Kemudian admin menekan tombol selengkapnya. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.24.

Tabel 6.24 Skenario Uji Validasi Fungsional Lihat Detail Pengajuan Pendampingan Perkuliahan

No	Skenario	Expected	Actual	Status
1	Aktor admin membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan perkuliahan. Kemudian admin menekan tombol selengkapnya	Detail pengajuan pendampingan perkuliahan yang dipilih ditampilkan.	Detail pengajuan pendampingan perkuliahan yang dipilih ditampilkan.	Valid.



6.2.19 Pengujian Validasi pada Fungsional Lihat Detail Pengajuan Pendampingan Skripsi

Pengujian pada fungsional lihat detail pengajuan pendampingan skripsi dilakukan oleh aktor admin. Skenario diawali dengan membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan skripsi. Kemudian admin menekan tombol selengkapnya. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.25.

Tabel 6.25 Skenario Uji Validasi Fungsional Lihat Detail Pengajuan Pendampingan Skripsi

No	Skenario	Expected	Actual	Status
1	Aktor admin membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan skripsi. Kemudian admin menekan tombol selengkapnya	Detail pengajuan pendampingan skripsi yang dipilih ditampilkan.	Detail pengajuan pendampingan skripsi yang dipilih ditampilkan.	Valid.

6.2.20 Pengujian Validasi pada Fungsional Lihat Detail Pengajuan Pendampingan Tugas Akhir

Pengujian pada fungsional lihat detail pengajuan pendampingan tugas akhir dilakukan oleh aktor admin. Skenario diawali dengan membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan tugas akhir. Kemudian admin menekan tombol selengkapnya. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.26.

Tabel 6.26 Skenario Uji Validasi Fungsional Tambah Jadwal

No	Skenario	Expected	Actual	Status
1	Aktor admin membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan tugas. Kemudian admin menekan tombol selengkapnya	Detail pengajuan pendampingan tugas akhir yang dipilih ditampilkan.	Detail pengajuan pendampingan tugas akhir yang dipilih ditampilkan.	Valid.

6.2.21 Pengujian Validasi pada Fungsional Pilih Pendamping Skripsi

Pengujian pada fungsional pilih pendamping skripsi dilakukan oleh aktor admin. Skenario diawali dengan membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan skripsi. Kemudian admin menekan



tombol selengkapnya. Setelah detail pengajuan pendampingan skripsi ditampilkan, admin memilih pendamping pada kolom yang disediakan dan menekan tombol konfirmasi. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.27.

Tabel 6.27 Skenario Uji Validasi Fungsional Pilih Pendamping Skripsi

No	Skenario	<i>Expected</i>	<i>Actual</i>	Status
1	Aktor admin membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan skripsi. Kemudian admin menekan tombol selengkapnya. Setelah detail pengajuan ditampilkan, admin memilih pendamping pada kolom yang disediakan dan menekan tombol konfirmasi.	Pendamping skripsi terpilih dan sistem menampilkan informasi “Konfirmasi berhasil”.	Pendamping skripsi terpilih dan sistem menampilkan informasi “Konfirmasi berhasil”.	Valid.

6.2.22 Pengujian Validasi pada Fungsional Pilih Pendamping Tugas Akhir

Pengujian pada fungsional pilih pendamping tugas akhir dilakukan oleh aktor admin. Skenario diawali dengan membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan tugas akhir. Kemudian admin menekan tombol selengkapnya. Setelah detail pengajuan pendampingan tugas akhir ditampilkan, admin memilih pendamping pada kolom yang disediakan dan menekan tombol konfirmasi. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.28.

Tabel 6.28 Skenario Uji Validasi Fungsional Pilih Pendamping Tugas Akhir

No	Skenario	<i>Expected</i>	<i>Actual</i>	Status
1	Aktor admin membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan tugas akhir. Kemudian admin menekan tombol selengkapnya. Setelah detail pengajuan ditampilkan, admin memilih pendamping	Pendamping tugas akhir terpilih dan sistem menampilkan informasi “Konfirmasi berhasil”.	Pendamping tugas akhir terpilih dan sistem menampilkan informasi “Konfirmasi berhasil”.	Valid.



No	Skenario	Expected	Actual	Status
	pada kolom yang disediakan dan menekan tombol konfirmasi.			

6.2.23 Pengujian Validasi pada Fungsional Terima Pengajuan Pendampingan Perkuliahan

Pengujian pada fungsional terima pengajuan pendampingan perkuliahan dilakukan oleh aktor admin. Skenario diawali dengan membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan perkuliahan. Kemudian admin menekan tombol selengkapnya. Setelah detail pengajuan pendampingan perkuliahan ditampilkan, admin menerima pengajuan pendampingan perkuliahan dengan menekan tombol konfirmasi. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.29.

Tabel 6.29 Skenario Uji Validasi Fungsional Terima Pengajuan Pendampingan Perkuliahan

No	Skenario	Expected	Actual	Status
1	Aktor admin membuka halaman pengajuan pendampingan, kemudian membuka tab pendampingan perkuliahan. Kemudian admin menekan tombol selengkapnya. Setelah detail pengajuan ditampilkan, admin menerima pengajuan pendampingan perkuliahan dengan menekan tombol konfirmasi.	Pengajuan pendampingan perkuliahan diterima dan sistem menampilkan informasi "Anda berhasil melakukan konfirmasi".	Pengajuan pendampingan perkuliahan diterima dan sistem menampilkan informasi "Anda berhasil melakukan konfirmasi".	Valid.

6.2.24 Pengujian Validasi pada Fungsional Lihat Pelaporan

Pengujian pada fungsional lihat pelaporan dilakukan oleh aktor admin, mahasiswa disabilitas dan mahasiswa *volunteer*. Skenario diawali dengan membuka halaman pelaporan. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.30.

Tabel 6.30 Skenario Uji Validasi Fungsional Lihat Pelaporan.

No	Skenario	Expected	Actual	Status
1	Aktor melakukan login,	Daftar pelaporan	Daftar	Valid.



No	Skenario	Expected	Actual	Status
	kemudian dilanjutkan dengan membuka halaman pelaporan.	masuk ditampilkan.	pelaporan masuk ditampilkan.	

6.2.25 Pengujian Validasi pada Fungsional Lihat Detail Pelaporan

Pengujian pada fungsional lihat detail pelaporan dilakukan oleh aktor admin, mahasiswa disabilitas dan mahasiswa *voulnteer*. Skenario diawali dengan membuka halaman pelaporan. Setelah halaman pelaporan ditampilkan, aktor memilih pelaporan yang ingin dilihat secara detail dengan menekan judul laporan. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.31.

Tabel 6.31 Skenario Uji Validasi Fungsional Lihat Detail Pelaporan.

No	Skenario	Expected	Actual	Status
1	Aktor membuka halaman pelaporan. Setelah halaman pelaporan ditampilkan, aktor memilih pelaporan yang ingin dilihat secara detail dengan menekan judul laporan	Detail pelaporan yang dipilih ditampilkan.	Detail pelaporan yang dipilih ditampilkan.	Valid.

