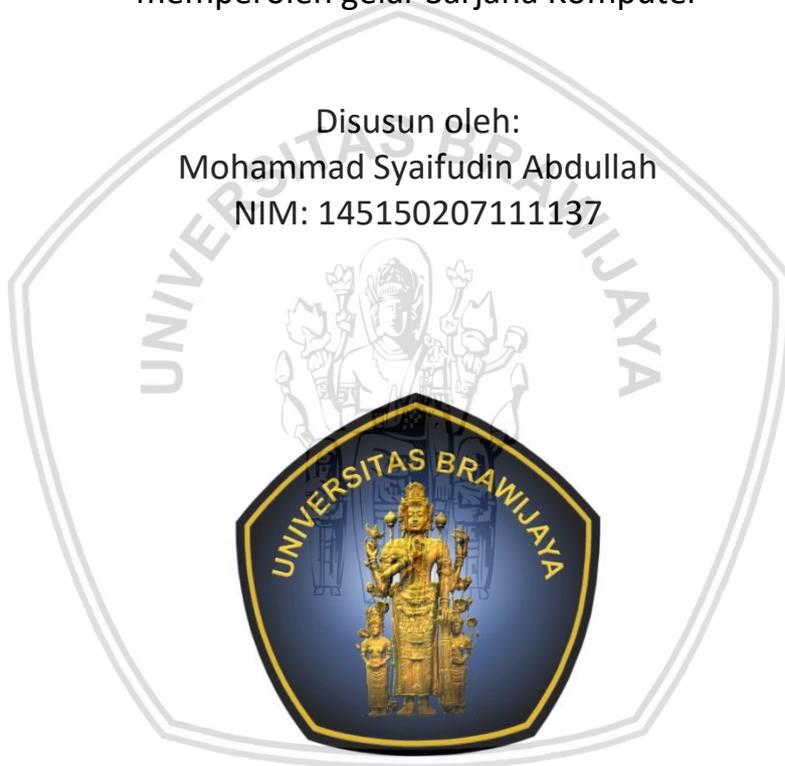


**PENGEMBANGAN PERANGKAT LUNAK V-APPS SEBAGAI  
APLIKASI UNTUK MENGINFORMASIKAN JADWAL DAMPING  
MAHASISWA DIFABEL (STUDI KASUS PUSAT STUDI DAN  
LAYANAN DISABILITAS UNIVERSITAS BRAWIJAYA)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Mohammad Syaifudin Abdullah  
NIM: 145150207111137



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018**

# PENGESAHAN

PENGEMBANGAN PERANGKAT LUNAK V-APPS SEBAGAI APLIKASI UNTUK  
MENGINFORMASIKAN JADWAL DAMPING MAHASISWA DIFABEL (STUDI KASUS  
PUSAT STUDI DAN LAYANAN DISABILITAS UNIVERSITAS BRAWIJAYA)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Mohammad Syaifudin Abdullah  
NIM: 145150207111137

Skripsi ini telah diuji dan dinyatakan lulus pada  
10 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Adam Hendra Brata, S.Kom., M.T., M.Sc.  
NIK: 2016079001051001

Komang Candra Brata, S.Kom., M.T., M.Sc.  
NIK: 2016078907111001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurmiawan, S.T, M.T, Ph.D

NIP : 19710518 200312 1 001 



## PERNYATAAN ORISINALITAS

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

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, 10 Juli 2018



Mohammad Syaifudin A

NIM: 145150207111137

## KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “PENGEMBANGAN PERANGKAT LUNAK V-APPS SEBAGAI APLIKASI UNTUK MENGINFORMASIKAN JADWAL DAMPING MAHASISWA DIFABEL (STUDI KASUS PUSAT STUDI DAN LAYANAN DISABILITAS UNIVERSITAS BRAWIJAYA)”.

Untuk kesempatan ini penulis juga menyampaikan rasa terimakasih kepada pihak-pihak yang telah membantu penulis selama penyusunan skripsi, diantaranya:

1. Allah SWT yang telah memberi kemudahan dalam semua proses penulisan skripsi ini.
2. Kepada kedua orang tua penulis, yaitu Bapak Timang Soewardoyo dan Ibu Sri Indrawati beserta keluarga besar yang selalu memberikan segala masukan, do’a, motivasi dan semangat yang tidak terputus.
3. Bapak Adam Hendra Brata, S.Kom., M.T., M.Sc. selaku dosen pembimbing I yang telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
4. Bapak Komang Candra Brata, S.Kom., M.T., M.Sc. selaku dosen pembimbing II yang juga telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, Bapak Ir. Heru Nuswasito, M.Kom, Bapak Suprpto, S.T., M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
6. 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.
7. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brwaijaya atas kesediaannya dalam mengajarkan dan membagikan ilmu yang bermanfaat bagi penulis.
8. Bapak Hermawan Dwi Putra, S.H, Bapak Prasetyo Iskandar, S.T, Bapak Andito Aryo D.P, S.Psi, dan Bapak Dhika Wirya Dipraja, S.T yang telah memberikan banyak bantuan dan kemudahan-kemudahan administrasi.
9. Rekan-rekan Tahu Telor Community (TTC) yang membantu memberikan ilmu dalam implementasi dari penelitian ini.
10. Teman-teman L-tifam yang telah membantu memberikan ilmu dalam mengerjakan skripsi ini.

11. Semua Pihak yang tidak semuanya bisa dituliskan disini yang terlibat secara langsung maupun tidak langsung dalam proses pengerjaan skripsi ini maupun sebagai pemberi semangat dan motivasi.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini baik dalam teknik penyajian materi maupun pembahasan. Demi kesempurnaan penelitian skripsi ini, saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga karya tulis ini bermanfaat dan dapat memberikan sumbangan yang berarti bagi pihak yang membutuhkan.

Malang, 10 Juli 2018

Penulis  
abdullahsyarifudin@gmail.com



## ABSTRAK

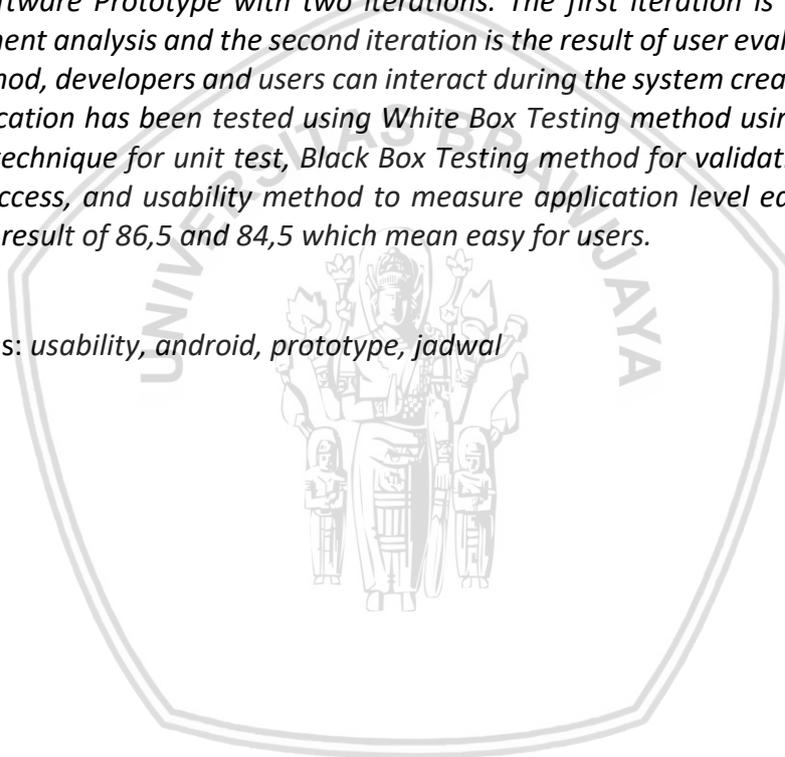
Pusat Studi dan Layanan Disabilitas Universitas Brawijaya adalah lembaga yang memiliki fungsi yaitu sebagai pusat pelayanan bagi penyandang disabilitas. Salah satu pelayanan yang diberikan kepada mahasiswa disabilitas adalah memberikan pendampingan pada saat perkuliahan. Untuk menginformasikan jadwal damping masih menggunakan cara manual yaitu menghubungi mahasiswa secara personal atau di tempel di papan pengumuman. Oleh karena itu, dibuat aplikasi V-Apps yang dapat menginformasikan jadwal damping. Aplikasi juga dapat melakukan konfirmasi kehadiran pendamping dan memberi *review* kepada pendamping yang sudah melakukan konfirmasi kehadiran. Aplikasi ini dibuat dengan menggunakan Android dan data yang digunakan adalah data dari sistem penjadwalan jadwal damping PSLD UB. Pengembangan menggunakan *Software Development Life Cycle Software Prototype* dengan dua kali iterasi. Iterasi pertama hasil dari analisis kebutuhan dan iterasi kedua adalah hasil dari evaluasi pengguna. Dengan metode ini pengembang dan calon pengguna dapat berinteraksi selama proses pembuatan sistem. Aplikasi telah diuji menggunakan metode *White Box Testing* menggunakan teknik *Basis Path Testing* untuk pengujian unit, metode *Black Box Testing* untuk pengujian validasi dengan keberhasilan 100% dan metode *usability* untuk mengukur tingkat kemudahan aplikasi kepada pengguna dengan hasil 86,5 dan 84,5 yang berarti mudah untuk pengguna.

Kata kunci: *usability, android, prototype, jadwal*

## ABSTRACT

*Disability Study and Service Center Brawijaya University is an institution that has a function that is as a service center for students with disabilities. One of the services given to students with a disability is to provide assistance during lectures. To inform the schedule is still using the manual way of contacting students or posted on the bulletin board. Therefore, a V-Apps application is created which can inform the damping schedule. Applications can also confirm the presence of a companion and provide a review to the companion who has confirmed the presence. This app is made using Android and the data used is data from schedule scheduling system from PSLD UB. Development using Software Development Life Cycle Software Prototype with two iterations. The first iteration is the result of requirement analysis and the second iteration is the result of user evaluation. With this method, developers and users can interact during the system creation process. An application has been tested using White Box Testing method using Base Path Testing technique for unit test, Black Box Testing method for validation test with 100% success, and usability method to measure application level ease to a user with the result of 86,5 and 84,5 which mean easy for users.*

Keywords: *usability, android, prototype, jadwal*



## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN .....	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan .....	3
1.4 Manfaat .....	4
1.5 Batasan masalah .....	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Dasar Teori.....	6
2.1.1 Pusat Studi dan Layanan Disabilitas Universitas Brawijaya.....	6
2.1.2 Pendamping .....	6
2.1.3 Mahasiswa Difabel.....	7
2.1.4 Rekayasa Perangkat Lunak.....	8
2.1.5 <i>Software Development Life Cycle (SDLC)</i> .....	8
2.1.6 <i>JavaScript Object Notation (JSON)</i> .....	11
2.1.7 <i>Entity Relationship Diagram</i> .....	13
2.1.8 Android .....	16
2.1.9 Android SDK .....	17
2.1.10 <i>Retrofit</i> .....	17
2.1.11 <i>QR Code ZXing</i> .....	17
2.1.12 <i>Unified Modeling Language (UML)</i> .....	18



2.1.13 Pengujian Perangkat Lunak.....	25
<b>BAB 3 METODOLOGI .....</b>	<b>29</b>
3.1 Studi Pustaka .....	30
3.2 Pengumpulan Data .....	30
3.3 Analisis Kebutuhan .....	31
3.4 Perancangan Sistem .....	31
3.5 Pembuatan <i>Prototype</i> .....	32
3.6 Evaluasi Pengguna .....	32
3.7 Implementasi Sistem .....	33
3.8 Pengujian dan Analisis .....	33
3.9 Pengambilan Kesimpulan dan Saran .....	34
<b>BAB 4 ANALISIS KEBUTUHAN .....</b>	<b>35</b>
4.1 Gambaran Umum Sistem.....	35
4.2 Identifikasi Aktor.....	35
4.3 Identifikasi Kebutuhan.....	36
4.3.1 Kebutuhan Fungsional .....	36
4.3.2 Kebutuhan Non Fungsional.....	40
4.4 Analisis Data.....	40
4.5 Pemodelan Kebutuhan .....	41
4.5.1 <i>Use Case Diagram</i> .....	41
4.5.2 <i>Use case Scenario</i> .....	42
<b>BAB 5 PERANCANGAN DAN IMPLEMENTASI .....</b>	<b>56</b>
5.1 Perancangan .....	56
5.1.1 Perancangan Arsitektur Sistem.....	56
5.1.2 Perancangan <i>Class Diagram</i> .....	57
5.1.3 <i>Sequence Diagram</i> .....	62
5.1.4 Perancangan Komponen.....	65
5.1.5 Perancangan Data .....	67
5.1.6 Perancangan Antarmuka .....	75
5.2 Implementasi .....	83
5.2.1 Spesifikasi Sistem .....	83
5.2.2 Implementasi Basis Data.....	84



5.2.3 Implentasi Kode Program .....	89
5.2.4 Implementasi Antarmuka .....	92
BAB 6 PENGUJIAN .....	97
6.1 Pengujian Fungsional .....	97
6.1.1 Pengujian Unit.....	97
6.1.2 Pengujian Validasi .....	105
6.2 Pengujian Non Fungsional .....	130
6.2.1 Pengujian Usabilitas .....	130
6.3 Analisis Hasil Pengujian .....	133
6.3.1 Pengujian Unit.....	133
6.3.2 Pengujian Validasi .....	134
6.3.3 Pengujian Usabilitas.....	134
BAB 7 Penutup .....	135
7.1 Kesimpulan .....	135
7.2 Saran .....	136
DAFTAR PUSTAKA.....	137
LAMPIRAN A TRANSKRIP WAWANCARA 1 .....	139
LAMPIRAN B TRANSKRIP WAWANCARA 2 .....	141
LAMPIRAN C TRANSKRIP WAWANCARA 3 .....	143
LAMPIRAN D KUISIONER .....	145
LAMPIRAN E EVALUASI PENGGUNA .....	146