6.2.26 Pengujian Validasi pada Fungsional Balas Laporan

Pengujian pada fungsional balas laporan dilakukan oleh aktor admin, mahasiswa disabilitas dan mahasiswa *voulnteer*. Skenario diawali dengan membuka halaman pelaporan. Setelah halaman pelaporan ditampilkan, aktor memilih pelaporan yang ingin dilihat secara detail dengan menekan judul laporan. Setelah halaman detail laporan ditampilkan, aktor menulis balasan dengan mengisi balasan pada kolom balasan dan unggah dokumen pendukung. Terdapat dua skenario uji yang dapat dilihat pada Tabel 6.32.

Tabel 6.32 Skenario Uji Validasi Fungsional Balas Laporan

No	Skenario	Expected	Actual	Status
1	Aktor membuka halaman pelaporan. Setelah halaman pelaporan ditampilkan, aktor memilih pelaporan yang ingin dilihat secara detail dengan menekan judul laporan. Setelah halaman detail laporan	Balasan berhasil di simpan ke <i>database</i> dan sistem menampilkan informasi “respon berhasil dikirim”.	Balasan berhasil di simpan ke <i>database</i> dan sistem menampilkan informasi “respon berhasil dikirim”.	Valid.



No	Skenario	Expected	Actual	Status
	ditampilkan, aktor menulis balasan dengan mengisi balasan pada kolom balasan dan unggah dokumen pendukung. Data masukan : - Balasan : Kami akan segera memproses laporan anda. - Dokumen pendukung : dokumen.jpg			
2	Aktor membuka halaman pelaporan. Setelah halaman pelaporan ditampilkan, aktor memilih pelaporan yang ingin dilihat secara detail dengan menekan judul laporan. Setelah halaman detail laporan ditampilkan, aktor menulis balasan dengan tidak mengisi balasan pada kolom balasan dan unggah dokumen pendukung. Data masukan : - Dokumen pendukung : dokumen.jpg	Balasan gagal di simpan dan sistem menampilkan informasi <i>"Please fill out this field"</i>.	Balasan gagal di simpan dan sistem menampilkan informasi <i>"Please fill out this field"</i>.	Valid.

6.2.27 Pengujian Validasi pada Fungsional Tulis Pelaporan

Pengujian pada fungsional tulis pelaporan dilakukan oleh aktor mahasiswa disabilitas dan mahasiswa *voulnteer*. Skenario diawali dengan membuka halaman pelaporan. Setelah halaman pelaporan ditampilkan, aktor menulis laporan dengan mengisi judul pada kolom judul, isi laporan pada kolom laporan dan unggah dokumen pendukung. Terdapat dua skenario uji yang dapat dilihat pada Tabel 6.33.



Tabel 6.33 Skenario Uji Validasi Fungsional Tulis Laporan

No	Skenario	Expected	Actual	Status
1	Aktor membuka halaman pelaporan. Setelah halaman pelaporan ditampilkan, aktor menulis laporan dengan mengisi judul pada kolom judul, isi laporan pada kolom laporan dan unggah dokumen pendukung. Data masukan : - Judul : Saya mendapat permasalahan pada pendampingan saya. - Laporan Pendampingan yang saya jalankan pada hari selasa belum dikonfirmasi oleh mahasiswa disabilitas. Mohon bantuannya. - Dokumen pendukung dokumen.jpg	Laporan berhasil di simpan ke database dan sistem menampilkan informasi “Pelaporan berhasil diajukan”.	Laporan berhasil di simpan ke database dan sistem menampilkan informasi “Pelaporan berhasil diajukan”.	Valid.
2	Aktor membuka halaman pelaporan. Setelah halaman pelaporan ditampilkan, aktor menulis laporan dengan mengisi isi laporan pada kolom laporan dan unggah dokumen pendukung. Aktor tidak mengisi judul laporan. Data masukan :	Pelaporan gagal di simpan dan sistem menampilkan informasi “Please fill out this field”.	Pelaporan gagal di simpan dan sistem menampilkan informasi “Please fill out this field”.	Valid.



No	Skenario	Expected	Actual	Status
	- Laporan Pendampingan yang saya jalankan pada hari selasa belum dikonfirmasi oleh mahasiswa disabilitas. Mohon bantuannya.			
	- Dokumen pendukung : dokumen.jpg			

6.2.28 Pengujian Validasi pada Fungsional Lihat Detail Pendampingan Skripsi

Pengujian pada fungsional lihat detail pendampingan skripsi dilakukan oleh aktor mahasiswa disabilitas dan mahasiswa *volunteer*. Skenario diawali dengan membuka halaman pendampingan skripsi. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.34.

Tabel 6.34 Skenario Uji Validasi Fungsional Lihat Detail Pendampingan Skripsi.

No	Skenario	Expected	Actual	Status
1	Aktor membuka halaman pendampingan skripsi.	Detail pendampingan skripsi ditampilkan.	Detail pendampingan skripsi ditampilkan.	Valid.

6.2.29 Pengujian Validasi pada Fungsional Lihat Detail Pendampingan Tugas Akhir

Pengujian pada fungsional lihat detail pendampingan tugas akhir dilakukan oleh aktor mahasiswa disabilitas dan mahasiswa *volunteer*. Skenario diawali dengan membuka halaman pendampingan tugas akhir. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.35.

Tabel 6.35 Skenario Uji Validasi Fungsional Lihat Detail Pendampingan Tugas Akhir.

No	Skenario	Expected	Actual	Status
1	Aktor membuka halaman pendampingan tugas akhir.	Detail pendampingan tugas akhir.	Detail pendampingan tugas akhir.	Valid.



No	Skenario	Expected	Actual	Status
		ditampilkan.	ditampilkan.	

6.2.30 Pengujian Validasi pada Fungsional Lihat Jadwal

Pengujian pada fungsional lihat jadwal dilakukan oleh aktor mahasiswa disabilitas dan mahasiswa *volunteer*. Skenario pengujian diawali dengan membuka halaman jadwal. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.36.

Tabel 6.36 Skenario Uji Validasi Fungsional Lihat Jadwal

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa disabilitas dan mahasiswa <i>volunteer</i> membuka halaman jadwal.	Halaman jadwal ditampilkan.	Halaman jadwal ditampilkan.	Valid.

6.2.31 Pengujian Validasi pada Fungsional Lihat Detail Jadwal

Pengujian pada fungsional lihat detail jadwal dilakukan oleh aktor mahasiswa disabilitas dan mahasiswa *volunteer*. Skenario pengujian diawali dengan membuka halaman jadwal, kemudian klik tombol selengkapnya dan detail jadwal ditampilkan. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.37.

Tabel 6.37 Skenario Uji Validasi Fungsional Lihat Jadwal

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa disabilitas dan mahasiswa <i>volunteer</i> membuka halaman jadwal, kemudian menekan tombol selengkapnya pada jadwal yang dipilih.	Halaman detail jadwal ditampilkan.	Halaman detail jadwal ditampilkan.	Valid.

6.2.32 Pengujian Validasi pada Fungsional Tambah Jadwal

Pengujian pada fungsional tambah jadwal dilakukan oleh aktor mahasiswa disabilitas dan mahasiswa *volunteer* dengan membuka halaman jadwal, kemudian menekan tombol tambah jadwal, kemudian mengisi informasi yang diperlukan dan menekan tombol tambah jadwal. Terdapat dua skenario uji yang dapat dilihat pada Tabel 6.38.



Tabel 6.38 Skenario Uji Validasi Fungsional Tambah Jadwal

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa membuka halaman jadwal, kemudian menekan tombol tambah jadwal, kemudian mengisi informasi yang diperlukan dan menekan tombol tambah jadwal. Data masukan : - Keterangan Jadwal : Statistika - Hari : Rabu - Ruangan : A.2.1.8 - Jam Mulai : 06:45 AM - Jam Selesai : 08:00 AM	Jadwal mahasiswa berhasil disimpan di <i>database</i> dan sistem menampilkan informasi "Jadwal berhasil ditambahkan".	Jadwal mahasiswa berhasil disimpan di <i>database</i> dan sistem menampilkan informasi "Jadwal berhasil ditambahkan".	Valid.
2	Aktor mahasiswa membuka halaman jadwal, kemudian menekan tombol tambah jadwal, kemudian tidak mengisi keterangan dan menekan tombol tambah jadwal. Data masukan : - Hari : Rabu - Ruangan : A.2.1.8 - Jam Mulai : 06:45 AM - Jam Selesai : 08:00 AM	Sistem menampilkan informasi "<i>Please fill out this field</i>".	Sistem menampilkan informasi "<i>Please fill out this field</i>".	Valid.



6.2.33 Pengujian Validasi pada Fungsional Hapus Jadwal

Pengujian pada fungsional hapus jadwal dilakukan oleh aktor mahasiswa disabilitas dengan membuka halaman jadwal, kemudian menekan tombol selengkapnya dan menekan tombol hapus jadwal. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.39.

Tabel 6.39 Skenario Uji Validasi Fungsional Hapus Jadwal

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa disabilitas membuka halaman jadwal kemudian menekan tombol selengkapnya dan menekan tombol hapus jadwal.	Jadwal mahasiswa berhasil dihapus dari <i>database</i> dan sistem menampilkan informasi "Jadwal berhasil dihapus".	Jadwal mahasiswa berhasil dihapus dari <i>database</i> dan sistem menampilkan informasi "Jadwal berhasil dihapus".	Valid.

6.2.34 Pengujian Validasi pada Fungsional Lihat Jadwal Pendampingan

Pengujian pada fungsional lihat jadwal pendampingan dilakukan oleh aktor mahasiswa disabilitas dan mahasiswa *volunteer* dengan membuka halaman jadwal pendampingan. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.40.

Tabel 6.40 Skenario Uji Validasi Fungsional Lihat Jadwal Pendampingan

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa disabilitas membuka halaman jadwal pendampingan.	Halaman jadwal pendampingan ditampilkan.	Halaman jadwal pendampingan ditampilkan.	Valid.

6.2.35 Pengujian Validasi pada Fungsional Lihat Detail Jadwal Pendampingan

Pengujian pada fungsional lihat jadwal pendampingan dilakukan oleh aktor mahasiswa disabilitas dan mahasiswa *volunteer* dengan membuka halaman jadwal pendampingan, kemudian menekan tombol selengkapnya. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.41.

Tabel 6.41 Skenario Uji Validasi Fungsional Lihat Detail Jadwal Pendampingan

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa disabilitas membuka	Halaman jadwal pendampingan	Halaman jadwal pendampingan	Valid.



No	Skenario	Expected	Actual	Status
	halaman jadwal pendampingan, kemudian menekan tombol selengkapnya.	ditampilkan.	ditampilkan.	

6.2.36 Pengujian Validasi pada Fungsional Mulai Pendampingan

Pengujian pada fungsional mulai pendampingan dilakukan oleh aktor mahasiswa *volunteer* dengan membuka halaman jadwal pendampingan, kemudian menekan tombol selengkapnya, kemudian menekan tombol mulai pendampingan. Terdapat dua skenario uji yang dapat dilihat pada Tabel 6.42.

Tabel 6.42 Skenario Uji Validasi Fungsional Mulai Pendampingan

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa <i>volunteer</i> membuka halaman jadwal pendampingan, menekan tombol selengkapnya, menekan tombol mulai pendampingan dan melakukan <i>scan qr code</i> .	Pendampingan dimulai dan sistem menampilkan informasi "Pendampingan dimulai".	Pendampingan dimulai dan sistem menampilkan informasi "Pendampingan dimulai".	Valid.
2	Aktor mahasiswa <i>volunteer</i> membuka halaman jadwal pendampingan, menekan tombol selengkapnya, menekan tombol mulai pendampingan dan melakukan <i>scan qr code</i> pada <i>qr code</i> yang tidak sesuai.	Pendampingan gagal dimulai dan sistem menampilkan informasi "Anda tidak berhak memulai pendampingan ini".	Pendampingan gagal dimulai dan sistem menampilkan informasi "Anda tidak berhak memulai pendampingan ini".	Valid.

6.2.37 Pengujian Validasi pada Fungsional Tampilkan QR Code

Pengujian pada fungsional tampilkan *qr code* dilakukan oleh aktor mahasiswa disabilitas dengan membuka halaman jadwal pendampingan, kemudian menekan tombol selengkapnya, kemudian menekan tombol tampilkan *qr code*. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.43.

Tabel 6.43 Skenario Uji Validasi Fungsional Tampilkan QR Code

No	Skenario	Expected	Actual	Status
----	----------	----------	--------	--------



No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa membuka halaman pendampingan, menekan tombol selengkapnya, menekan tombol mulai tampilkan <i>qr code</i> .	Sistem menampilkan <i>qr code</i> pendampingan.	Sistem menampilkan <i>qr code</i> pendampingan.	Valid.

6.2.38 Pengujian Validasi pada Fungsional Isi *Logbook* Pendampingan Skripsi

Pengujian pada fungsional isi *logbook* pendampingan skripsi dilakukan oleh aktor mahasiswa *volunteer* dengan membuka halaman pendampingan skripsi, kemudian menekan tombol isi *logbook*, kemudian mengisi data *logbook* dan menekan tombol *upload*. Terdapat dua skenario uji yang dapat dilihat pada Tabel 6.44.

Tabel 6.44 Skenario Uji Validasi Fungsional Isi *Logbook* Pendampingan Skripsi

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa <i>volunteer</i> membuka halaman pendampingan skripsi, kemudian menekan tombol isi <i>logbook</i> , kemudian mengisi data <i>logbook</i> dan menekan tombol <i>upload</i> . Data masukan : Tanggal Pendampingan : 03/12/2020 - Jam Mulai : 01:00 PM - Jam Selesai : 02:00 PM Keterangan Pendampingan : penulisan bab 1 skripsi	<i>Logbook</i> disimpan ke <i>database</i> dan sistem menampilkan informasi “ <i>Logbook</i> berhasil diajukan”.	<i>Logbook</i> disimpan ke <i>database</i> dan sistem menampilkan informasi “ <i>Logbook</i> berhasil diajukan”.	Valid.



No	Skenario	Expected	Actual	Status
2	Aktor mahasiswa <i>volunteer</i> membuka halaman pendampingan skripsi, kemudian menekan tombol isi <i>logbook</i> , kemudian tidak mengisi data keterangan pada <i>logbook</i> dan menekan tombol upload. Data masukan : - Tanggal Pendampingan : 03/12/2020 - Jam Mulai : 01.00 PM - Jam Selesai : 02.00 PM	Sistem menampilkan informasi " <i>Please fill out this field</i> "	Sistem menampilkan informasi " <i>Please fill out this field</i> "	Valid.