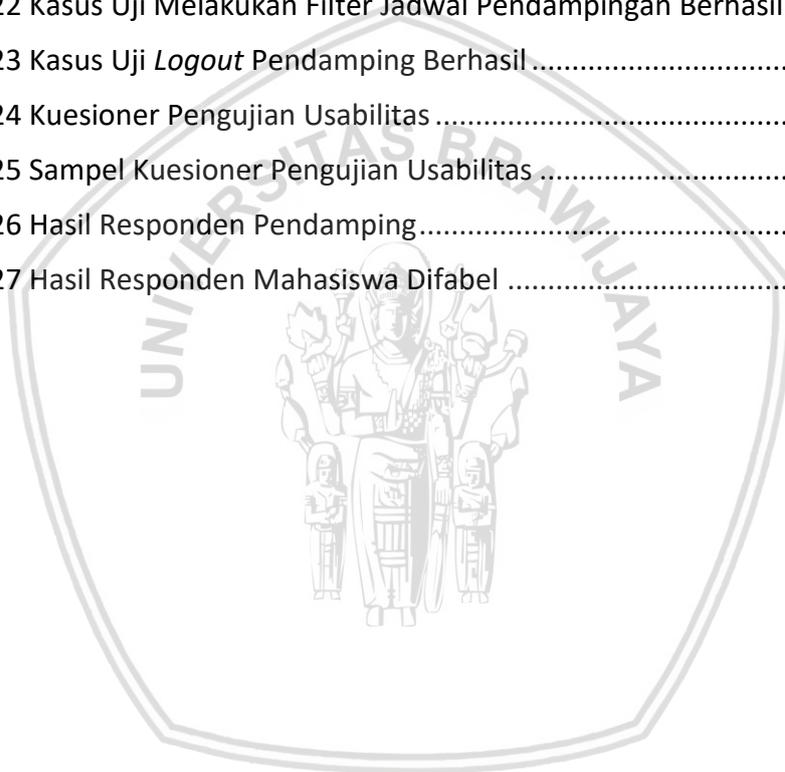


## DAFTAR TABEL

Tabel 2.1 Kelebihan dan Kekurangan <i>Software Prototyping</i> .....	10
Tabel 2.2 Versi Android .....	16
Tabel 2.3 Simbol Pada <i>Activity Diagram</i> .....	23
Tabel 2.4 Template Kuisisioner SUS .....	27
Tabel 4. 1 Identifikasi Aktor .....	35
Tabel 4.2 Kebutuhan Fungsional Pengguna Umum <i>Prototype 1</i> .....	36
Tabel 4.3 Kebutuhan Fungsional Pendamping <i>Prototype 1</i> .....	37
Tabel 4.4 Kebutuhan Fungsional Pendamping <i>Prototype 2</i> .....	38
Tabel 4.5 Kebutuhan Fungsional Mahasiswa Difabel <i>Prototype 1</i> .....	38
Tabel 4.6 Kebutuhan Fungsional Mahasiswa Difabel <i>Prototype 2</i> .....	40
Tabel 4.7 Kebutuhan Non Fungsional <i>Prototype 1</i> .....	40
Tabel 4.8 <i>Use Case Scenario Login Prototype 1</i> .....	42
Tabel 4.9 <i>Use Case Scenario</i> Melihat Jadwal Damping <i>Prototype 1</i> .....	43
Tabel 4.10 <i>Use Case Scenario</i> Menelpon Difabel <i>Prototype 1</i> .....	44
Tabel 4.11 <i>Use Case Scenario</i> Mengirim Pesan Difabel <i>Prototype 1</i> .....	45
Tabel 4. 12 <i>Use Case Scenario</i> Melakukan Konfirmasi Kehadiran <i>Prototype 1</i> ....	46
Tabel 4.13 <i>Use Case Scenario</i> Melakukan Filter Jadwal Damping <i>Prototype 1</i> ....	47
Tabel 4.14 <i>Use Case Scenario Logout</i> Pendamping <i>Prototype 1</i> .....	47
Tabel 4.15 <i>Use Case Scenario</i> Melihat Profil Difabel <i>Prototype 2</i> .....	48
Tabel 4.16 <i>Use Case Scenario</i> Melihat Jadwal <i>Prototype 1</i> .....	49
Tabel 4.17 <i>Use Case Scenario</i> Menelpon Pendamping <i>Prototype 1</i> .....	50
Tabel 4.18 <i>Use Case Scenario</i> Mengirim Pesan Pendamping <i>Prototype 1</i> .....	51
Tabel 4. 19 <i>Use Case Scenario</i> Menampilkan QR Code <i>Prototype 1</i> .....	51
Tabel 4.20 <i>Use Case Scenario</i> Melakukan Filter Jadwal Pendampingan <i>Prototype 1</i> .....	52
Tabel 4.21 <i>Use Case Scenario</i> Memberi Review Pendamping <i>Prototype 1</i> .....	53
Tabel 4.22 <i>Use Case Scenario Logout</i> Mahasiswa Difabel <i>Prototype 1</i> .....	54
Tabel 4.23 <i>Use Case Scenario</i> Melihat Profil Pendamping <i>Prototype 2</i> .....	55
Tabel 5.1 Tabel <i>User</i> .....	69
Tabel 5.2 Tabel <i>Students</i> .....	69

Tabel 5.3 Tabel <i>Majors</i> .....	70
Tabel 5.4 Tabel <i>Faculties</i> .....	70
Tabel 5.5 Tabel <i>Schedule_Student</i> .....	70
Tabel 5.6 Tabel <i>Schedule_Volunteer</i> .....	71
Tabel 5.7 Tabel <i>Accompaniment</i> .....	71
Tabel 5.8 Tabel <i>Permit</i> .....	72
Tabel 5.9 Tabel API V-Apps.....	72
Tabel 5.10 Penjelasan Antarmuka Halaman <i>Login Prototype 1</i> .....	76
Tabel 5.11 Antarmuka Melihat Jadwal Damping <i>prototype 1</i> .....	77
Tabel 5.12 Antarmuka Halaman <i>Navigation Bar</i> Pendamping <i>Prototype 1</i> .....	78
Tabel 5.13 Antarmuka Melihat Profil Difabel <i>Prototype 1</i> .....	79
Tabel 5.14 Antarmuka Melihat Jadwal Pendampingan <i>Prototype 1</i> .....	79
Tabel 5.15 Antarmuka Halaman <i>Navigation Bar Prototype 1</i> .....	81
Tabel 5.16 Antarmuka Melihat Profil Pendamping <i>Prototype 2</i> .....	82
Tabel 5.17 Perancangan Antarmuka Halaman <i>Review Prototype 1</i> .....	82
Tabel 5.18 Spesifikasi Perangkat Keras Komputer.....	83
Tabel 5.19 Spesifikasi Perangkat Keras <i>Smartphone</i> .....	83
Tabel 5.20 Spesifikasi Perangkat Lunak Komputer.....	84
Tabel 5.21 Spesifikasi Perangkat Lunak <i>Smartphone</i> .....	84
Tabel 6.1 Desain <i>Test Case loadVolunteerSchedule</i> .....	99
Tabel 6.2 Hasil <i>Test Case loadVolunteerSchedule</i> .....	99
Tabel 6.3 Desain <i>Test Case loadMahasiswaSchedule</i> .....	101
Tabel 6.4 Hasil <i>Test Case loadMahasiswaSchedule</i> .....	101
Tabel 6.5 Desain <i>Test Case onActivityResult</i> .....	104
Tabel 6.6 Hasil <i>Test Case onActivityResult</i> .....	105
Tabel 6.7 Kasus Uji <i>Login</i> Berhasil.....	105
Tabel 6.8 Kasus Uji <i>Login</i> Gagal.....	106
Tabel 6.9 Kasus Uji Melihat Jadwal Damping Berhasil.....	106
Tabel 6.10 Kasus Uji Menelpon Difabel Berhasil.....	107
Tabel 6.11 Kasus Uji Mengirim Pesan Difabel Berhasil.....	107
Tabel 6.12 Kasus Uji Melakukan Konfirmasi Kehadiran Berhasil.....	108
Tabel 6.13 Kasus Uji Melihat Profil Difabel Berhasil.....	108

Tabel 6.14 Kasus Uji Melakukan Filter Jadwal Damping Berhasil .....	108
Tabel 6.15 Kasus Uji <i>Logout</i> Pendamping Berhasil .....	109
Tabel 6.16 Kasus Uji Melihat Jadwal Pendampingan Berhasil .....	109
Tabel 6.17 Kasus Uji Fungsi Menelpon Pendamping Berhasil .....	110
Tabel 6.18 Kasus Uji Mengirim Pesan Pendamping Berhasil .....	110
Tabel 6.19 Kasus Uji Menampilkan <i>QR Code</i> Berhasil .....	110
Tabel 6.20 Kasus Uji Memberi <i>Review</i> Pendamping Berhasil .....	111
Tabel 6.21 Kasus Uji Melihat Profil Pendamping Berhasil .....	111
Tabel 6.22 Kasus Uji Melakukan Filter Jadwal Pendampingan Berhasil .....	111
Tabel 6.23 Kasus Uji <i>Logout</i> Pendamping Berhasil .....	112
Tabel 6.24 Kuesioner Pengujian Usabilitas .....	130
Tabel 6.25 Sampel Kuesioner Pengujian Usabilitas .....	132
Tabel 6.26 Hasil Responden Pendamping .....	133
Tabel 6.27 Hasil Responden Mahasiswa Difabel .....	133



## DAFTAR GAMBAR

Gambar 2.1 Siklus Hidup Pengembangan Perangkat Lunak .....	9
Gambar 2.2 SDLC <i>Software Prototyping</i> . .....	9
Gambar 2.3 Bentuk JSON <i>Object</i> .....	11
Gambar 2.4 Bentuk JSON <i>Array</i> .....	11
Gambar 2.5 Bentuk JSON <i>Value</i> . .....	12
Gambar 2.6 Bentuk JSON <i>String</i> .....	12
Gambar 2.7 Bentuk JSON <i>Number</i> . .....	13
Gambar 2.8 Simbol dalam <i>Entity Relationship Diagram</i> .....	14
Gambar 2.9 Contoh Kondisi Satu ke Satu .....	14
Gambar 2.10 Contoh Kondisi Satu ke Banyak.....	15
Gambar 2.11 Contoh Kondisi Banyak ke Banyak .....	15
Gambar 2.12 Contoh Kondisi Banyak ke Banyak .....	16
Gambar 2.13 Contoh Diagram <i>Use Case</i> .....	18
Gambar 2.14 Simbol Aktor .....	19
Gambar 2.15 Contoh Hubungan <i>Association</i> .....	19
Gambar 2.16 Contoh Hubungan <i>Extends</i> .....	19
Gambar 2.17 Contoh Hubungan <i>Depends On</i> .....	20
Gambar 2.18 <i>Class</i> .....	20
Gambar 2.19 <i>Association</i> .....	21
Gambar 2.20 <i>Composition</i> .....	21
Gambar 2.21 <i>Agregation</i> .....	21
Gambar 2.22 Contoh <i>Class Diagram</i> .....	22
Gambar 2.23 Contoh <i>Sequence Diagram</i> .....	24
Gambar 2.24 Transformasi <i>Flowchart</i> (a) ke <i>Flowgraph</i> (b) .....	26
Gambar 2.25 <i>Percentil Rank SUS</i> . .....	28
Gambar 3.1 Diagram Metodologi Penelitian. ....	29
Gambar 4.1 <i>Use Case Diagram V-Apps Prototype 1</i> .....	41
Gambar 4.2 <i>Use Case Diagram V-Apps Prototype 2</i> .....	42
Gambar 5.1 Perancangan Arsitektur Sistem <i>Prototype 1</i> .....	56
Gambar 5. 2 Perancangan <i>Class Diagram Prototype 1</i> .....	58

Gambar 5.3 Perancangan <i>Class Diagram Prototype 2</i> .....	59
Gambar 5. 4 Perancangan <i>Class Diagram Package Model</i> .....	60
Gambar 5. 5 Perancangan <i>Class Diagram Package View</i> .....	61
Gambar 5. 6 Perancangan <i>Class Diagram Package API</i> .....	62
Gambar 5.7 <i>Sequence Diagram</i> Melihat Jadwal Damping <i>Prototype 1</i> .....	63
Gambar 5.8 <i>Sequence Diagram</i> Melakukan Konfirmasi Kehadiran <i>Prototype 1</i> ..	63
Gambar 5.9 <i>Sequence Diagram</i> Melihat Profil Difabel <i>Prototype 2</i> .....	64
Gambar 5.10 <i>Sequence Diagram</i> Melihat Jadwal Pendampingan <i>Prototype 1</i> ....	64
Gambar 5.11 <i>Sequence Diagram</i> Melihat Profil Pendamping <i>Prototype 2</i> .....	65
Gambar 5.12 Perancangan <i>Entity Relationship Diagram Prototype 1</i> .....	68
Gambar 5.13 Perancangan Antarmuka Halaman <i>Login Prototype 1</i> .....	75
Gambar 5.14 Perancangan Antarmuka Melihat Jadwal Damping <i>prototype 1</i> ....	76
Gambar 5.15 Perancangan Antarmuka Halaman <i>Navigation Bar</i> Pendamping <i>Prototype 1</i> .....	77
Gambar 5.16 Perancangan Antarmuka Melihat Profil Difabel <i>prototype 1</i> .....	78
Gambar 5.17 Perancangan Antarmuka Melihat Jadwal Pendampingan <i>Prototype 1</i> .....	79
Gambar 5.18 Perancangan Antarmuka Halaman <i>Navigation Bar Prototype 1</i> ....	80
Gambar 5.19 Perancangan Antarmuka Melihat Profil Pendamping <i>Prototype 2</i> .	81
Gambar 5.20 Perancangan Antarmuka Halaman <i>Review Prototype 1</i> .....	82
Gambar 5.21 Implementasi Basis Data .....	85
Gambar 5.22 Implementasi Antarmuka Halaman <i>Login</i> . .....	92
Gambar 5.23 Implementasi Antarmuka Halaman Melihat Jadwal Damping .....	93
Gambar 5.24 Implementasi Antarmuka Halaman <i>Navigation Bar</i> Pendamping..	93
Gambar 5.25 Implementasi Antarmuka Halaman Melihat Profil Difabel.....	94
Gambar 5. 26 Implementasi Antarmuka Halaman Melihat Jadwal Pendampingan .....	94
Gambar 5.27 Implementasi Antarmuka Halaman <i>Navigation Bar</i> Dofabel. ....	95
Gambar 5.28 Implementasi Antarmuka Halaman Melihat Profil Pendamping....	95
Gambar 5.29 Implementasi Antarmuka Halaman <i>Review</i> .....	96
Gambar 6.1 Kode Program <i>method loadVolunteerSchedule()</i> .....	97
Gambar 6.2 <i>Flow Graph method loadVolunteerSchedule()</i> .....	98
Gambar 6.3 Kode Program <i>method loadMahasiswaSchedule()</i> .....	100



Gambar 6.4 *Flow Graph* method `loadMahasiswaSchedule()` ..... 100  
Gambar 6.5 Kode Program `onActivityResult()` ..... 102  
Gambar 6.6 *Flow Graph* method `onActivityResult()` ..... 103



## DAFTAR LAMPIRAN

Lampiran A Transkrip Wawancara 1 .....	139
Lampiran B Transkrip Wawancara 2 .....	141
Lampiran C Transkrip Wawancara 3 .....	143
Lampiran D kuisisioner .....	145
Lampiran E Evaluasi Pengguna.....	147



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Pusat Studi dan Layanan Disabilitas Universitas Brawijaya (PSLD UB) merupakan lembaga yang memiliki dua fungsi, yaitu sebagai pusat pelayanan bagi penyandang disabilitas (difabel) di Universitas Brawijaya, dan sebagai pusat studi yang mengaji isu-isu disabilitas. Mahasiswa penyandang disabilitas adalah mereka yang memiliki ketunaan (*impairment*) sehingga mereka membutuhkan alat bantu khusus, modifikasi lingkungan atau teknik – teknik alternatif untuk dapat berpartisipasi dalam proses belajar mengajar. Salah satu pelayanan yang diberikan bagi mahasiswa penyandang disabilitas yaitu mendapatkan pendampingan pada saat perkuliahan. Serta mendapatkan pendamping pengganti apabila pendamping utama tidak bisa mendampingi. Pendamping merupakan mahasiswa Universitas Brawijaya yang sudah mengikuti seleksi rekrutmen pendamping di Pusat Studi dan Layanan Disabilitas Universitas Brawijaya (PSLD UB) dan telah mengikuti pelatihan *disability awareness* dan bahasa isyarat.

Bedasarkan hasil wawancara kepada staff administrasi pendampingan PSLD UB, proses pendampingan ini sudah dijadwalkan oleh pihak PSLD UB. Sistem penjadwalan yang ada di PSLD UB sudah dilakukan secara otomatis dengan menggunakan sistem yang ada, akan tetapi penyampaian informasi yang dilakukan kepada pendamping dan mahasiswa difabel masih dilakukan secara manual. Penyampaian informasi yang dilakukan masih menggunakan *hard copy* yang di tempel di papan pengumuman atau pihak administrasi pendampingan PSLD UB menghubungi secara personal kepada mahasiswa yang bersangkutan. Menurut staff administrasi pendampingan PSLD UB kurang mudah menyampaikan informasi jadwal tersebut, karena informasi yang disampaikan terkadang tidak tersampaikan. Hal ini disebabkan karena banyaknya jumlah pendamping dan mahasiswa difabel. Berdasarkan data jumlah mahasiswa difabel per semester ganjil 2016/2017, PSLD UB mempunyai 80 mahasiswa difabel dengan berbagai jenis kecacatan dan 70 volunteer dari berbagai macam fakultas di Universitas Brawijaya. Hal ini tidak sebanding dengan staff administrasi pendampingan PSLD UB yang hanya berjumlah 1 orang dan harus menyampaikan informasi jadwal pendampingan secara personal ke semua pendamping dan mahasiswa difabel.