6.2.39 Pengujian Validasi pada Fungsional Isi *Logbook* Pendampingan Tugas Akhir

Pengujian pada fungsional isi *logbook* pendampingan tugas akhir dilakukan oleh aktor mahasiswa *volunteer* dengan membuka halaman pendampingan tugas akhir, kemudian menekan tombol isi *logbook*, kemudian mengisi data *logbook* dan menekan tombol *upload*. Terdapat dua skenario uji yang dapat dilihat pada Tabel 6.45.

Tabel 6.45 Skenario Uji Validasi Fungsional Isi *Logbook* Pendampingan Tugas Akhir

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa <i>volunteer</i> membuka halaman pendampingan tugas akhir, kemudian menekan tombol isi <i>logbook</i> , kemudian mengisi data <i>logbook</i> dan menekan tombol upload. Data masukan : - Tanggal	<i>Logbook</i> disimpan ke <i>database</i> dan sistem menampilkan informasi " <i>Logbook</i> berhasil diajukan".	<i>Logbook</i> disimpan ke <i>database</i> dan sistem menampilkan informasi " <i>Logbook</i> berhasil diajukan".	Valid.



No	Skenario	Expected	Actual	Status
1	Pendampingan : 03/12/2020 - Jam Mulai : 01:00 PM - Jam Selesai : 02:00 PM - Keterangan : Pendampingan penulisan kesimpulan laporan			
2	Aktor mahasiswa <i>volunteer</i> membuka halaman pendampingan tugas akhir, kemudian menekan tombol isi <i>logbook</i> , kemudian tidak mengisi data keterangan pada <i>logbook</i> dan menekan tombol upload. Data masukan : - Tanggal Pendampingan : 03/12/2020 - Jam Mulai : 01:00 PM - Jam Selesai : 02:00 PM	Sistem menampilkan informasi " <i>Please fill out this field</i> "	Sistem menampilkan informasi " <i>Please fill out this field</i> "	Valid.

6.2.40 Pengujian Validasi pada Fungsional Konfirmasi *Logbook* Pendampingan Skripsi

Pengujian pada fungsional konfirmasi *logbook* pendampingan skripsi dilakukan oleh aktor mahasiswa disabilitas dengan membuka halaman pendampingan skripsi, kemudian menekan tombol konfirmasi pada *logbook* yang dipilih. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.46.

Tabel 6.46 Skenario Uji Validasi Fungsional Konfirmasi *Logbook* Pendampingan Skripsi

No	Skenario	Expected	Actual	Status
----	----------	----------	--------	--------



No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa disabilitas membuka halaman pendampingan skripsi, kemudian menekan tombol konfirmasi pada <i>logbook</i> yang dipilih.	<i>Logbook</i> berhasil dikonfirmasi dan sistem menampilkan informasi “ <i>Logbook</i> berhasil dikonfirmasi.”	<i>Logbook</i> berhasil dikonfirmasi dan sistem menampilkan informasi “ <i>Logbook</i> berhasil dikonfirmasi.”	Valid.

6.2.41 Pengujian Validasi pada Fungsional Konfirmasi *Logbook* Pendampingan Tugas Akhir

Pengujian pada fungsional konfirmasi *logbook* pendampingan tugas akhir dilakukan oleh aktor mahasiswa disabilitas dengan membuka halaman pendampingan tugas akhir, kemudian menekan tombol konfirmasi pada *logbook* yang dipilih. Terdapat satu skenario uji yang dapat dilihat pada Tabel 6.47.

Tabel 6.47 Skenario Uji Validasi Fungsional Konfirmasi *Logbook* Pendampingan Tugas Akhir

No	Skenario	Expected	Actual	Status
1	Aktor mahasiswa disabilitas membuka halaman pendampingan tugas akhir, kemudian menekan tombol konfirmasi pada <i>logbook</i> yang dipilih.	<i>Logbook</i> berhasil dikonfirmasi dan sistem menampilkan informasi “ <i>Logbook</i> berhasil dikonfirmasi.”	<i>Logbook</i> berhasil dikonfirmasi dan sistem menampilkan informasi “ <i>Logbook</i> berhasil dikonfirmasi.”	Valid.

6.3 Pengujian Kompatibilitas

Pengujian kompatibilitas adalah pengujian yang dilakukan untuk memastikan apakah perangkat lunak pendampingan mahasiswa disabilitas mampu dijalankan pada beberapa lingkungan. Dengan melakukan pengujian kompatibilitas, dapat diketahui seberapa baik suatu perangkat lunak dapat bekerja. Pada pengujian kompatibilitas pada perangkat lunak pendampingan mahasiswa disabilitas, diharapkan perangkat lunak dapat berjalan di beberapa peramban seperti Google Chrome, Mozilla Firefox dan peramban lainnya.

Pada pengujian kompatibilitas ini, digunakan sebuah alat bantu berupa aplikasi bernama Sortsite. Sortsite merupakan aplikasi yang mampu untuk melakukan pemindaian keseluruhan dari suatu *website* dan mampu melakukan penilaian terhadap kualitas dari suatu *website* yang telah dipindai. Dengan

menggunakan Sortsite maka pengujian kompatibilitas dapat dilakukan dengan memperhatikan beberapa parameter penilaian seperti yang terdapat pada Tabel 6.48.

Tabel 6.48 Parameter Kompatibilitas pada Sortsite.

No	Simbol	Kategori	Keterangan
1		<i>Minor Issues</i>	Simbol yang menandakan terdapat permasalahan yang disebabkan suatu peramban atau ketidaksesuaian <i>layout website</i> yang bersifat minor.
2		<i>Major Issues</i>	Simbol yang menandakan terdapat permasalahan yang disebabkan suatu peramban atau ketidaksesuaian <i>layout website</i> yang bersifat major.
3		<i>Critical Issues</i>	Simbol yang menandakan terdapat permasalahan pada suatu peramban yang menyebabkan hilangnya konten <i>website</i> .

Parameter yang dijabarkan pada Tabel 6.48 kemudian digunakan untuk merepresentasikan hasil pengujian kompatibilitas yang dilakukan aplikasi Sortsite terhadap perangkat lunak pendampingan mahasiswa disabilitas. Hasil pengujian kompatibilitas yang dilakukan oleh aplikasi Sortsite dapat dilihat dalam Gambar 6.4.

Errors

Accessibility

Compatibility

Search

Standards

Usability

This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS			Android		Key
Version	11	79	72	13	66	79	≤ 11	12	13	≤ 3	4*	
Critical Issues												Missing content or functionality
Major Issues												Major layout or performance problems
Minor Issues												Minor layout or performance problems

* Most Android devices from 4.4 onwards use Chrome as the default browser, older versions use the original Android stock browser

Gambar 6.4 Hasil Pengujian Kompatibilitas dari Perangkat Lunak Pendampingan Mahasiswa Disabilitas

Setelah dilakukan pengujian kompatibilitas terhadap perangkat lunak pendampingan mahasiswa disabilitas, didapatkan hasil tidak terdapat permasalahan – permasalahan yang termasuk dalam kategori *critical issue*. Akan tetapi diketahui terdapat beberapa permasalahan *major issue* dan permasalahan *minor issue* pada semua lingkungan uji yang disediakan. Permasalahan ini disebabkan adanya beberapa *Cascading Style Sheet* (CSS) yang tidak di dukung pada beberapa *browser* sehingga menyebabkan beberapa CSS tidak dapat digunakan secara maksimal di beberapa *browser*. Selain dari permasalahan yang tergolong dalam *major issue* dan *minor issue*, tidak terdapat permasalahan yang



tergolong dalam *critical issue* sehingga dapat disimpulkan bahwa seluruh fungsionalitas perangkat lunak pendampingan mahasiswa disabilitas dapat digunakan secara baik dalam semua lingkungan uji.



BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan proses pengembangan perangkat lunak pendampingan mahasiswa disabilitas, maka didapatkan kesimpulan mulai dari hasil analisis dan perancangan, implementasi serta pengujian dari perangkat lunak pendampingan mahasiswa disabilitas. Hasil dari pengembangan perangkat lunak pendampingan mahasiswa disabilitas dapat disimpulkan sebagai berikut.

1. Pada tahap analisis kebutuhan, didapatkan 37 kebutuhan fungsional dan 2 kebutuhan non fungsional dari perangkat lunak pendampingan mahasiswa disabilitas. Kebutuhan fungsional dan non fungsional ini didapatkan dari hasil wawancara kepada dua *stakeholder* pada Lembaga Pusat Studi dan Layanan Disabilitas Universitas Brawijaya. Dua *stakeholder* tersebut merupakan pihak yang berkaitan langsung dengan proses pendampingan mahasiswa disabilitas di lingkungan Universitas Brawijaya.

Terdapat 37 kebutuhan fungsional yang terdiri dari login, hapus jadwal pendampingan, ubah pendamping, generate jadwal, lihat pengajuan pendampingan skripsi, lihat pengajuan pendampingan tugas akhir, lihat jadwal mahasiswa, ubah jadwal mahasiswa, logout, ubah profil, ajukan pendampingan skripsi, ajukan pendampingan tugas akhir, tambah jadwal, hapus jadwal, lihat jadwal pendampingan, lihat detail jadwal pendampingan, mulai pendampingan, tampilan *qr code*, isi *logbook* pendampingan skripsi, isi *logbook* pendampingan tugas akhir, konfirmasi *logbook* pendampingan skripsi, konfirmasi *logbook* pendampingan tugas akhir. Selain itu terdapat tiga aktor pada perangkat lunak pendampingan mahasiswa disabilitas antara lain admin, mahasiswa disabilitas dan mahasiswa *volunteer*.

2. Pada proses perancangan, perangkat lunak pendampingan mahasiswa disabilitas dirancang dengan pendekatan perancangan berbasis objek (*object oriented design*). Perancangan dari perangkat lunak pendampingan mahasiswa dibagi ke dalam empat jenis perancangan yaitu *sequence diagram*, *class diagram*, *algoritme* dan antarmuka. Perancangan *sequence diagram* dilakukan terhadap 37 kebutuhan fungsional untuk mengetahui bagaimana interaksi objek serta alur dari fungsi yang dikembangkan. Dari hasil perancangan didapatkan *class diagram* sejumlah 24 kelas. Kemudian dilanjutkan dengan melakukan perancangan algoritme berupa *pseudocode* dari fungsi yang akan dikembangkan serta antarmuka untuk mengetahui bagaimana hasil akhir dari perangkat lunak yang akan dikembangkan.

Pada proses implementasi dari perangkat lunak pendampingan mahasiswa disabilitas, digunakan konsep pendekatan berbasis objek (*object oriented programming*) dalam proses pengembangannya. Implementasi dibagi ke dalam dua jenis yaitu implementasi *back-end* dan *front-end*. Pada implementasi *back-end*, digunakan bahasa pemrograman *Hypertext Preprocessor* (PHP) serta menggunakan *framework* yaitu *Laravel* 6.



Sedangkan pada implementasi *front-end* digunakan bahasa pemrograman berupa *Hypertext Markup Language* (HTML), *Cascading Style Sheet* (CSS), *Javascript* serta *framework Bootstrap* 4.

3. Pada tahap pengujian, dilakukan tiga jenis pengujian terhadap perangkat lunak pendampingan mahasiswa disabilitas yaitu pengujian *white box*, *black box* serta kompatibilitas. Pada pengujian *white box*, dilakukan pengujian terhadap tiga *method* yang terdapat pada perangkat lunak yaitu *generate_jadwal()*, *konfirmasi_pendampingan()* serta *upload_pengajuan_pendampingan_skripsi()* dimana mendapati hasil 100% pada pengujian *white box*. Pada pengujian *black box* dilakukan pengujian kepada 37 kebutuhan fungsional dimana didapatkan hasil 100%. Kemudian pada pengujian kompatibilitas dilakukan dengan bantuan aplikasi Sortsite. Pada pengujian kompatibilitas didapatkan hasil bahwa perangkat lunak pendampingan mahasiswa disabilitas mampu berjalan pada lingkungan yang berbeda serta keseluruhan fungsional dari perangkat lunak dapat dijalankan dengan baik pada seluruh lingkungan uji.

7.2 Saran

Terdapat beberapa saran yang diberikan untuk menunjang perangkat lunak pendampingan mahasiswa disabilitas. Saran tersebut antara lain :

1. Menambahkan fitur notifikasi dengan mengirimkan pesan menggunakan *Whatsapp Business API* sehingga setiap aktifitas yang dilakukan pada perangkat lunak pendampingan mahasiswa disabilitas dapat diketahui secara cepat oleh aktor terkait.
2. Integrasi antara perangkat lunak pendampingan mahasiswa dengan data mahasiswa pada sistem akademik mahasiswa (SIAM) Universitas Brawijaya sehingga dapat terhindar dari kekeliruan data mahasiswa.



LAMPIRAN A SURAT KELAYAKAN ETIK / IZIN



KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
UNIVERSITAS BRAWIJAYA

PUSAT STUDI DAN LAYANAN DISABILITAS (PSLD)

Jl. Veteran, Ketawanggede, Malang 65145, Indonesia

Telp : +62-341-551611

E-mail : psldbrawijaya@ub.ac.id ; psld.brawijaya@gmail.com

[http : www.psls.ub.ac.id](http://www.psls.ub.ac.id)

KETERANGAN KELAIKAN ETIK

Nomor : 171/UN10.C20.11/PN/2019

Dengan ini menyatakan bahwa:

Judul Penelitian : Pengembangan Perangkat Lunak Pendampingan Mahasiswa Disabilitas (Studi Kasus: Pusat Studi dan Layanan Disabilitas Universitas Brawijaya)
Nama Peneliti : Harun Arrasyid Hasibuan
Strata : S1 Teknik Informatika
Supervisor : Fajar Pradana, S.ST., M.Eng

Telah melalui proses penilaian kelayakan etik oleh Komisi Etik Kajian Disabilitas Pusat Studi dan Layanan Disabilitas (PSLD) Universitas Brawijaya pada tanggal 03 Oktober 2019, dinilai memenuhi persyaratan etika penelitian pada penyandang disabilitas, dan dinyatakan layak etik pada tanggal 30 Oktober 2019. Surat keterangan kelayakan etik ini berlaku 6 bulan sejak ditetapkan.

Peneliti diwajibkan untuk menginformasikan hal-hal sebagai berikut kepada Komisi Etik Kajian Disabilitas PSLD Universitas Brawijaya:

1. Seluruh perubahan pada penelitian dan alasan perubahannya, termasuk implikasi etis (jika ada)
2. Ketidakmampuan peneliti untuk melanjutkan penelitian
3. Penundaan penelitian lebih dari 6 bulan
4. Berakhirnya penelitian

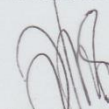
Peneliti diwajibkan untuk menyerahkan laporan penelitian setelah penelitian selesai dilaksanakan.