Berdasarkan hasil wawancara dengan salah satu mahasiswa penyandang tunarungu yang terdaftar di PSLD UB, mahasiswa difabel terkadang kesulitan untuk mendapatkan informasi tambahan seperti informasi jadwal damping yang baru. Apabila ingin mendapatkan informasi jadwal damping yang baru, harus ke kantor PSLD untuk memotret *hard copy* yang di tempel di papan pengumuman atau menunggu informasi tersebut di *share group WhatsApp*. Apabila pergantian jadwal damping selama seminggu tidak tersampaikan, maka mahasiswa difabel tidak mendapatkan fasilitas pelayanan pendampingan.

Berdasarkan hasil pemaparan yang disampaikan oleh salah satu mahasiswa penyandang tunarungu, dia akan kesusahan untuk mengikuti pembelajaran di kelas apabila tidak mendapatkan pendamping, serta tidak memahami apa yang diajarkan dan tugas yang diberikan oleh dosen. Hal ini akan menyebabkan PSLD UB mendapatkan komplain karena tidak memenuhi kewajiban untuk memberikan pelayanan berupa pendampingan kepada mahasiswa difabel. Masalah lain yang muncul disampaikan oleh salah satu pendamping yang terdaftar di PSLD UB. Ketika proses pendampingan selesai, pendamping dan mahasiswa difabel harus menuju kantor PSLD UB. Tujuannya yaitu pendamping mengisi *form* kehadiran pendampingan dan mahasiswa difabel mengisi *form* penilaian terhadap pendamping yang mendampingi. Apabila *form* kehadiran pendampingan dan penilaian pendamping tidak diisi maka proses pendampingan dianggap tidak dijalankan. Menurut pemaparan salah satu pendamping yang terdaftar di PSLD UB, mahasiswa difabel yang bisa berkomunikasi lewat *handphone* dan menggunakan aplikasi dengan memahami *user interface* dengan baik yang ada di *smartphone* mereka hanya mahasiswa disabilitas yang Tuli dan Tunadaksa. Berdasarkan data yang ada, sekitar 80 mahasiswa disabilitas yang terdaftar di PSLD UB, terdapat 55% Tuli dan 15% Tunadaksa.

Dari masalah-masalah yang terjadi, penulis melakukan survey dengan memberika quisioner yang berisi pertanyaan kepada 15 pendamping dan 15 mahasiswa difabel untuk mengetahui solusi yang ada. Dalam hasil survey responden dalam mengatasi masalah kurang mudahnya dalam menyampaikan informasi jadwal damping, terdapat beberapa cara yang dapat dilakukan. Misalnya, informasi tersebut dapat disampaikan menggunakan *website*, atau menggunakan aplikasi. Akan tetapi berdasarkan *survey* yang penulis lakukan, 20% responden menjawab informasi ditampilkan di *website*, dan 80% memilih menggunakan aplikasi berbasis android. Dari hasil data tersebut, calon pengguna cenderung memilih penyampaian informasi pendampingan menggunakan aplikasi. Penulis juga menanyakan *platform* yang akan digunakan oleh calon pengguna terhadap aplikasi yang dibuat, 25 dari 30 responden menyatakan menggunakan *platform* android.

Dari hasil survey yang dilakukan, diperlukan aplikasi yang dapat memudahkan dalam penyampaian informasi jadwal damping antara pendamping dan mahasiswa difabel. Dan untuk membuat aplikasi tersebut, diperlukan data damping yang ada di sistem penjadwalan kantor PSLD UB. Sehingga, data yang ditampilkan benar-benar data pendampingan yang sudah dijadwalkan. Oleh karena itu, dibangun sebuah aplikasi bernama aplikasi *Volunteer Apps* (V-Apps) yang memiliki fungsi untuk menampilkan informasi jadwal pendampingan. Aplikasi *Volunteer Apps* berjalan pada platform Android. Hal ini didasari karena banyaknya calon pengguna yang menggunakan *platform* Android.

Pengembangan Aplikasi *Volunteer Apps* akan menggunakan model *prototyping* sebagai model *Software Development Life Cycle* (SDLC). Model

*prototyping* dipilih karena memungkinkan untuk memahami kebutuhan calon pengguna pada tahap awal pengembangan. Serta membantu mendapatkan *feedback* yang berharga dari calon pengguna dan membantu penulis memahami apa yang sebenarnya diharapkan dari produk yang sedang dikembangkan. *Prototyping* digunakan untuk memungkinkan calon pengguna mengevaluasi sistem yang dirancang di awal oleh penulis dan mencobanya sebelum di implementasikan (tutorialpoint, 2017).

Diharapkan dengan adanya Aplikasi *Volunteer Apps*, PSLD UB dapat menyampaikan informasi jadwal pendampingan secara mudah dan menyeluruh kepada pendamping dan Mahasiswa difabel, sehingga tidak ada lagi informasi yang tidak tersampaikan dan menyebabkan mahasiswa difabel tidak mendapatkan fasilitas pendampingan. Mahasiswa difabel dapat mengetahui pendamping pengganti secara langsung tanpa menghubungi kantor PSLD. Pendamping dan mahasiswa difabel tidak perlu ke kantor PSLD UB untuk mengisi kartu kehadiran dan penlaiaan. Karena konfirmasi kehadiran dan penilaian pendamping dapat dilakukan dengan menggunakan aplikasi *Volunteer Apps*. Dengan ini, PSLD UB tidak perlu menghubungi secara personal untuk menyampaikan informasi jadwal pendampingan kepada pendamping dan mahasiswa difabel. Kemudahan dalam menggunakan aplikasi ini dibuktikan dengan menggunakan *usability testing*. Pengujian ini dilakukan untuk memastikan bahwa sistem yang dibangun sudah sesuai dan dapat digunakan dengan mudah.

## 1.2 Rumusan masalah

Berdasarkan uraian latar belakang yang sudah dipaparkan di atas, maka rumusan masalah yang dapat diambil dari penelitian ini adalah:

1. Apa saja kebutuhan yang digunakan dalam pengembangan perangkat lunak V-Apps sebagai aplikasi untuk menginformasikan jadwal damping mahasiswa difabel?
2. Bagaimana hasil rancangan, implementasi, dan pengujian dalam pengembangan perangkat lunak V-Apps sebagai aplikasi untuk menginformasikan jadwal damping mahasiswa difabel?
3. Bagaimana tingkat kemudahan pengguna pada saat menggunakan aplikasi V-Apps?

## 1.3 Tujuan

Tujuan yang ingin dicapai dari hasil penelitian ini adalah:

1. Mengetahui kebutuhan yang digunakan dalam pengembangan perangkat lunak V-Apps sebagai aplikasi untuk menginformasikan jadwal damping mahasiswa difabel.
2. Menghasilkan rancangan dan implementasi berdasarkan hasil analisis kebutuhan yang telah dilakukan.

3. Mengetahui hasil pengujian pada pengembangan perangkat lunak V-Apps sebagai aplikasi untuk menginformasikan jadwal damping mahasiswa difabel.
4. Mengetahui tingkat kemudahan pada aplikasi V-Apps untuk membantu menginformasikan jadwal damping mahasiswa difabel.

#### **1.4 Manfaat**

Manfaat yang didapatkan dari pembuatan skripsi ini adalah:

1. Menghasilkan suatu pengembangan perangkat lunak V-Apps sebagai aplikasi untuk menginformasikan jadwal damping mahasiswa difabel.
2. Membantu pihak PSLD Universitas Brawijaya dalam mengoptimalkan sumber daya, sehingga pada akhirnya meningkatkan produktivitas PSLD Universitas Brawijaya.
3. Dapat digunakan sebagai alternatif penyampaian informasi jadwal damping bagi pendamping dan mahasiswa difabel.
4. Memberikan pilihan kepada PSLD Universitas Brawijaya bahwa dalam menyampaikan jadwal damping mahasiswa difabel ada cara yang lebih mudah.
5. Pendamping dan Mahasiswa difabel dapat terbantu dengan adanya aplikasi V-apps.
6. Penelitian ini diharapkan dapat menambah pengalaman penulis dalam mengembangkan aplikasi mobile android yang berguna bagi masyarakat.

#### **1.5 Batasan masalah**

Berdasarkan uraian latar belakang dan rumusan masalah yang telah diuraikan sebelumnya, dan pembahasan penelitian ini tidak menyimpang dari apa yang di rumuskan. Maka diperlukan batasan-batasan, batasan-batasan penelitian ini adalah:

1. Studi kasus penelitian ini dilakukan di Pusat Studi dan Layanan Disabilitas Universitas Brawijaya, sehingga permasalahan yang dianalisis pada penelitian ini adalah permasalahan yang hanya dialami oleh Pusat Studi dan Layanan Disabilitas Universitas Brawijaya.
2. Pengguna aplikasi hanya dituju kepada pendamping dan mahasiswa difabel yang terdaftar di Pusat Studi dan Layanan Disabilitas Universitas Brawijaya.
3. Data yang digunakan adalah data jadwal damping pada sistem penjadwalan jadwal damping di kantor PSLD UB dengan menggunakan API.
4. Aplikasi yang dikembangkan hanyalah aplikasi berbasis android.

#### **1.6 Sistematika pembahasan**

Penyusun Skripsi ini menggunakan kerangka pembahasan yang tersusun sebagai berikut:

1. BAB I PENDAHULUAN

Pada bab ini menjelaskan tentang dasar penyusunan skripsi. Dalam penyusunan dasar skripsi ini dijabarkan dalam latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

## 2. BAB II LANDASAN KEPUSTAKAAN

Bab ini menjelaskan dasar teori yang akan digunakan pada skripsi. Dalam bab ini berisi pustaka, referensi, dan sumber-sumber yang dapat membantu dalam menyelesaikan permasalahan dalam skripsi.

## 3. BAB III METODOLOGI

Bab ini menjelaskan bagaimana metodologi yang akan digunakan untuk perancangan dalam skripsi ini serta perancangan yang diterapkan.

## 4. BAB IV REKAYASA PRASYARAT / KEBUTUHAN

Bab ini menjelaskan proses analisis kebutuhan pengguna dan kebutuhan sistem serta pemilihan metode.

## 5. BAB V PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan proses perancangan berdasarkan kebutuhan yang telah didapatkan dan implementasi atau tahap-tahap pembuatan aplikasi sistem informasi penjadwalan berdasarkan kebutuhan dan perancangan yang telah didefinisikan.

## 6. BAB VI PENGUJIAN

Bab ini berisi pengujian aplikasi yang telah dibangun. Pengujian perangkat lunak berdasarkan *black box*, *white box testing*, dan *usability*.

## 7. BAB VII PENUTUP

Bab ini berisi kesimpulan dari hasil pembuatan dan pengujian aplikasi serta saran bagi pengembangan selanjutnya.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Dasar Teori

#### 2.1.1 Pusat Studi dan Layanan Disabilitas Universitas Brawijaya

Pusat Studi dan Layanan Disabilitas Universitas Brawijaya (PSLD UB) adalah lembaga yang memiliki dua fungsi, yaitu: (1) sebagai pusat pelayanan bagi penyandang disabilitas (difabel) di Universitas Brawijaya; (2) sebagai pusat studi yang mengaji isu-isu disabilitas. Lembaga ini didirikan pada tanggal 19 Maret 2012, dan disahkan melalui Surat Keputusan Rektor Universitas Brawijaya Nomor: 135/SK/2012.

PSLD UB diprakarsai oleh beberapa dosen muda dan mahasiswa UB, yang ingin melakukan perubahan setelah melihat data dari Badan Pendidikan dan Penelitian Kementerian Sosial (Mardjuki, 2010) yang menunjukkan bahwa hanya 0,95% penyandang disabilitas yang memiliki ijazah S1. Salah satu penyebab hal ini adalah minimnya akses Perguruan Tinggi di Indonesia bagi penyandang disabilitas. Akses perguruan tinggi bagi penyandang disabilitas, meliputi: metode belajar yang adaptif, ruang kelas dan gedung kuliah yang *accessible*, pelayanan yang tidak diskriminatif dan memahami kebutuhan penyandang disabilitas. Kehadiran PSLD UB diharapkan dapat memberikan kesempatan seluas-luasnya bagi penyandang disabilitas dalam mengakses perguruan tinggi. Selain itu PSLD UB juga diharapkan dapat melahirkan kajian-kajian terkini tentang isu-isu disabilitas, dan dapat menjadi *Resource Centre* bagi seluruh PSLD di Indonesia.

#### 2.1.2 Pendamping

Dalam Peraturan Menteri Pendidikan dan Kebudayaan RI Nomor 46 tahun 2014 tentang Pendidikan Khusus, Pendidikan Layanan Khusus dan / atau Pembelajaran Khusus Pada Pendidikan Tinggi, dijelaskan bahwa Penyelenggara Pendidikan Tinggi wajib memberikan layanan khusus bagi mahasiswa difabel. Pendamping merupakan salah satu wujud layanan khusus yang diberikan kepada Mahasiswa penyandang disabilitas. Pendamping merupakan mahasiswa Universitas Brawijaya yang mengikuti seleksi rekrutmen pendamping dan telah mengikuti pelatihan Disability Awareness dan Bahasa Isyarat (Buku Panduan PSLD, 2016). Berikut dijelaskan tentang kewajiban dan hak pendamping mahasiswa difabel:

##### **Kewajiban Pendamping**

1. Mendampingi MABA pada saat perkuliahan, mendampingi saat mengerjakan tugas kuliah, PKK MABA, dan mengurus administrasi perkuliahan.
2. Mencari pengganti jika tidak dapat melaksanakan tugas pendampingan (paling lambat H-1 hari pendampingan).

3. Membantu MABA dalam memahami materi perkuliahan dan mendapatkan bahan kuliah dengan format yang dapat diakses sesuai dengan kebutuhan MABA, termasuk mendampingi ketika bertemu dosen untuk bimbingan.
4. Memberikan informasi dan mengoreksi tugas (perkuliahan/OSPEK) MABA (redaksional/teknis, bukan substansi).
5. Belajar dan memperdalam pemahaman tentang bahasa isyarat dan disability awareness.
6. Menyelesaikan kontrak 1 tahun yang telah disepakati

#### **Hak Pendamping**

1. Menolak permintaan mahasiswa difabel diluar tugas (kewajiban) pendamping.
2. Mendapatkan biaya transportasi dan komunikasi sesuai dengan kontrak.
3. Mendapatkan libur pada hari Minggu.
4. Mengundurkan diri pada saat kontrak habis

#### **2.1.3 Mahasiswa Difabel**

Mahasiswa Penyandang Disabilitas adalah mereka yang memiliki ketunaan (*impairment*) sehingga mereka membutuhkan alat bantu khusus, modifikasi lingkungan atau teknik-teknik alternatif untuk dapat berpartisipasi dalam proses belajar dan kegiatan akademik lainnya dan memiliki peluang yang sama seperti mahasiswa lainnya untuk berhasil (Buku Panduan PSLD, 2016). Berikut ini adalah penjelasan tentang kewajiban yang harus dilaksanakan oleh mahasiswa difabel dan juga hak yang harus didapatkan oleh mahasiswa difabel dalam pendidikan:

##### **Kewajiban Mahasiswa Difabel**

1. Mengerjakan sendiri tugas kuliah dan tugas PKK MABA.
2. Mandiri dalam mobilitas dari rumah (kosan) ke kampus.
3. Mengikuti kegiatan perkuliahan dan PKK MABA dengan disiplin.
4. Mengisi kartu presensi (daftar hadir) pendamping.