Ketua
Pusat Studi dan Layanan Disabilitas
Universitas Brawijaya



Zubaidah Ningsih AS., Ph.D
NIP. 197905242003122002

Malang, 30 Oktober 2019
Ketua Bidang Kajian
Pusat Studi dan Layanan Disabilitas
Universitas Brawijaya



Unita Werdi Rahaieng, M.Psi
NIK. 2013048103262001



LAMPIRAN B WAWANCARA

B.1 Wawancara Mekanisme Pendampingan

- Narasumber : Sulistyowatik, A.Md
- Tahun : 2019
- Diwawancara oleh : Harun Arrasyid Hasibuan
- Tanggal Wawancara : 04 November

Pertanyaan	Jawaban
Bagaimana mekanisme mahasiswa disabilitas mendaftar pendampingan?	Proses pendaftaran dibagi menjadi dua jalur. Pertama melalui jalur Seleksi Program Khusus Penyandang Disabilitas (SPKDP) serta melalui pengajuan. Mahasiswa yang mendaftar melalui pengajuan dapat mengisi form pengajuan yang telah disediakan. Mahasiswa dapat mengisi pengajuan pendampingan sesuai dengan kebutuhannya. Sedangkan mahasiswa yang melakukan pendaftaran melalui jalur SPKPD tidak perlu untuk melakukan pengajuan lagi dikarenakan PSLD telah menerima data mahasiswa baru jalur SPKPD dari universitas.
Tools/alat yang digunakan untuk melakukan penjadwalan pendampingan yang dilakukan oleh PSLD UB? Apakah terdapat kesulitan yang ditemui selama menggunakan alat tersebut?	Selama ini kita masih manual menggunakan <i>Microsoft excel</i> . Jadwal yang ada kita input, kemudian dilakukan pencocokan secara manual. Misal terdapat mahasiswa disabilitas yang perlu didampingi jam 8 – 9, maka akan di cari mahasiswa pendamping yang kosong sekitar jam 8 – 9 tersebut kemudian inputkan sebagai pendampingnya.
Berapa lama rata – rata waktu yang diperlukan pihak PSLD UB dalam menentukan jadwal pendampingan mahasiswa?	Waktu yang diperlukan bisa memakan waktu beberapa jam bergantung keadaan jadwal masing – masing mahasiswa. Pada awalnya penjadwalan pendampingan dilakukan dengan mencari pasangan pendamping yang memiliki jurusan yang sama dengan mahasiswa disabilitas, akan tetapi terjadi permasalahan dimana penjadwalan menjadi sulit dilakukan dikarenakan banyaknya jadwal yang bentrok apabila pendamping merupakan mahasiswa yang memiliki keminatan yang sama dengan mahasiswa disabilitas, sehingga pendampingan dilakukan dengan mencocokkan antara jadwal kosong mahasiswa pendamping dan jadwal perkuliahan dari mahasiswa disabilitas.



	Selain itu, sering terjadi perubahan pada jadwal pendampingan yang telah ditetapkan. Misal ketika mahasiswa disabilitas memiliki jadwal pendampingan jam 8 kemudian perkuliahannya dipindahkan oleh dosen yang bersangkutan ke jam 10, perlu dilakukan penjadwalan ulang untuk mencari mahasiswa yang kosong di waktu tersebut.
Apa mekanisme yang harus dilakukan apabila mahasiswa disabilitas/pendamping berhalangan hadir pada jadwal yang telah ditentukan ?	Jika terdapat mahasiswa yang berhalangan hadir, mahasiswa diwajibkan menghubungi saya melalui <i>whatsapp</i> untuk melakukan izin dan maksimal pengajuan izin tersebut adalah sore hari.
Apakah terdapat mekanisme pendampingan mahasiswa disabilitas dalam melaksanakan tugas akhir / skripsi ?	Terdapat mekanisme pendampingan khusus skripsi di PSKD UB. Akan tetapi terdapat perbedaan dengan pendampingan perkuliahan biasa dimana jika pendampingan perkuliahan ada kemungkinan pendamping berbeda – beda setiap jadwal pendampingannya, sedangkan pada pendampingan skripsi, mahasiswa pendamping tetap sama. Hal ini dikarenakan proses pengerjaan skripsi yang lama sehingga akan menyulitkan apabila pendamping berubah – ubah. Selain itu jadwal pendampingan ditentukan secara mandiri sesuai kesepakatan antara mahasiswa disabilitas dan pendamping.
Apakah kemampuan pendampingan mahasiswa juga berpengaruh dalam proses penjadwalan? Sebagai contoh mahasiswa dengan disabilitas tertentu hanya bisa didampingi oleh mahasiswa pendamping yang memiliki kemampuan untuk mendampingi mahasiswa tersebut.	Untuk kemampuan pendampingan mahasiswa sebenarnya jelas berpengaruh dalam proses pendampingan. Mahasiswa pendamping jelas harus memiliki kompetensi untuk melakukan pendampingan terhadap mahasiswa disabilitas untuk menunjang proses pendampingan dan komunikasi antara mahasiswa disabilitas dan pendamping. Akan tetapi di PSKD UB, setiap mahasiswa yang berstatus sebagai pendamping telah memiliki kompetensi untuk mendampingi setiap jenis disabilitas yang ada di Universitas Brawijaya. Hal ini dikarenakan sebelum ditetapkan sebagai pendamping, mahasiswa telah dilakukan pelatihan selama 2 hari untuk memastikan mahasiswa memiliki kompetensi untuk melakukan pendampingan pada setiap mahasiswa disabilitas. Sebagai contoh mahasiswa pendamping harus bisa menggunakan bahasa isyarat untuk mendampingi mahasiswa tuna rungu /



	wicara. Atau ketika mendampingi tuna daksa yang memerlukan kursi roda, mahasiswa pendamping harus mengetahui prosedur bagaimana cara mendorong kursi roda yang benar dan sebagainya.
Bagaimana melakukan konfirmasi terhadap proses pendampingan yang telah berjalan?	Proses pendampingan dilakukan oleh mahasiswa pendamping sesuai dengan jadwal yang telah ditentukan. Setelah pendampingan selesai dilaksanakan, mahasiswa pendamping diwajibkan untuk mengisi <i>logbook</i> pendampingan sebagai bukti telah melaksanakan pendampingan dan mahasiswa disabilitas memberikan tanda tangan pada <i>logbook</i> pendampingan yang telah di isi oleh pendamping.
Tambahan: Pada proses koordinasi pendampingan, kesulitan biasanya terjadi pada proses penjadwalan pendampingan. Hal ini terjadi dikarenakan jadwal mungkin sekali berubah bergantung keadaan dari masing – masing mahasiswa baik itu mahasiswa disabilitas maupun pendamping. Perubahan yang mendadak ini mengakibatkan perlu dilakukannya penyesuaian kembali pada jadwal pendampingan yang ada dan hal ini biasanya terjadi setiap harinya. Ketika dilakukannya perubahan, hasil perubahan akan dikirimkan kembali ke grup <i>whatsapp</i> yang beranggotakan mahasiswa disabilitas dan pendamping sehingga mahasiswa disabilitas maupun pendamping harus melakukan penyesuaian kembali berdasarkan jadwal pendampingan tersebut.	

B.2 Wawancara Mekanisme Pendampingan Skripsi, Tugas Akhir Serta Pelaporan

- Narasumber : Rachmawati Ayu Kuswoyo, S.Pd
- Tahun : 2019
- Diwawancara oleh : Harun Arrasyid Hasibuan
- Tanggal Wawancara : 23 Desember

Pertanyaan	Jawaban
Bagaimana mekanisme penjadwalan yang dilakukan untuk pendampingan skripsi dan tugas akhir ?	Pendampingan skripsi dan tugas akhir dilakukan atas kesepakatan antara mahasiswa disabilitas dan pendamping. PSKD UB tidak melakukan penjadwalan kepada pendampingan pendampingan ini dikarenakan sifat dari pendampingan yang menyesuaikan dengan progres pengerjaan tugas yang dikerjakan mahasiswa disabilitas.
Bagaimana mekanisme pelaksanaan skripsi? Apakah pendamping harus memiliki pengalaman mengerjakan	Pendampingan skripsi dilakukan oleh mahasiswa pendamping untuk memudahkan mahasiswa disabilitas terutama dalam pengolahan dokumen skripsi serta hal – hal lain yang sifatnya menunjang. Pendampingan dalam pengolahan dokumen skripsi diperlukan terutama terhadap mahasiswa yang



skripsi sebelumnya?	berstatus sebagai tuna netra karena biasanya permasalahan yang terjadi pada mahasiswa tuna netra adalah pengetikan kalimat yang sering terbalik-balik. Selain itu beberapa mahasiswa disabilitas lain juga memerlukan pendampingan seperti ketika berkonsultasi dengan pembimbing dan narasumbernya.
Bagaimana pihak PSLD UB mengetahui perkembangan dari pendampingan tugas akhir maupun skripsi?	Mahasiswa pendamping diwajibkan mengisi <i>logbook</i> pendampingan. Dari <i>logbook</i> pendampingan itulah kita mengetahui bagaimana perkembangan dari tugas akhir maupun skripsi mahasiswa disabilitas pada saat pendampingan. Selain itu kita mewajibkan pendampingan skripsi dilakukan di lingkungan PSLD UB agar kita bisa mengawasi secara langsung serta membantu proses skripsi maupun tugas akhir dari mahasiswa disabilitas. Selain itu biasanya mahasiswa disabilitas juga menghubungi saya secara pribadi apabila terdapat kendala dalam pengerjaan tugasnya maupun dalam proses pendampingannya.
Untuk keluhan / saran yang diberikan mahasiswa disabilitas maupun pendamping selama ini berarti disampaikan ke Ibu?	Selama ini keluhan maupun saran yang diberikan mahasiswa disabilitas maupun pendamping memang disampaikan ke saya secara langsung maupun menghubungi saya lewat <i>whatsapp</i> . Keluhan maupun saran tersebut saya catat di aplikasi <i>Microsoft excel</i> sebagai bahan evaluasi kedepannya.
Apakah mahasiswa pendamping tugas akhir / skripsi juga melakukan pendampingan perkuliahan? Jika iya bagaimana cara untuk menghindari jadwal yang bentrok mengingat jadwal pendampingan skripsi / tugas akhir berdasarkan kesepakatan antara mahasiswa disabilitas dan pendamping?	Khusus untuk mahasiswa pendamping yang mendampingi pengerjaan skripsi maupun tugas akhir tidak dimasukan dalam penjadwalan pendampingan perkuliahan. Hal ini dilakukan agar proses pendampingan bisa lebih maksimal terutama pendampingan skripsi yang bahkan bisa dilaksanakan hampir setiap harinya.
Tambahan: Pada sesi wawancara ini, Ibu Rachmawati memberikan saran untuk menambahkan fitur pelaporan untuk memberikan sarana bagi mahasiswa disabilitas dan pendamping dalam menyampaikan kritik maupun saran. Saran pengembangan ini diberikan karena selama ini PSLD belum memiliki sarana untuk penyampaian pelaporan selain melalui <i>whatsapp</i> pribadi dari Ibu	



Rachmawati. Mekanisme pelaporan ini diharapkan mampu membantu mahasiswa disabilitas dan pendamping dalam mengajukan pelaporan serta mengetahui perkembangan pelaporan yang dilakukannya.

Pada sesi wawancara ini juga diberikan saran untuk menambahkan fitur cek lokasi secara otomatis dari pengajuan *logbook* pendampingan tugas akhir / skripsi karena sifat pendampingan tugas akhir / skripsi yang diwajibkan untuk dilaksanakan di lingkungan PSLD.



LAMPIRAN C FORMULIR PELAKSANAAN PENDAMPINGAN MAHASISWA DISABILITAS

C.1 Formulir Pengajuan Pendampingan/Tutor

FORM PENGAJUAN PENDAMPINGAN

I. Identitas

Nama :

Alamat :

.....(*alamat di Malang)

Email :

No.Hp :

Agama :

Fakultas :

Jurusan :

Semester :

II. Keterangan Disabilitas

Jenis disabilitas :

Alat bantu yang dibutuhkan :

Kesulitan yang dihadapi :

III. Kebutuhan Pendampingan

☐ Kuliah

Keterangan:

☐ Penyusunan Tugas Akhir/Laporan/Skripsi (*coret yang tidak perlu)

Keterangan:

☐ PPL/KKN/PKN (*coret yang tidak perlu)

Keterangan:

☐ Pengurusan beasiswa

Keterangan:

☐ Pengurusan administrasi akademik

Keterangan:

☐ Praktikum

Keterangan:

☐ OSPEK

Keterangan:

☐ Lainnya



Dokumen yang perlu dilampirkan:

1. Surat keterangan disabilitas
2. Surat rekomendasi dosen PA (Pembimbing Akademik)
3. Transkrip nilai

Seluruh informasi yang tercantum dalam formulir ini adalah benar sehingga dapat dipertanggungjawabkan dan dipergunakan sebagaimana mestinya.

Malang,

(TTD dan nama terang)



LAMPIRAN D PENGUJIAN UNIT MENGGUNAKAN LIBRARY PHPUNIT

D.1 Pengujian Unit Method `generate_jadwal()` Jalur 1

```
PS D:\Code-Project\diments> ./vendor/bin/phpunit --filter test_case_1 tests/Unit/GenerateJadwalTest.php --debug
PHPUnit 8.4.3 by Sebastian Bergmann and contributors.

Test 'Tests\Unit\GenerateJadwalTest::test_case_1' started
Test 'Tests\Unit\GenerateJadwalTest::test_case_1' ended

Time: 9.59 seconds, Memory: 16.00 MB

OK (1 test, 1 assertion)
```

D.2 Pengujian Unit Method `generate_jadwal()` Jalur 2

```
PS D:\Code-Project\diments> ./vendor/bin/phpunit --filter test_case_2 tests/Unit/GenerateJadwalTest.php --debug
PHPUnit 8.4.3 by Sebastian Bergmann and contributors.

Test 'Tests\Unit\GenerateJadwalTest::test_case_2' started
Test 'Tests\Unit\GenerateJadwalTest::test_case_2' ended

Time: 828 ms, Memory: 16.00 MB

OK (1 test, 1 assertion)
```