##### **Hak Mahasiswa Difabel**

1. Mendapatkan pendampingan pada saat perkuliahan, mengerjakan tugas kuliah, PKK MABA, dan mengurus administrasi perkuliahan.
2. Mendapatkan pendamping pengganti jika pendamping utama tidak bisa mendampingi.
3. Mendapatkan informasi yang jelas tentang tugas dan aktivitas perkuliahan lainnya.
4. Mendapatkan bahan kuliah dengan format yang dapat diakses sesuai dengan kebutuhan Mahasiswa Difabel.

5. Mendapatkan pelayanan yang adil.
6. Menolak untuk didampingi karena sudah merasa mandiri.
7. Mengajukan pendamping pengganti jika merasa tidak nyaman dengan pendamping yang telah ditunjuk mendampingi.

#### 2.1.4 Rekayasa Perangkat Lunak

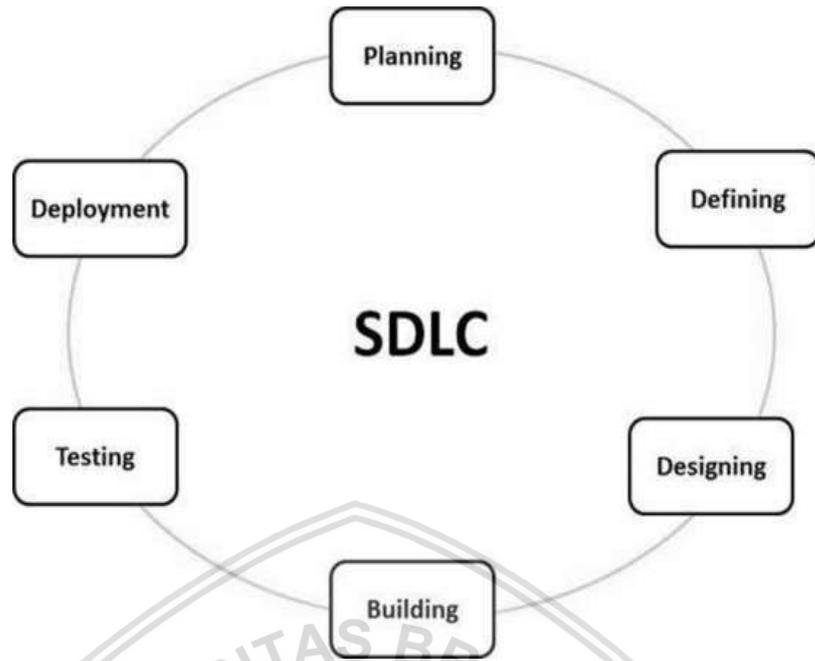
Rekayasa perangkat lunak merupakan sebuah disiplin ilmu yang mempelajari berbagai aspek perencanaan serta produksi perangkat lunak agar biaya dan waktu yang dibutuhkan untuk mencapai suatu kualitas tertentu dari perangkat lunak tersebut dapat lebih optimal. Tentunya untuk mencapai semua itu akan membutuhkan pendekatan sistematis dan terorganisir, dan kreativitas dalam pemilihan suatu metode yang digunakan pada kondisi tertentu (Sommerville, 2010).

Pendekatan sistematis yang disebutkan tersebut dapat dikatakan sebagai *software process*. *Software process* dibagi menjadi 4 kegiatan utama yang umum, berikut adalah penjelasannya:

1. *Software Specification*, tahapan yang akan menentukan batasan-batasan pada operasi pada perangkat lunak.
2. *Software Development*, suatu proses tahapan perancangan perangkat lunak dan juga proses membangun perangkat lunak berdasarkan perancangan yang telah dibuat atau dilakukan.
3. *Software Validation*, adalah suatu tahapan untuk melakukan pengujian perangkat lunak yang dibangun agar sesuai dengan spesifikasi perangkat lunak.
4. *Software Evolution*, merupakan tahapan penyesuaian perangkat lunak yang dilakukan berdasarkan permintaan perubahan kebutuhan pada suatu perangkat lunak.

#### 2.1.5 Software Development Life Cycle (SDLC)

SDLC atau bisa disebut *Software Development Life Cycle* adalah siklus hidup suatu perangkat lunak dimana suatu perangkat lunak akan menempuh suatu proses untuk meningkatkan kualitas pada perangkat lunak. Serta proses pembangunan secara menyeluruh. *Software Development Life Cycle* digunakan oleh para pengembang untuk mendefinisikan suatu rencana secara jelas untuk mengembangkan perangkat lunak, memeliharanya, melakukan pemantauan dan juga melakukan suatu perubahan atau suatu perbaikan perangkat lunak (tutorialpoint, 2017). Siklus hidup suatu perangkat lunak dapat dilihat pada Gambar 2.1.

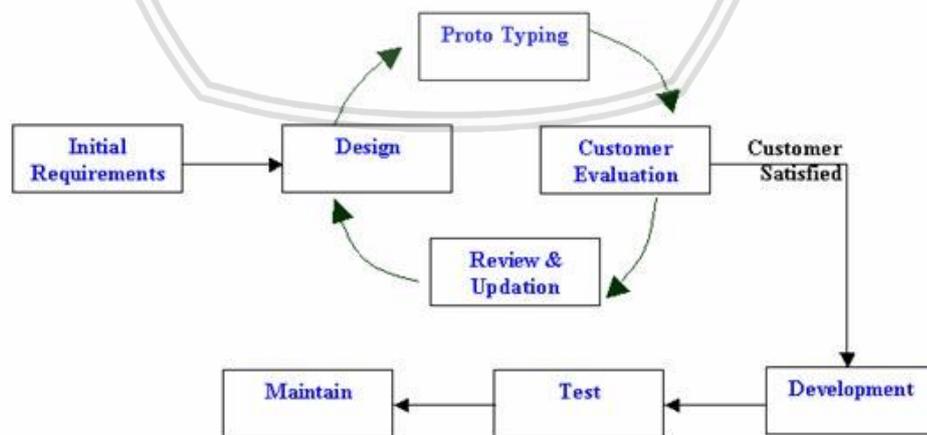


Gambar 2.1 Siklus Hidup Pengembangan Perangkat Lunak

Sumber : (tutorialpoint, 2017)

### 2.1.5.1 Software Prototyping

Salah satu SDLC yang dapat digunakan adalah *Software prototyping*. *Software prototyping* mengacu pada pembuatan sebuah *prototype* yang akan digunakan untuk memahami suatu kebutuhan pengguna pada tahap awal pengembangan. Suatu *prototype* yang dibuat memiliki fungsi yang menggambarkan suatu perangkat lunak dan bisa jadi untuk logika atau *algoritme* belum ada. Pada Gambar 2.2 menjelaskan alur dari *software prototyping*.



Proto Type Model

Gambar 2.2 SDLC Software Prototyping.



Suatu *prototype* adalah cara dimana pengguna dapat mengevaluasi suatu perangkat lunak yang dibuat oleh pengembang dan dicoba sebelum dilakukan pemasaran. *Prototype* ini akan berguna bagi pengembang untuk mengetahui kebutuhan yang diinginkan pengguna secara spesifik yang mungkin belum dipikirkan oleh pengembang sebelumnya.

Suatu *Software Prototyping* memiliki sebuah alur yaitu inialisasi kebutuhan, pembuatan desain dan kemudian dilanjutkan dengan pembuatan suatu *prototype*. Apabila pembuatan *prototype* sudah selesai. Maka dilakukan evaluasi pengguna, jika pada evaluasi pengguna ternyata pengguna tersebut kurang puas dengan *prototype* maka akan dilakukan *review* dan suatu perubahan desain yang akan disesuaikan dengan kebutuhan pengguna. Apabila pengguna sudah puas pada tahap evaluasi, maka akan dilanjutkan pada tahap implementasi. Batasan berhenti adalah apabila pengguna sudah puas dengan hasil *prototype* yang dibuat (Imron, 2017). Dalam *Software prototyping* terdapat beberapa tipe, seperti *rapid prototyping*, *Evolutionary prototyping*, *Incremental prototyping*, dan *Extreme prototyping* (tutorialpoint, 2017). Pada Tabel 2.3 akan ditampilkan keuntungan dan kerugian dalam penggunaan metode *prototyping*.

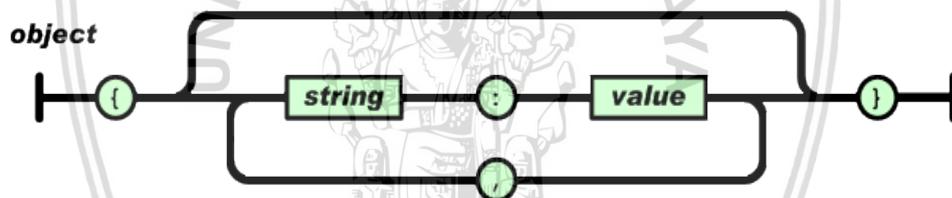
**Tabel 2.1 Kelebihan dan Kekurangan *Software Prototyping***

Keuntungan	Kerugian
Pengguna terlibat dalam pengembangan produk	Resiko analisis kebutuhan tidak cukup karena ketergantungan pada <i>prototype</i> .
Pengguna mendapat permasalahan lebih mendalam tentang sistem karena model kerjanya diperlihatkan.	Ada kemungkinan pengguna bingung akan <i>prototype</i> dan perangkat lunak yang sebenarnya.
Mengurangi waktu dan biaya serta kelemahan dapat dideteksi dapat dideteksi dari awal.	Dapat meningkatkan kompleksitas sistem sehingga membuat ruang lingkup sistem menjadi lebih luas dari yang seharusnya.
Umpan balik terhadap pengguna lebih cepat tersampaikan.	Bisa saja pengembang menggunakan kembali <i>prototype</i> untuk membangun sistem walaupun <i>prototype</i> tersebut kekurangan layak.
Kesalahan fungsi dalam perangkat lunak dapat diidentifikasi sejak awal.	Upaya diinvestasikan dalam membangun <i>prototype</i> mungkin lebih banyak jika diawasi.
Fungsi yang dirasa sulit bagi pengguna dapat diidentifikasi sejak awal	

### 2.1.6 JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) merupakan format teks yang dapat digunakan sebagai pertukaran data terstruktur antara semua bahasa pemrograman. JSON terinspirasi dari *Object literal JavaScript*. File JSON sendiri digunakan untuk transfer data dari *web-server* ke aplikasi. Bentuk data yang disimpan dalam file JSON disimpan dalam file JSON disimpan dalam bentuk *plaintext* sehingga mudah dalam membacanya (Heriyanto, 2016).

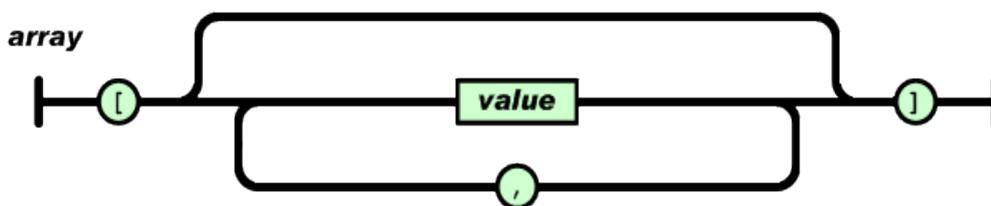
JSON terbuat dari dua unsur yaitu kumpulan pasangan nama/nilai dan daftar nilai yang terurut. Pasangan nilai / nama dapat dinyatakan sebagai, *object*, *record*, *sturct*, *dictionary*, *hash table*, *keyed list*, atau *assosiative array*. Sedangkan untuk nilai yang terurut dinyatakan dalam *array*, *vector*, *list*, atau *squence*. Struktur data ini disebut struktur data universal. JSON menggunakan beberapa bentuk dalam penggunaannya seperti *object*, *array*, *value*, *string*, dan *number* (json, nd). Struktur dari bentuk tersebut dapat dilihat pada Gambar 2.3, Gambar 2.4, Gambar 2.5, Gambar 2.6, serta Gambar 2.7.



Gambar 2.3 Bentuk JSON Object

Sumber : (json, 2017)

*Object* merupakan nilai / nama yang tidak terurut. *Object* diawali dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua). Tanda , (koma) digunakan untuk memisahkan setiap pasangan nilai / nama.

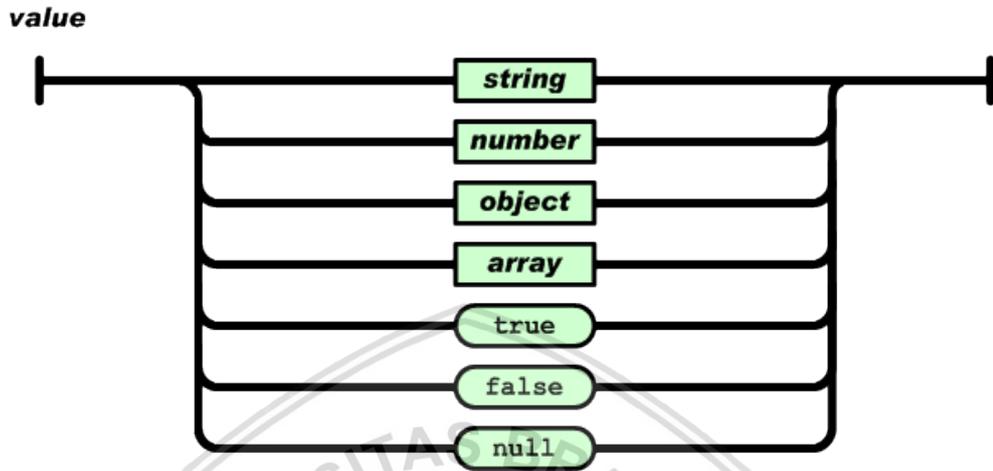


Gambar 2.4 Bentuk JSON Array

Sumber : (json. 2017)



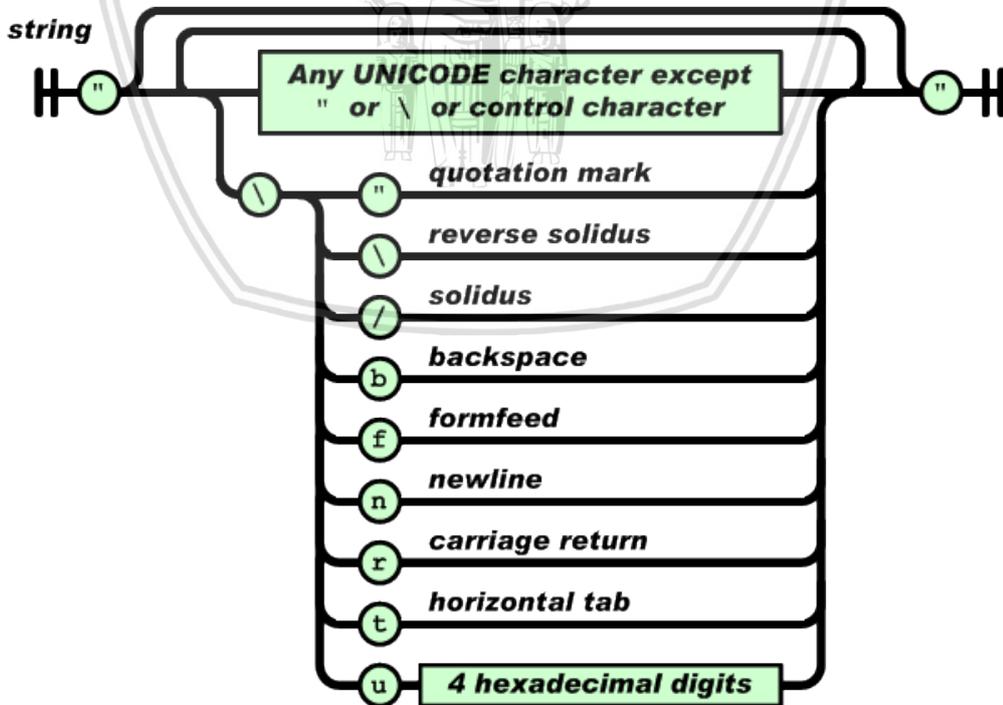
*Array* merupakan kumpulan nilai yang terurut. *Array* diawali dengan [ (kurung kotak buka) dan diakhiri dengan ] (kurung tutup kotak). Tanda , (koma) digunakan untuk memisahkan nilai.