D.3 Pengujian Unit Method `konfirmasi_pendampingan()` Jalur 1

```
PS D:\Code-Project\diments> ./vendor/bin/phpunit --filter test_case_1 tests/Unit/KonfirmasiPendampinganTest.php --debug
PHPUnit 8.4.3 by Sebastian Bergmann and contributors.

Test 'Tests\Unit\KonfirmasiPendampinganTest::test_case_1' started
Test 'Tests\Unit\KonfirmasiPendampinganTest::test_case_1' ended

Time: 486 ms, Memory: 16.00 MB

OK (1 test, 1 assertion)
PS D:\Code-Project\diments>
```



D.4 Pengujian Unit Method konfirmasi_pendampingan() Jalur 2

```
PS D:\Code-Project\diments> ./vendor/bin/phpunit --filter test_case_2
tests/Unit/KonfirmasiPendampinganTest.php --debug
PHPUnit 8.4.3 by Sebastian Bergmann and contributors.

Test 'Tests\Unit\KonfirmasiPendampinganTest::test_case_2' started
Test 'Tests\Unit\KonfirmasiPendampinganTest::test_case_2' ended

Time: 547 ms, Memory: 16.00 MB

OK (1 test, 1 assertion)
PS D:\Code-Project\diments>
```

D.5 Pengujian Unit Method konfirmasi_pendampingan() Jalur 3

```
PS D:\Code-Project\diments> ./vendor/bin/phpunit --filter test_case_3
tests/Unit/KonfirmasiPendampinganTest.php --debug
PHPUnit 8.4.3 by Sebastian Bergmann and contributors.

Test 'Tests\Unit\KonfirmasiPendampinganTest::test_case_3' started
Test 'Tests\Unit\KonfirmasiPendampinganTest::test_case_3' ended

Time: 599 ms, Memory: 16.00 MB

OK (1 test, 1 assertion)
PS D:\Code-Project\diments>
```



D.6 Pengujian Unit Method

upload_pengajuan_pendampingan_skripsi() Jalur 1

```
PS D:\Code-Project\diments> ./vendor/bin/phpunit --filter test_case_1
tests/Unit/UploadPengajuanPendampinganSkripsiTest.php --debug
PHPUnit 8.4.3 by Sebastian Bergmann and contributors.

Test 'Tests\Unit\UploadPengajuanPendampinganSkripsiTest::test_case_1'
started
Test 'Tests\Unit\UploadPengajuanPendampinganSkripsiTest::test_case_1'
ended

Time: 637 ms, Memory: 16.00 MB

OK (1 test, 1 assertion)
```

D.7 Pengujian Unit Method

upload_pengajuan_pendampingan_skripsi() Jalur 2

```
PS D:\Code-Project\diments> ./vendor/bin/phpunit --filter test_case_2
tests/Unit/UploadPengajuanPendampinganSkripsiTest.php --debug
PHPUnit 8.4.3 by Sebastian Bergmann and contributors.

Test 'Tests\Unit\UploadPengajuanPendampinganSkripsiTest::test_case_2'
started
Test 'Tests\Unit\UploadPengajuanPendampinganSkripsiTest::test_case_2'
ended

Time: 789 ms, Memory: 16.00 MB

OK (1 test, 1 assertion)
```

D.8 Pengujian Unit Method

upload_pengajuan_pendampingan_skripsi() Jalur 3

```
PS D:\Code-Project\diments> ./vendor/bin/phpunit --filter test_case_3
tests/Unit/UploadPengajuanPendampinganSkripsiTest.php --debug
PHPUnit 8.4.3 by Sebastian Bergmann and contributors.

Test 'Tests\Unit\UploadPengajuanPendampinganSkripsiTest::test_case_3'
started
Test 'Tests\Unit\UploadPengajuanPendampinganSkripsiTest::test_case_3'
ended

Time: 734 ms, Memory: 16.00 MB

OK (1 test, 1 assertion)
```



DAFTAR REFERENSI

- Anas, H., Ilyas, M., Tariq, Q., & Hummayun, M. (2016). Requirements Validation Techniques: An Empirical Study. *International Journal of Computer Applications*, 148(14), 5–10. <https://doi.org/10.5120/ijca2016910911>
- Anggito, A., & Johan, S. (2018). *Metodologi Penelitian Kualitatif*. Sukabumi: CV Jejak (Jejak Publisher).
- Azis, M. F. (2005). *Object Oriented Programming dengan PHP5*. <https://doi.org/979-20-6469-6>
- Bergmann, S. (2005). *PHPUnit Pocket Guide: Test-Driven Development in PHP* (1st ed.). Sebastopol.
- DESAL, S., & SRIVASTAVA, A. (2016). *SOFTWARE TESTING : A Practical Approach* (Second Edi). Delhi: PHI Learning Pvt. Ltd.
- Giurgiu, L., & Gligorea, I. (2017). Responsive Web Design Techniques. *International Conference KNOWLEDGE-BASED ORGANIZATION*, 23(3), 37–42. <https://doi.org/10.1515/kbo-2017-0153>
- Handayani, F. S., & Putri, M. P. (2018). Implementasi Metode Prototipe Pada Website Penelusuran Minat Peserta Didik Sebagai Layanan Bimbingan Konseling. *CSRID (Computer Science Research and Its Development Journal)*, 10(1), 44. <https://doi.org/10.22303/csrid.10.1.2018.44-57>
- Howe, S. (2014). *Learn to Code HTML and CSS: Develop and Style Websites*. New Riders.
- Jakobus, B., & Marah, J. (2016). *Mastering Bootstrap 4*. Birmingham.
- Naik, K. (2008). *SOFTWARE TESTING AND QUALITY ASSURANCE Theory and Practice*. Canada: John Wiley & Sons, Inc.
- Nixon, R. (2012). *Learning PHP, MySQL, JavaScript, & CSS* (Second Edi). O'Reilly Media, Inc.
- PowerMapper. (2019). SortSite 5 User Manual. Retrieved May 2, 2020, from <https://www.powermapper.com/products/sortsites/>
- Rumbaugh, J., Jacobson, I., & Booch, G. (2013). The Unified Modeling Language Reference Manual. In *Journal of Chemical Information and Modeling* (Vol. 53). <https://doi.org/10.1017/CBO9781107415324.004>
- Sejarah PSLD. (2013). Retrieved June 20, 2020, from <http://psld.ub.ac.id/in/about/sejarah-psld/>
- Sekretariat Kementrian Negara. (2015). *Undang - Undang No. 8 Tahun 2016 Tentang Penyandang Disabilitas*. (243), 1–5. <https://doi.org/10.1017/CBO9781107415324.004>
- Sitanggang, A. S. (2016). Pemodelan Rancangan Proses Penjadwalan Mata Kuliah Di International Program Kedalam Sistem Informasi Unikom Berbasis



Android. *Jurnal Teknologi Dan Informasi (JATI)*, 1(12), 57–66. Retrieved from <https://ojs.unikom.ac.id/in%0Adex.php/jati/issue/view/74>

Sommerville, I. (2011). *SOFTWARE ENGINEERING* (Ninth Edit). Addison-Wesley.

Song, I. (2013). *Developing Sequence Diagrams in UML*. (September 2001). <https://doi.org/10.1007/3-540-45581-7>

Stauffer, M. (2017). *Laravel: Up and Running*. O'Reilly Media, Inc.

Valade, J. (2004). *PHP and MySQL for Dummies, Second Edition*. Retrieved from <papers2://publication/uuid/E8B140A9-150B-491C-BE59-768FF4AFDE12>