Gambar 2.5 Bentuk JSON Value.

Sumber : (json, 2017)

Value dapat berupa object, array, string dalam tanda dikutip dua ("), number, atau true / false / null.

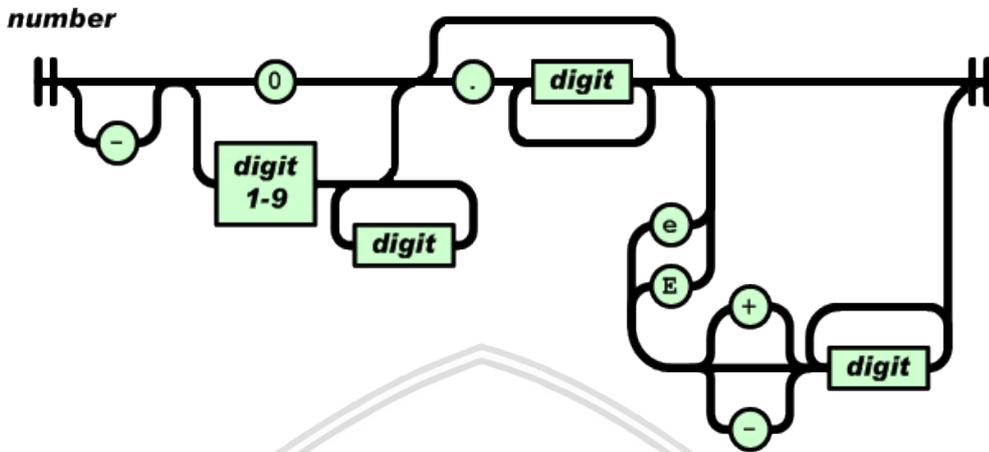


Gambar 2.6 Bentuk JSON String.

Sumber : (JSON, 2017)



*String* merupakan kumpulan dari nol karakter hingga karakter *Unicode* yang dibungkus dengan tanda kutip dua (“).



Gambar 2.7 Bentuk JSON Number.

*Number* disini mirip dengan angka pada bahasa C maupun java. Kecuali tidak dipergunakannya format oktal dan hexadesimal.

**2.1.7 Entity Relationship Diagram**

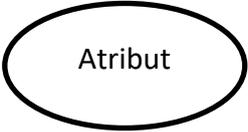
Pada diagram E-R (*Entity Relationship*) menurut Fthansyah (2002:79), adalah suatu komponen himpunan entitas serta himpunan relasi yang masing-masing dilengkapi dengan atribut-atribu yang merepresentasikan seluruh fakta dari dunia nyata yang kita tinjau. Diagram *Entity Relationship* pertama kali dideskripsikan oleh *peter chen* dalam bukunya yaitu “*The Entity Relationship Model Towar a Unified of Data*”. Pada buku tersebut Chen mencoba merumuskan dasar-dasar model setelah itu dikembangkan dan dimodifikasi oleh chen dan banyak pakar lain.

Pada penggunaan diagram E-R memiliki dua bentuk yaitu ada yang memiliki bentuk *physical* dan *conceptual*. ERD *Physical* merupakan penggambaran relasi maupun hubungan antar entitas yang telah digambarkan dalam bentuk fisik tabel-tabel serta komponen tabel tersebut. Sedangkan ERD *conceptual* merupakan penggambaran relasi atau hubungan antar komponen yang telah dilengkapi dengan kunci relasi, yang berguna untuk dihubungkan entitasnya dengan relasinya.

Berikut adalah beberapa simbol yang dipergunakan dalam ERD *Conceptual* :

Notasi	Keterangan
<div style="border: 1px solid black; width: 50px; height: 50px; margin: 0 auto; display: flex; align-items: center; justify-content: center;">                     Entitas                 </div>	<b>Entitas</b> , adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.



 <p>Relasi</p>	<p><b>Relasi</b>, menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.</p>
 <p>Atribut</p>	<p><b>Atribut</b>, berfungsi untuk mendeskripsikan karakter entitas ( atribut yang berfungsi sebagai key diberi garis bawah)</p>
	<p><b>Garis</b>, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.</p>

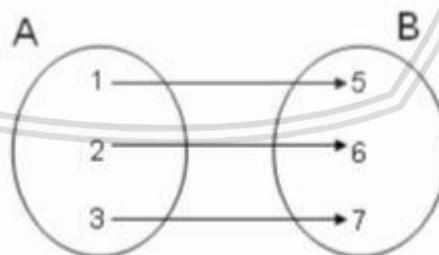
**Gambar 2.8** Simbol dalam *Entity Relationship Diagram*

**2.1.7.1 Kardinalitas/Derajat Relasi**

Kardinalitas merupakan relasi yang terjadi antara dua himpunan entitas (sebagai contoh A dan B) dapat dicontohkan ebagai berikut:

1. Satu ke Satu (*One to One*)

Bahwa setiap entitas pada himpunan entitas A berhubungan dengan paling banyak yaitu satu entitas pada himpunan entitas B begitu juga sebaliknya.



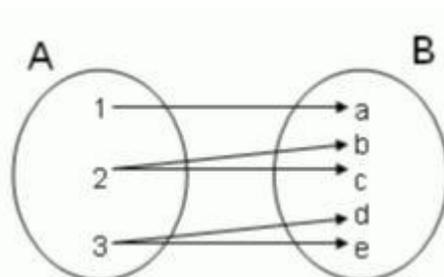
**Gambar 2.9** Contoh Kondisi Satu ke Satu

2. Satu ke Banyak (*One to Many*)

Berarti bahwa setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, akan tetapi tidak sebaliknya yang mana setiap entitas pada himpunan entitas



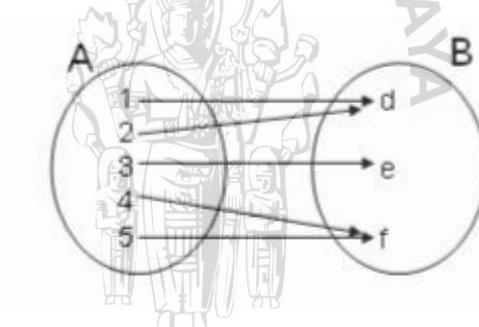
B dapat berhubungan dengan paling banyak satu entitas pada himpunan entitas A.



**Gambar 2.10 Contoh Kondisi Satu ke Banyak**

3. Banyak ke Satu (*Many to One*)

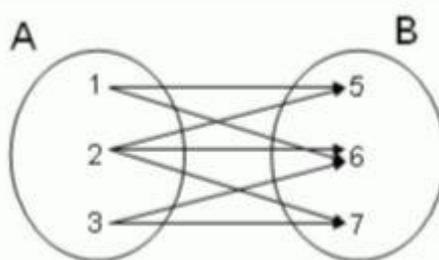
Berarti bahwa setiap entitas pada himpunan entitas A berhubungan dengan paling banyak yaitu satu entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.



**Gambar 2.11 Contoh Kondisi Banyak ke Banyak**

4. Banyak ke Banyak (*Many to Many*)

Memiliki arti bahwa pada setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada setiap himpunan entitas B begitu juga sebaliknya.



**Gambar 2.12 Contoh Kondisi Banyak ke Banyak**

**2.1.8 Android**

Android adalah sebuah sistem operasi yang dapat digunakan oleh *smartphone* atau bisa disebut telepon pintar yang berbasis linux. Karena berbasis linux, oleh karena itu android dapat dikatakan sebagai sistem operasi yang disebut *open source* atau *platform* terbuka. Sehingga *developer* dapat dengan mudah untuk membuat aplikasi-aplikasi *mobile* android tanpa mengalami gangguan hak cipta.

Android pertama kali di rilis yaitu pada tanggal 5 November 2007. Google membeli Android Inc yang merupakan pendatang baru dalam suatu perusahaan pembuatan perangkat lunak ponsel. Setelah google membeli Android Inc, google melakukan pembentukan *Open Handset Alliance* untuk mengembangkan android. Terdapat dua jenis distribut pada android, yang pertama adalah menggunakan *Google Mail Service* (GMS), sedangkan yang kedua tanpa dukungan Google dapat dikatakan dengan *Open Handset Distribution* (OHD) (android.com, 2017).

Berikut adalah tabel yang memperlihatkan versi android yang telah beredar setelah perilisian yaitu pada tahun 2007 (android.com)

**Tabel 2.2 Versi Android**

Versi	Nama	Tahun Rilis
Android 1.0	Beta	2008
Android 1.5	Cupcake	2009
Android 1.6	Donut	2009
Android 2.0	Eclair	2009
Android 2.2	Froyo	2010
Android 2.3	Gingerbread	2010
Android 3.0	Honeycomb	2011
Android 4.0	Ice Cream Sandwich	2011
Android 4.1	Jelly Bean	2012
Android 4.4	Kitkat	2013
Android 5.0	Lollipop	2014
Android 6.0	Marshmellow	2015
Android 7.0	Nougat	2016
Android 8.0	Oreo	2017

### 2.1.9 Android SDK

Android-SDK merupakan tools bagi para programmer yang ingin mengembangkan aplikasi berbasis google android. Android SDK mencakup seperangkat alat pengembangan yang komperhensif. Android SDK terdiri dari *debugger, libraries, handset emulator*, dokumentasi, contoh kode, dan tutorial. Saat ini android sudah mendukung arsitektur x86 pada Linux (distribusi linux apapun untuk desktop moderen), Max OS X 10.4.8 atau lebih, Windows XP atau Vista. Persyaratan mencakup JDK, Apache Ant dan Python 2.2 atau lebih baru. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan plugin Android Development Tools (ADT), dengan ini pengembangan dapat menggunakan teks editor untuk mengedit file Java dan XML serta menggunakan peralatan command line untuk menciptakan, membangun, melakukan debug aplikasi Android dan pengendalian perangkat Android (misalnya, reboot, menginstal paket perangkat lunak dengan jarak jauh).

Android SDK telah dirilis pada tanggal 12 November 2007. Dan pada tanggal 15 Juli 2008 tim Android Developer Challenge sengaja mengirimkan email ke semua pendatang di Android Developer Challenge untuk mengumumkan bahwa rilis SDK terbaru telah tersedia pada halaman download pribadi. Email tersebut juga ditujukan kepada pemenang Android Developer Challenge putaran pertama. Sebuah pernyataan bahwa Google telah menyediakan rilis SDK terbaru untuk beberapa pengembang dan bukan untuk orang lain (android.com).

### 2.1.10 Retrofit

Pada suatu arsitektur REST, REST server menyediakan *resources* yang dapat direpresentasikan dalam bentuk format teks yaitu JSON atau XML. JSON singkatan dari JavaScrip Object adalah suatu format ringkas pertukaran data yang sering digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang dapat disebut serialisasi dan dapat direpresentasikan oleh berbagai bahasa pemograman. *Resource* yang terbentuk yaitu format JSON yang disediakan REST *server* dapat dimangaatkan oleh suatu aplikasi android dengan *library retrofit*. *Retrofit* adalah *Rest Client* untuk android dan java dari *squareup* (Sutrisno, 2017).

### 2.1.11 QR Code ZXing

Barcode adalah suatu cara agar mesin/senso dapat membaca tanda dan menggunakan data yang ada didalamnya. Perkembangannya cukup banyak jenis yang dibuat. Jenis yang populer sekarang adalah *QR Code*. Kita dapat mendapatkan itu dimanapun, palagi kebanyakan pengguna *mobile* dapat menggunakan perangkatnya untuk melakukan *scanner qr code*.

Banyak *library* (.Net) untuk membaca dan membuat *qr code*. Salah satunya adalah *Zxing*. Walaupun masih *beta*, tapi hasil dari *library* bagus dan tersedia banyak *platform*. *Zxing.Net* adalah port dari *java project Zxing* yang bertujuan sama yaitu *barcode*. Dan *library* ini merupakan media *barcode* yang digunakan adalah *bitmap/image*. Jadi lebih bisa dikatakan berkerja dengan *Zxing.Net* adalah membaca dan membuat *barcode* lewat *image barcode* (Soeleman,2017).

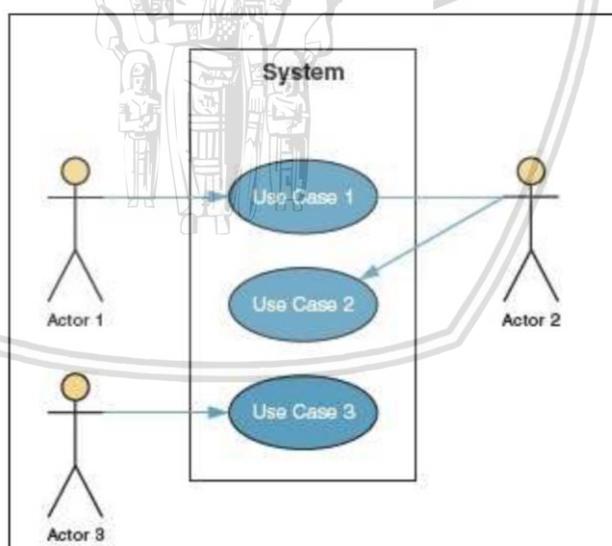
### 2.1.12 Unifed Modeling Language (UML)

*Unified Modifying Language* atau UMT merupakan suatu konvensi pemodelan yang digunakan untuk menspesifikasikan serta mendeskripsikan sebuah sistem piranti lunak yang terkait dengan suatu objek (Whitten & Bentley, 2007). UML dibagi menjadi beberapa tipe diagram, yaitu:

#### a. Use Case Diagram

*Use case diagram* merupakan suatu diagram yang menggambarkan interaksi antara sistem dengan pengguna (Whitten & Bentley, 2007). Bisa dikatakan bahwa diagram akan mendeskripsikan siapa yang akan menggunakan sistem tersebut dengan cara bagaimana pengguna berinteraksi dengan sistem.

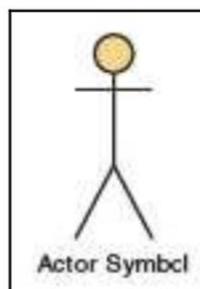
1. *Use Case* adalah suatu urutan langkah-langkah yang secara tindakan saling terkait, baik itu terotomatisasi maupun secara manual agar dapat melengjapi suatu tugas.



Gambar 2.13 Contoh Diagram *Use Case*

(Sumber: Whitten & Bentley, 2007)

2. Aktor merupakan sesuatu yang perlu berinteraksi dengan sistem untuk melakukan pertukaran informasi.



**Gambar 2.14 Simbol Aktor**

(Sumber: Whitten & Bentley)

3. Suatu hubungan atau disebut *relationship* pada diagram *use case* digambarkan sebagai sebuah garis antara dua simbol. Pemaknaan hubungan berbeda-beda tergantung bagaimana garis tersebut digambar dan tipe simbol apa yang digunakan untuk menghubungkan garis tersebut. Berikut adalah tipe hubungan yang dimiliki pada diagram *use case*:

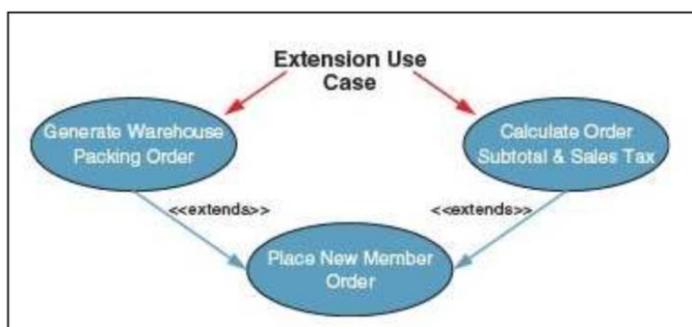
1) *Association* atau gabungan merupakan suatu hubungan antara pelaku dengan *use case*, dimana akan terjadi interaksi di antara mereka.



**Gambar 2.15 Contoh Hubungan Association**

(Sumber: Whitten & Bentley, 2007)

2) *Extends*: merupakan *use case* yang terdiri dari langkah yang diekstraksi dari *use case* yang lebih kompleks agar dapat menyederhanakan masalah yang original dan untuk memperluas fungsinya.

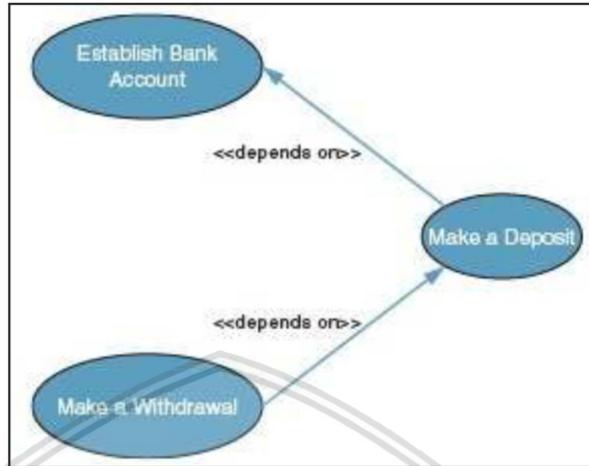


**Gambar 2.16 Contoh Hubungan Extends**

(Sumber: Whitten & Bentley, 2007)



- 3) *Depends On*: merupakan *use case* yang memiliki suatu ketergantungan pada suatu *use case* lain untuk menetapkan rangkaian *use case* yang dapat dikembangkan.

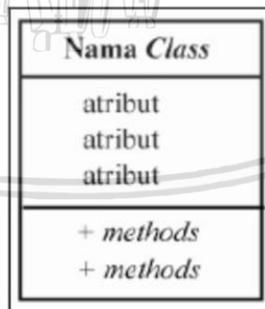


**Gambar 2.17** Contoh Hubungan *Depends On*

b. *Class Diagram*

*Class Diagram* menggambarkan suatu struktur objek yang terdapat pada suatu sistem (Whitten & Bentley, 2007). Diagram ini akan menunjukkan suatu objek yang terdapat pada sistem serta relasi antar objek.

- 1) *Class* digambarkan sebagai sebuah kotak yang dibagi atas tiga bagian. Pertama adalah bagian atas yaitu nama dari *class*. Kedua adalah bagian tengah yang mendefinisikan atribut *class*. Ketiga yaitu bagian akhir mendefinisikan *method* dari suatu *class*.

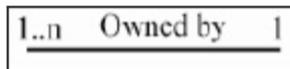


**Gambar 2.18** *Class*

(Sumber: Whitten & Bentley, 2007)

- 2) *Association* adalah sebuah hubungan yang paling umum antara dua *class* serta dilambangkan oleh sebuah garis yang menghubungkan dua *class* tersebut. Contoh hubungan tersebut adalah *one-to-one*, *one-to-many*, dan *many-to-many*.





**Gambar 2.19 Association**

(Sumber: Whitten & Bentley, 2007)

- 3) *Composition* adalah apabila sebuah *class* tidak dapat berdiri sendiri erta harus merupakan bagian dari *class* yang lain, maka *class* tersebut memiliki relasi *composition* terhadap *class* tempat dia bergantung. Suatu hubungan *composition* digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi.



**Gambar 2.20 Composition**

(Sumber: Whitten & Bentley, 2007)

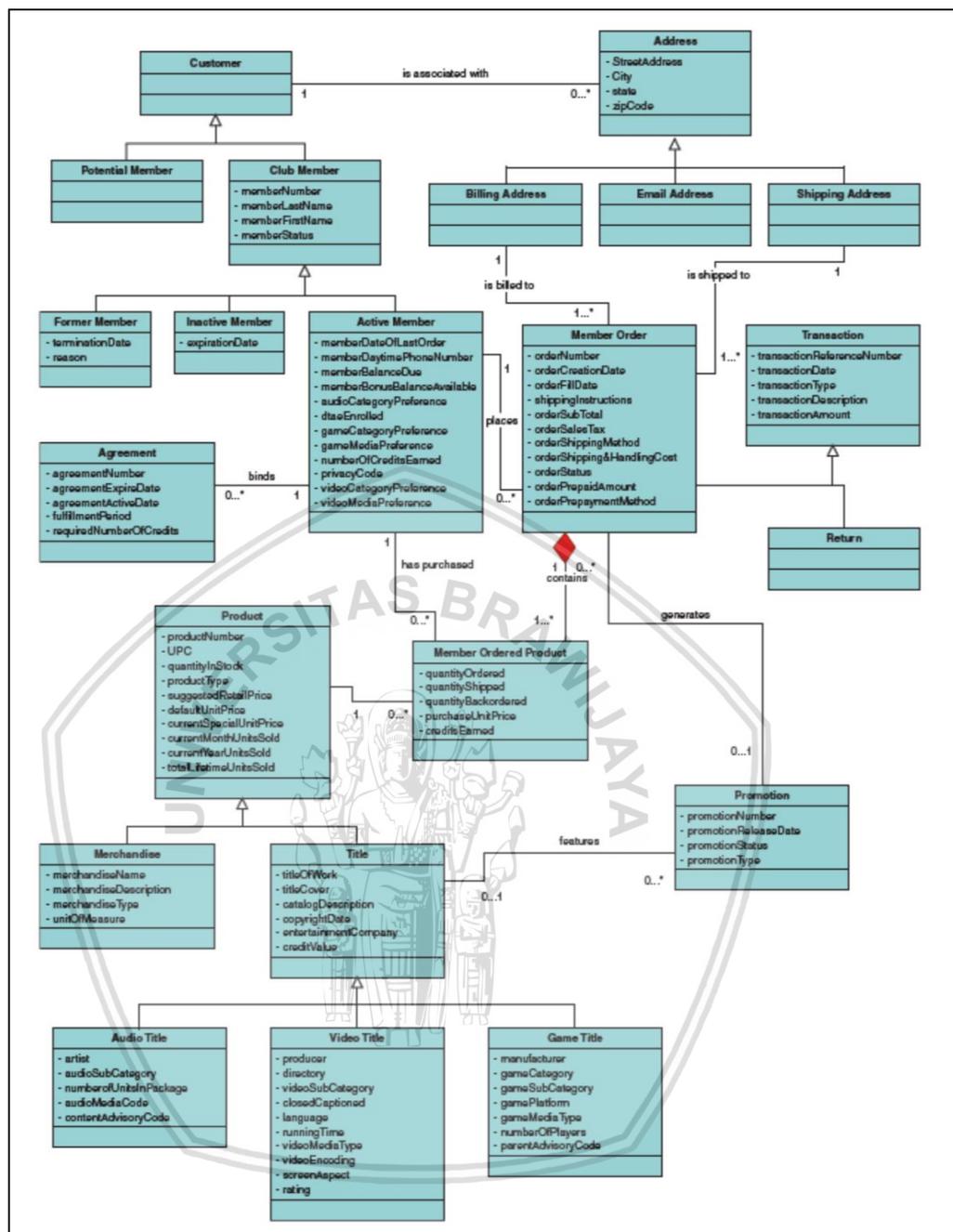
- 4) *Agregation* adalah apabila sebuah *class* dapat berdiri sendiri walaupun bagian dari *class* yang lain.



**Gambar 2.21 Agregation**

(Sumber: Whitten & Bentley, 2007)

- 5) Bagaimana cara untuk mengakses *veasibility* UML menyediakan tiga tingkat dari *visibility*, yaitu:
1. *Public* : ditandai dengan menggunakan simbol “+”
  2. *Protected* : ditandai dengan menggunakan simbol “#”
  3. *Private* : ditandai dengan menggunakan simbol “-”



Gambar 2.22 Contoh Class Diagram

(Sumber : Whitten & Bentley, 2007)

c. Activity Diagram

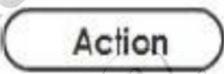
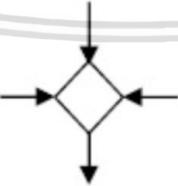
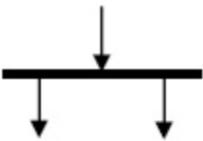
Activity diagram merupakan sebuah diagram yang dapat dipakai untuk mendeskripsikan secara grafis suatu aliran kerja dari sebuah proses bisnis. Langkah-langkah dari suatu use case, atau suatu logika dari sebuah perilaku objek atau object behavior/method (Whitten & Bentley, 2007).



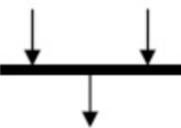
Activity diagram adalah merupakan representasi grafis dari aliran kerja (*workflow*) dari aktivitas-aktivitas yang bertahap dan aksi-aksi di dalam suatu sistem.

Diagram ini juga didukung proses seleksi, pengulangan (iterasi), dan konkurensi. *Activity* diagram dikatakan sejenis dengan state diagram/*statechart* diagram. Diagram ini akan mendeskripsikan kumpulan aktivitas dengan menunjukkan sekuensial dari aktivitas yang dilakukan. Diagram ini juga bisa menunjukkan suatu aktivitas yang bersifat kondisional atau paralel. Berikut adalah notasi-notasi yang dapat digunakan pada *activity diagram* :

**Tabel 2.3 Simbol Pada Activity Diagram**

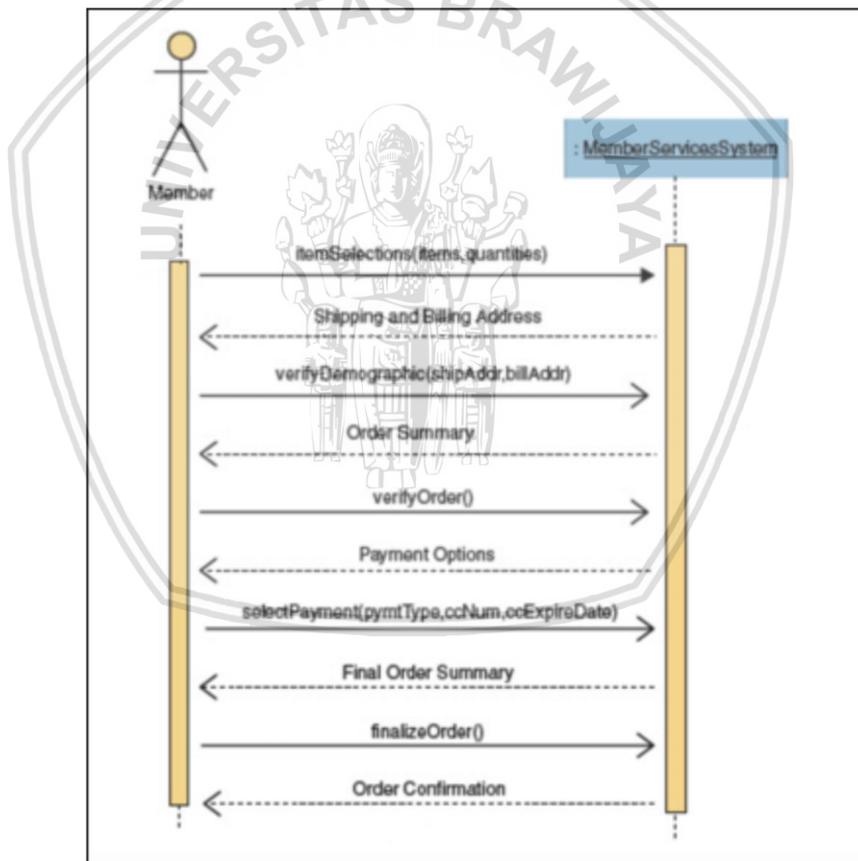
No	Simbol	Gambar	Keterangan
1	<i>Initial Node</i>		Merupakan awal dari proses
2	<i>Action</i>		Menunjukkan aktifitas yang terjadi
3	<i>Flow</i>		Untuk berlanjut ke aksi yang lain dengan dilengkapi kata-kata bila berasal dari <i>decision</i>
4	<i>Decision</i>		Memiliki satu buah <i>flow</i> masuk dan dua atau lebih <i>flow</i> keluar menunjukkan bahwa lebih dari satu kemungkinan terjadi
5	<i>Merge</i>		Menyatukan <i>flow</i> yang sebelumnya terpisah oleh <i>decision</i>
6	<i>Fork</i>		Menunjukkan aksi atau proses paralel yang dapat terjadi secara bersamaan.



7	Join		Semua aksi yang masuk ke join harus telah diselesaikan sebelum proses berlanjut
8	Final Activity		Merupakan akhir dari proses

d. *Sequence Diagram*

*Sequence diagram* menggambarkan bagaimana suatu objek-objek saling berinteraksi melalui pesan dalam eksekusi dari sebuah *use case* atau suatu operasi (Whitten & Bentley, 2007). *Sequence diagram* membantu untuk mengidentifikasi pesan tingkat tinggi yang masuk dan keluar dari sistem.



**Gambar 2.23 Contoh *Sequence Diagram***

(Sumber: Whitten & Bentley, 2007)

Beberapa notasi yang terdapat di dalam system sequence diagram, yaitu:

1. *Actor*, adalah aktor yang dapat memulai pada *use case* ditunjukkan dengan menggunakan simbol *use case* aktor.



2. *System*, berupa suatu kotak yang menunjukkan sistem yang sedang berjalan dengan ditandai oleh notasi titik dua (:) pada awal nama sistem.
3. *Lifelines*, merupakan garis putus-putus menurun dari simbol aktor dan sistem.
4. *Activation bars*, berupa sebuah balok yang terletak diatas *lifelines*, dan menunjukkan periode waktu ketika terdapat interaksi secara aktif.
5. *Input messages*, adalah berupa anak panah mendatar dari sesuatu aktor ke sistem yang menunjukkan pesan masuk. Kata pertama diawali dengan huruf kecil, dan kata berikutnya diawali dengan huruf besar dan tidak ada spasi. Terakhir diikuti dengan tanda kurung yang berisi parameter yang dibutuhkan.
6. *Output messages*, berupa anak panah mendatar dengan garis putus-putus dari sistem ke aktor yang menunjukkan pesan keluar.
7. *Receiver Actor*, adalah aktor lain yang dapat menerima pesan dari sistem.
8. *Frame*, berupa sebuah kotak yang menambahkan pesan terpisah untuk menunjukkan perulangan, *alternative*, atau pilihan.

### 2.1.13 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah suatu tahapan dari pengembangan perangkat lunak agar dapat diuji kesesuaian perangkat lunak yang telah dibangun dengan spesifikasi kebutuhan perangkat lunak. Dengan pengujian perangkat lunak, pengembangan dapat mengetahui permasalahan yang terjadi pada perangkat lunak dan cara penanganannya (Kolawa, 2007).

#### 2.1.13.1 Kasus Uji

Kasus Uji atau disebut *test case* merupakan kasus pengujian yang memiliki suatu kemungkinan tinggi dalam menemukan kesalahan yang sebelumnya tidak ditemukan. Sebelum dilakukan pengujian, akan dibuat rancangan *test case* agar proses pengujian lebih efisien dari segi waktu dan usaha (Rouf, 2012). Kasus uji dilakukan dengan menggunakan metode pengujian yang sesuai.

#### 2.1.13.2 Pengujian Unit

Merupakan pengujian yang cara mengujinya menggunakan komponen individu dari kode sumber, kumpulan modul program dengan data kontrol yang terasosiasi, prosedur penggunaan, dan juga prosedur pelaksanaan untuk menguji kelayakannya untuk digunakan oleh aplikasi dalam menjalankan fungsi-fungsi yang dimiliki.

#### 2.1.13.3 Pengujian Validasi

Pengujian Validasi merupakan suatu pengujian yang dapat dilakukan untuk menguji kesesuaian keluaran dari sistem dengan kebutuhan spesifikasi kebutuhan sistem. Keberhasilan pengujian validasi diukur dengan seberapa banyak fungsi yang dikenali pengguna.

**2.1.13.4 Black Box Testing**

Merupakan suatu metode pengujian yang disajikan berdasarkan deskripsi eksternal pada perangkat lunak, termasuk spesifikasi kebutuhan sistem, serta perancangannya (Amman & Offutt, 2008). Pada metode pengujian ini akan dilakukan untuk menguji kesesuaian fungsi pada perangkat lunak dengan kebutuhan yang telah didefinisikan, sehingga metode ini cocok digunakan dalam pengujian validasi.

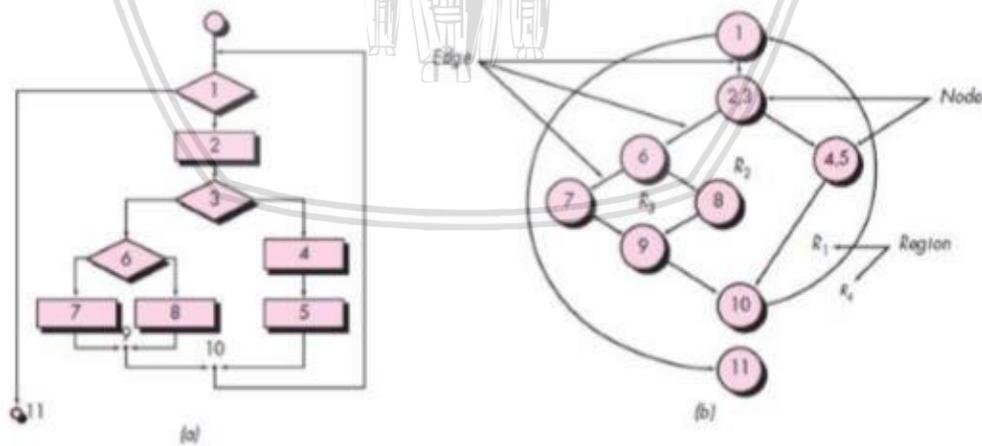
**2.1.13.5 White Box Testing**

Merupakan suatu metode pengujian yang dapat dilakukan terhadap internal dari kode sumber, yaitu pada pernyataan, percabangan, dan kondisi individual (Amman & Offutt, 2008). Pada metode *white box testing* memiliki tujuan yaitu untuk menjabarkan cara kerja terhadap kode sumber perangkat lunak yang dibangun secara rini dan juga metode ini cocok digunakan pada pengujian unit.

**2.1.13.6 Basis Path Testing**

*Basis Path testing* adalah teknik yang dilakukan dalam *white box testing* untuk menghasilkan semua kemungkinan kasus uji yang terjadi pada sebuah unit. Pada implementasinya, cara teknik ini menggunakan metode *cyclomatic complexity* milik Thomas J. McCabe agar dapat menentukan jalur independen, yang kemudian digunakan agar dapat menghasilkan kasus uji (Westfall, 2008)

Disini akan ditampilkan contoh penghitungan *cyclomatic complexity* berdasarkan *flowgraph* yang dihasilkan pada *flowchart* program Gambar dibawah ini.



**Gambar 2.24 Transformasi Flowchart (a) ke Flowgraph (b)**

Sumber: (Pressman, 2010)

1.  $V(G) = E - N + 2$  dimana E = jumlah *edge* dan N = jumlah *node*  
 $V(G) = 11 - 9 + 2$   
 $V(G) = 4$

2.  $V(G) = P+1$  dimana  $P = \text{predicate node}$  pada graf

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

$$V(G) = 4$$

3.  $V(G) = R$  dimana  $R = \text{jumlah region}$  pada graf

$$V(G) = 4$$

### 2.1.13.7 Usability Test

Usability Test digunakan untuk menguji kegunaan aplikasi yang dibuat telah sesuai dengan yang diinginkan atau tidak. Nilai Usability bisa didapatkan dengan System Usability Scale (SUS). SUS memiliki kelebihan yaitu cara ini lebih murah karena tidak memerlukan banyak resource dan lebih cepat karena telah tersedia template kuesioner sehingga lebih cepat dari pada membuat kuesioner baru (Jaff Sauro, 2011). Tabel 2.4 Menunjukkan contoh template dari SUS.

**Tabel 2.4 Template Kuisisioner SUS**

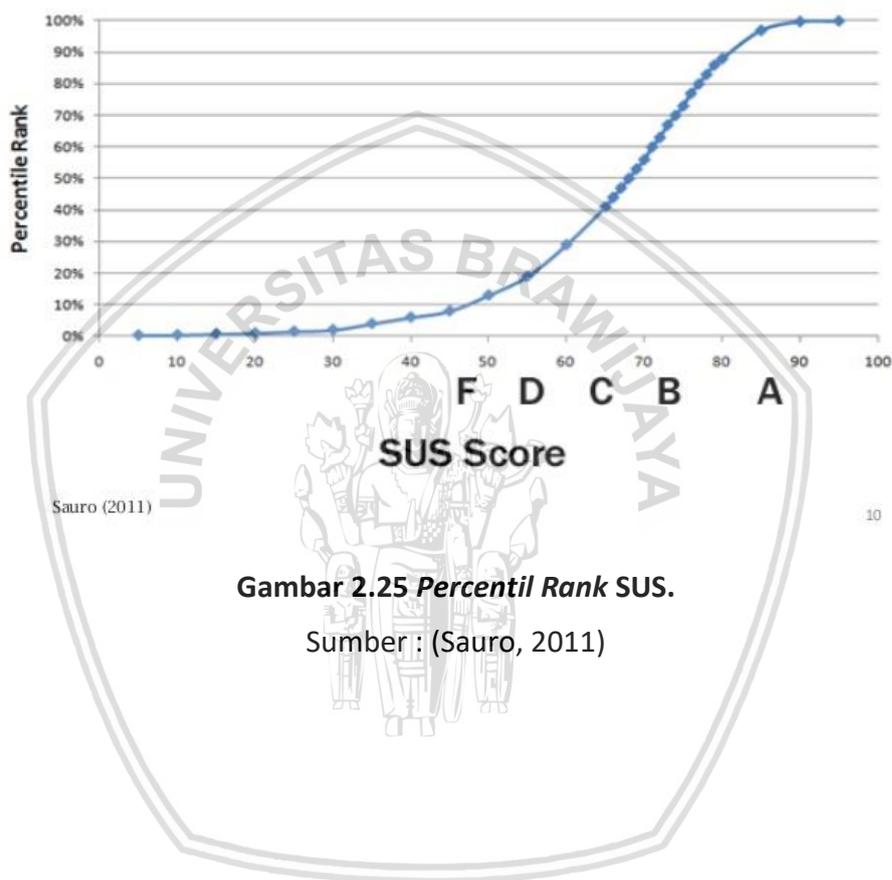
No	Template Question
1	I think that I would like to use this system frequently.
2	I found the system unnecessarily complex.
3	I thought the system was easy to use.
4	I think that I would need the support of a technical person to be able to use this system.
5	I found the various functions in this system were well integrated.
6	I thought there was too much inconsistency in this system.
7	I would imagine that most people would learn to use this system very quickly.
8	I found the system very cumbersome to use.
9	I felt very confident using the system.
10	I needed to learn a lot of things before I could get going with this system.

Sumber : (usabilitygeek : 2015)

SUS dihitung dengan memberikan skala satu hingga lima pada setiap pertanyaan. Untuk mendapatkan nilai akhir maka nilai pada pertanyaan dengan nomor ganjil akan dikurangi satu, dan untuk nomor genap maka diberi nilai lima kemudian dikurangi dengan nilai dari tes. Kemudian dijumlahkan antara nomor ganjil dan genap selanjutnya dikalikan dengan angka 2,5. Hasil sempurna dari SUS adalah 100 (Thomas, 2015).

Nilai rata-rata dari SUS adalah 68. jika nilai yang di dapatkan dibawah 68 maka dapat dikatakan ada masalah pada aplikasi yang dibuat sehingga membuat usability tidak optimal. Pada Gambar 2.3 ditunjukkan grafik bagaimana percentil rank nilai rata-rata SUS yang dikaitkan dengan huruf (Sauro, 2011).

## Grading on a Curve

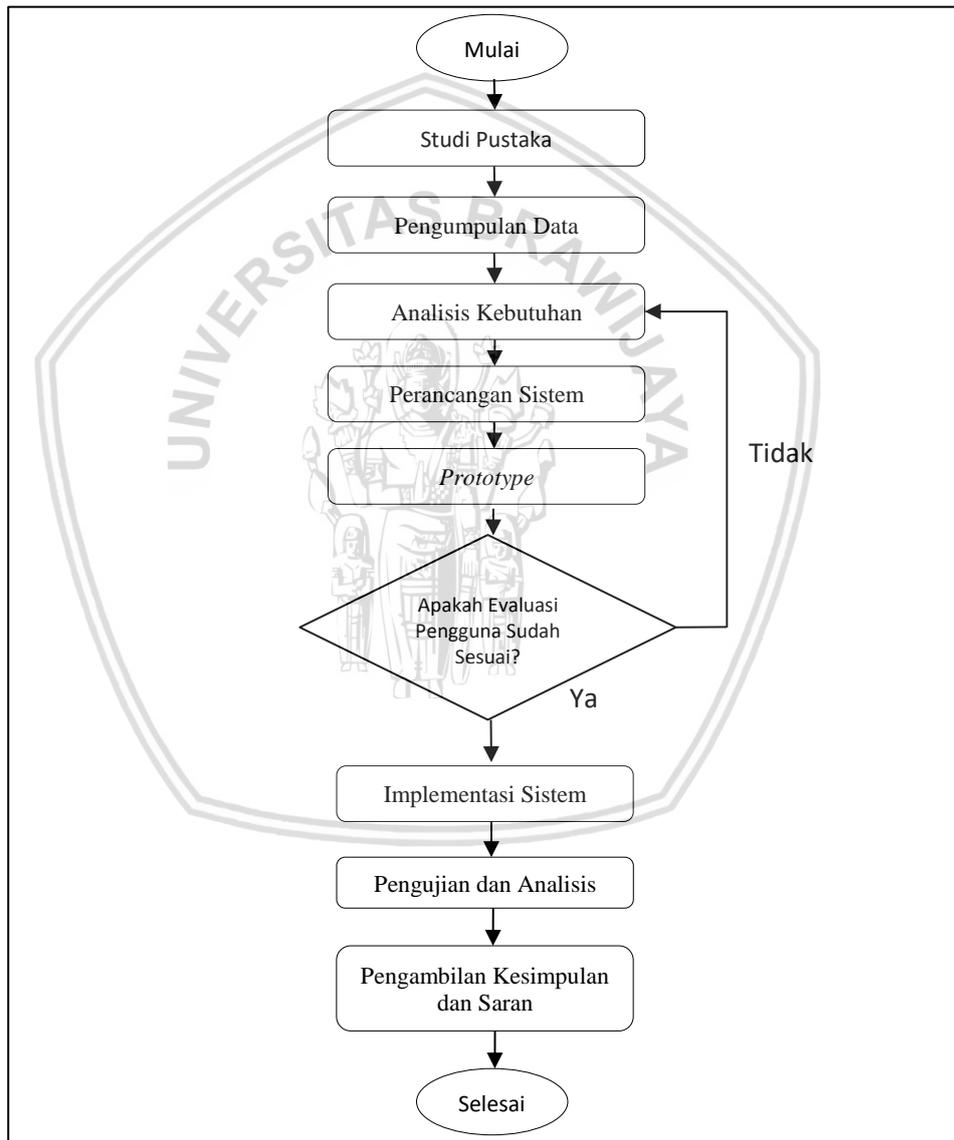


**Gambar 2.25 Percentil Rank SUS.**

Sumber : (Sauro, 2011)

### BAB 3 METODOLOGI

Bab ini menjelaskan mengenai metode penelitian yang dilakukan dalam pengembangan aplikasi V-Apps untuk menginformasikan jadwal damping mahasiswa difabel. Metodologi yang digunakan dalam penelitian ini mengadopsi SDLC *prototyping* dengan dua kali iterasi. Beberapa hal dalam metodologi penelitian yaitu tahapan penelitian, studi pustaka, pengumpulan data, analisis kebutuhan sistem, perancangan sistem, pembuatan *prototype*, evaluasi pengguna, implementasi sistem, pengujian dan analisis, pengambilan kesimpulan dan saran.



Gambar 3.1 Diagram Metodologi Penelitian.

### 3.1 Studi Pustaka

Studi pustaka menjelaskan dasar teori yang digunakan didapat dari penelitian yang pernah dilakukan sebelumnya, buku, *ebook*, *journal*, dan beberapa pustaka dari internet untuk menunjang penulisan tugas akhir. Dasar teori yang mendukung dalam penulisan skripsi ini adalah sebagai berikut:

1. Pusat Studi dan Layanan Disabilitas Universitas Brawijaya
2. Pendamping
3. Mahasiswa Difabel
4. Rekayasa Perangkat Lunak
5. *System Development Life Cycle (SDLC)*
  - a. *Prototyping*
6. *JavaScript Object Notation (JSON)*
7. *Entity Relathionship Diagram*
8. *Unifed Modeling Language (UML)*
9. *Retrofit*
10. *QR Code*
11. *White Box Testing*
12. *Black Box Testing*
13. *Usability Testing*

### 3.2 Pengumpulan Data

Pada penelitian ini data diperoleh dari sistem penjadwalan jadwal damping Pusat Studi dan Layanan Disabilitas Universitas Brawijaya. Data yang digunakan berupa jadwal damping yang ada di kantor pelayanan PSLD, data yang didapat akan digunakan untuk ditampilkan di aplikasi. Data yang diambil selanjutnya adalah data diri dari pendamping dan juga mahasiswa difabel, data ini akan digunakan untuk keperluan pengguna di aplikasi V-Appls. Penelitian skripsi ini menggunakan metode *survey*. Pada survey kali ini digunakan wawancara dan kuisisioner, alasan digunakannya wawancara adalah karena agar dapat mengetahui lebih dalam bagaimana PSLD UB menyampaikan informasi jadwal damping, dan bagaimana pendamping dan mahasiswa difabel menerima informasi tersebut. Wawancara ditujukan kepada 1 staff PSLD UB, 1 pendamping, dan 1 mahasiswa difabel. Kuisisioner digunakan untuk mendapatkan kebutuhan Aplikasi V-Appls dalam menampilkan informasi jadwal pendampingan. Kuisisioner ditujukan kepada 15 pendamping dan 15 mahasiswa difabel.

### 3.3 Analisis Kebutuhan

Analisis kebutuhan adalah proses tahapan bagaimana mendefinisikan kebutuhan yang harus dipenuhi sistem baik itu kebutuhan fungsional maupun non fungsional agar aplikasi sesuai dengan apa yang diinginkan pengguna. Analisis kebutuhan yang dilakukan meliputi penjabaran secara umum mengenai sistem atau aplikasi yang dibangun, mengidentifikasi aktor, mengidentifikasi kebutuhan fungsional aktor dan kebutuhan non fungsional. Berdasarkan kebutuhan fungsional yang telah diidentifikasi, dapat dibuat pemodelan kebutuhan dengan menggunakan salah satu diagram UML yaitu *use case diagram*. Tahapan setelah membuat *use case diagram* akan dilanjutkan dengan pembuatan *use case scenario* untuk menjelaskan cara kerja sistem.

Pada proses analisis kebutuhan aplikasi V-Apps akan dilakukan dua kali iterasi, iterasi pertama bertujuan untuk mendapatkan spesifikasi kebutuhan dari pengguna berdasarkan wawancara. Wawancara akan dilakukan kepada 1 staff PSD UB bagian administrasi pendampingan, 1 pendamping, dan 1 mahasiswa difabel. Hasil dari spesifikasi kebutuhan adalah daftar kebutuhan yang harus dipenuhi sistem. Sedangkan untuk iterasi kedua yaitu mengevaluasi kebutuhan pengguna berdasarkan hasil dari tahap evaluasi pengguna. Jika dalam tahap evaluasi pengguna terdapat kesalahan pada kebutuhan fungsional, maka kebutuhan tersebut akan dirubah atau dilakukan analisis ulang pada iterasi kedua. Apabila pada tahap evaluasi pengguna iterasi pertama tidak ada keluhan, maka proses akan berlanjut ke tahap implementasi. Setelah iterasi kedua, maka pada tahap evaluasi pengguna akan berlanjut ke tahap implementasi.

Setelah kebutuhan didapatkan, selanjutnya menuliskan data hasil pengamatan tersebut pada diagram *use case*. Diagram *use case* digunakan digunakan untuk mengetahui relasi atau hubungan aktor serta kebutuhan fungsionalnya. Dilanjutkan dengan pembuatan *use case scenario* untuk mendeskripsikan *scenario* dari setiap *use case* yang sudah dibuat. Analisis kebutuhan ini dirancang untuk mempermudah dalam mengetahui kebutuhan pengguna dan menjadi dasar dalam perancangan perangkat lunak. Analisis kebutuhan dapat digunakan sebagai referensi dalam melakukan validasi kebutuhan saat pengujian perangkat lunak.

### 3.4 Perancangan Sistem

Perancangan sistem adalah merupakan proses bagaimana memperoleh desain sistem yang menunjukkan alur kerja sistem. Perancangan sistem dilakukan apabila analisis kebutuhan telah diselesaikan, sehingga semua kebutuhan dapat digambarkan dengan jelas dan terdefinisi. Analisis dilakukan dengan dua kali iterasi, iterasi pertama merupakan rancangan yang didapat dari hasil analisis kebutuhan awal. Iterasi kedua merupakan rancangan yang didapat dari hasil analisis kebutuhan yang sudah di evaluasi pengguna. Berikut tahapan - tahapan perancangan dalam membangun sistem ini:

#### 1. Perancangan Arsitektur

Dalam perancangan arsitektur ini akan dilakukan pemodelan dengan menggunakan diagram UML seperti *sequence* diagram dan class diagram. *Sequence diagram* yang dibuat memiliki batasan hanya lima *sequence* yang diperoleh dari hasil analisis kebutuhan.

#### 2. Perancangan Komponen

Perancangan komponen menggambarkan rincian sub-sistem dari setiap komponen perangkat lunak. Dalam perancangan komponen ini akan dituliskan dua sampel algoritme-algoritme utama yang diambil dari setiap kelas pada *controller*. Algoritme ini akan ditulis dalam bentuk *pseudocode*.

#### 3. Perancangan Data

Dalam perancangan basis data ini menggunakan ERD (*Entity Relational Diagram*) yang didapat dari sistem penjadwalan jadwal damping PSDL UB. Perancangan data ini akan menjadi dasar dalam implementasi *database* pada sistem. Serta dituliskan API yang dapat digunakan pada aplikasi V-Apps.

#### 4. Perancangan Antarmuka

Perancangan antarmuka akan ditunjukkan bagaimana rancangan antarmuka aplikasi yang ditampilkan dapat mempermudah interaksi antara pengguna dan sistem. Dalam perancangan antarmuka ini akan dituliskan beberapa sampel antarmuka utama. Perancangan antarmuka ini akan menjadi dasar dari implementasi antarmuka sistem.

### 3.5 Pembuatan *Prototype*

Setelah proses perancangan sistem selesai dilaksanakan, proses selanjutnya adalah pembuatan *prototype*. *Prototype* yang dibuat memiliki batasan hanya tampilan *User Interface* (UI) serta beberapa fitur yang menggambarkan sistem. Beberapa fitur yang menggambarkan sistem adalah fitur melihat jadwal damping. Akan tetapi fitur dalam *prototype* ini hanya *prototype* yang belum dapat memproses data, hanya menampilkan data. Tujuan dari *prototype* ini adalah mencari kekurangan atau kepuasan dari analisis kebutuhan yang telah didapatkan sebelumnya.

*Prototype* pertama (*Prototype 1*) adalah *portotype* yang telah dibuat menggunakan *Marvelapp* dengan UI sesuai dengan perancangan serta data telah dimasukkan sebanyak 30%. *Prototype* kedua (*Prototype 2*) merupakan *prototype* hasil evaluasi pengguna. *Prortotype* ini disesuaikan dengan hasil evaluasi pengguna, sehingga apabila ada fitur yang berubah pada evaluasi pengguna, maka pada *prototype* ini juga berubah baik itu UI, UX, serta data. Jumlah data pada *prototype* ini tetap berada pada 30%.

### 3.6 Evaluasi Pengguna

Setelah proses pembuatan *prototype*, kemudian dilakukanlah evaluasi pengguna. *Testing* akan melibatkan perwakilan pengguna dengan batasan jumlah 5 pendamping dan 5 mahasiswa difabel. Pengguna akan mengerjakan

tugas-tugas khusus setelah menggunakan *prototype*. Tahap evaluasi pengguna ini bertujuan untuk mengevaluasi hasil dari penggunaan *prototype* pada proses sebelumnya. Jika pengguna telah puas dengan *prototype* maka akan dilanjutkan ke tahap implementasi sistem, jika pada tahap *prototype* pengguna masih merasa kurang puas, maka akan dilakukan analisis kebutuhan kembali sesuai kebutuhan pengguna.

Pada tahap ini dilakukan iterasi *prototype* sebesar dua kali. Tahap pertama untuk menemukan kekurangan pada sistem dan mengavaluasinya, jika pada tahap pertama pengguna telah dengan sistem, maka akan berlanjut ke tahap implementasi. Tahap kedua evaluasi digunakan untuk menilai hasil evaluasi tahap pertama, dan hasil evaluasi tahap kedua akan dibawa ke tahap implementasi.

### 3.7 Implementasi Sistem

Implementasi sistem adalah fase pembangunan sistem yang mengacu pada perancangan sistem. Implementasi sistem ini dibangun diatas *framework android* dengan menggunakan bahasa Java. Proses yang ada dalam implementasi sistem ini antara lain:

1. Implementasi sistem dilakukan dengan mengacu pada spesifikasi sistem yang digunakan untuk membangun aplikasi ini.
2. Implementasi basis data dilakukan berdasarkan dari hasil rancangan yang dibuat. Yaitu implementasi *database* dari sistem penjadwalan PSLD UB dan Hasil dari API yang digunakan oleh aplikasi.
3. Implementasi kode program mengacu pada perancangan komponen yang memuat algoritme-algoritme pada perancangan sistem. Logika-logika program tersebut di implementasikan menggunakan bahasa pemrograman Java.
4. Implementasi antarmuka, Implementasi ini dibuat berdasarkan perancangan antarmuka pada perancangan sistem dan juga mengacu pada *prototype* yang telah dibuat. Implementasi ini menggunakan bahasa pemrograman Java.

### 3.8 Pengujian dan Analisis

Setelah proses implementasi dilakukan tahap selanjutnya adalah proses pengujian. Tujuan dari pengujian sistem adalah untuk mengetahui apakah sistem telah berjalan sesuai dengan yang diharapkan atau tidak. Jika terdapat suatu kesalahan maka proses analisis dilakukan untuk menganalisis kesalahan tersebut untuk dilakukan perbaikan aplikasi agar sesuai dengan kebutuhan yang dilakukan. Strategi pengujian sistem yang akan dibangun ini mempunyai beberapa tahapan yang harus dilakukan yaitu sebagai berikut:

#### 1. Unit Testing

Pengujian unit menggunakan proses pengujian dengan *white box testing*. *White box testing* yang digunakan adalah *basis path testing*.

Dengan *basis path testing* akan mendapatkan *flow graph* yang merupakan hasil dari kode program. Berdasarkan *flow graph* akan dibuat *cyclomatic complexity* serta himpunan basis dari jalur independen atau jalur dasar. Pengujian ini akan dilakukan dengan tiga sampel uji.

## 2. *Validation Testing*

Pengujian validasi menggunakan *black box testing*. Dalam pengujian ini akan dilakukan skenario menjalankan semua fungsi yang ada pada sistem atau aplikasi. Fungsi disini merupakan fungsi yang terdapat pada *use case diagram*. Semua fungsi akan diuji apakah hasil dari pengujian tersebut menghasilkan hasil yang sesuai dengan kebutuhan fungsional atau non fungsional atau tidak sama sekali. Hasil tersebut akan dicatat pada tabel untuk diketahui apakah fungsi tersebut *valid* atau tidak *valid*.

## 3. *Usability Testing*

Pengujian ini dilakukan untuk memastikan bahwa sistem yang dibangun sudah sesuai serta dapat digunakan dengan mudah, mudah dipahami oleh *user* dan mudah untuk dioperasikan oleh *user* nantinya. Pengujian ini dilakukan ketika sistem sudah siap digunakan dan digunakan oleh pengguna sehingga pengguna dapat merasakan akan kemudahan menggunakan sistem dan juga sistem mudah dipelajari dan dipahami.

### 3.9 Pengambilan Kesimpulan dan Saran

Proses pengambilan kesimpulan dan saran dilakukan pada tahap terakhir. Pengambilan kesimpulan dilakukan berdasarkan hasil pengujian dan analisis yang telah dilakukan sebelumnya. Tahap terakhir dari pembuatan dokumen ini adalah pengambilan saran. Saran ditulis dengan tujuan untuk memperbaiki kesalahan yang ada baik dalam penulisan dokumen maupun program aplikasi serta memberikan pertimbangan dalam pengembangan perangkat lunak kedepannya.

## BAB 4 ANALISIS KEBUTUHAN

Bab ini berisi analisis kebutuhan dalam membangun aplikasi V-Apps sebagai aplikasi untuk menginformasikan jadwal damping mahasiswa difabel. Pada tahap analisis kebutuhan ini terdiri dari empat tahap, yaitu gambaran umum sistem, identifikasi aktor, identifikasi kebutuhan, dan analisis data. Dari tahap – tahap tersebut akan dijelaskan bagaimana sistem ini akan berjalan, mendeskripsikan aktor serta apa yang dapat dilakukan, pengumpulan data serta analisis data, identifikasi kebutuhan pengguna yaitu kebutuhan fungsional dan kebutuhan non fungsional.

### 4.1 Gambaran Umum Sistem

Aplikasi *Volunteer Apps* (V-Apps) merupakan aplikasi yang dibuat dengan tujuan untuk memberikan informasi jadwal pendampingan kepada pendamping dan mahasiswa difabel. Pendamping dan mahasiswa difabel dapat melihat jadwal damping serta status jadwal mereka. Apabila status berwarna biru menandakan terjadwal, kuning menandakan dikonfirmasi, hijau menandakan selesai. Jadwal damping yang ditampilkan adalah jadwal damping yang sudah di jadwalkan oleh staff Administrasi Pendampingan PSDL UB melalui sistem penjadwalan secara otomatis. Aplikasi V-Apps ini juga dapat memberikan pendamping mengkonfirmasi kehadirannya serta mahasiswa difabel dapat memberikan *review* kepada pendamping yang telah melakukan konfirmasi kehadiran. Aplikasi V-Apps juga memberikan fitur untuk telpon dan sms, sehingga antara pendamping dan mahasiswa difabel bisa saling berkomunikasi.

### 4.2 Identifikasi Aktor

Pada tahap ini ditunjukkan aktor yang akan berinteraksi dengan sistem. Aktor berguna untuk memberikan gambaran tentang apa yang dapat dilakukan aktor dalam menggunakan sistem. Aktor dalam aplikasi ini ditunjukkan pada tabel 4.1.

**Tabel 4. 1 Identifikasi Aktor**

Aktor	Deskripsi
Pengguna Umum	Pengguna umum adalah aktor yang belum terautentifikasi oleh sistem sehingga tidak dapat mengakses fitur yang ada pada aplikasi.
Pendamping	Pendamping adalah aktor yang sudah terautentifikasi oleh sistem sebagai pendamping sehingga dapat mengakses fitur yang ada pada aplikasi. Salah satu fitur utama pendamping yaitu melihat jadwal

	damping sehingga pendamping tahu mahasiswa difabel yang akan pendamping dampingi.
Mahasiswa Difabel	Mahasiswa difabel adalah aktor yang sudah terautentifikasi oleh sistem sebagai mahasiswa difabel sehingga dapat mengakses fitur yang ada pada aplikasi. Salah satu fitur utama mahasiswa difabel yaitu melihat pendamping yang akan mendampingi mahasiswa difabel saat jam perkuliahan.

### 4.3 Identifikasi Kebutuhan

*Software Requirement Specification* (SRS) digunakan untuk proses identifikasi kebutuhan. Fungsi dari SRS yaitu menampung daftar dari kebutuhan sistem agar mudah dipahami. Kebutuhan yang ditampung oleh SRS yaitu kebutuhan fungsional dan kebutuhan non fungsional. Setiap kebutuhan akan diberikan kode VAP-F-X-YY untuk kebutuhan fungsional dan VAP-NF-X-YY untuk kebutuhan non fungsional. VAP merupakan singkatan dari *Volunteer Apps*, F untuk kebutuhan fungsional, NF untuk kebutuhan non fungsional, X sebagai kode aktor, dan YY sebagai nomor kode fungsi.

Tahap awal dalam mengidentifikasi kebutuhan yaitu untuk mendapatkan daftar kebutuhan awal yang digunakan dalam aplikasi. Setelah dibangun *prototype* dari daftar kebutuhan, kemudian tahap kedua akan dilakukan evaluasi pengguna untuk mendapatkan perubahan data sesuai dari hasil evaluasi pengguna agar aplikasi benar – benar sesuai dengan pengguna.

#### 4.3.1 Kebutuhan Fungsional

Kebutuhan fungsional dari kode SRS dari *Volunteer Apps* disingkat menjadi VAP. Kebutuhan fungsional pengguna ditunjukkan dengan kode huruf F dan disajikan pada tabel 4.2, tabel 4.3, tabel 4.4, tabel 4.6, dan tabel 4.7. Kebutuhan fungsional memiliki dua tahap, yaitu pada tahap pertama identifikasi kebutuhan awal yang disebut *prototype 1* serta tahap kedua yaitu identifikasi kebutuhan setelah di evaluasi yang disebut *prototype 2*.

##### 4.3.1.1 Pengguna Umum

**Tabel 4.2 Kebutuhan Fungsional Pengguna Umum *Prototype 1***

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	VAP-F-1-01	Login	Sistem dapat mengautentifikasi pengguna untuk masuk



			sebagai pendamping atau mahasiswa difabel.
--	--	--	--

#### 4.3.1.2 Pendamping

**Tabel 4.3 Kebutuhan Fungsional Pendamping *Prototype 1***

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	VAP-F-2-01	Melihat Jadwal Damping	Sistem dapat menampilkan jadwal damping yang sesuai dengan jadwal damping pendamping. Sehingga pendamping mengetahui mahasiswa difabel yang akan mereka dampingi dan mengetahui status jadwal mereka. Warna biru menandakan status jadwal terjadwal, kuning menandakan status jadwal dikonfirmasi, hijau menandakan status jadwal selesai
2	VAP-F-2-02	Menelpon Difabel	Sistem dapat membuka aplikasi telepon beserta no telepon difabel sehingga pendamping dapat melakukan panggilan kepada mahasiswa difabel yang dituju.
3	VAP-F-2-03	Mengirim Pesan Difabel	Sistem dapat membuka aplikasi pesan beserta no telepon difabel sehingga pendamping dapat mengirim pesan kepada mahasiswa difabel yang sudah dijadwalkan.
4	VAP-F-2-04	Melakukan Konfirmasi Kehadiran	Pendamping dapat melakukan konfirmasi kehadiran dengan cara <i>scan qr code</i> yang dimiliki mahasiswa difabel.

5	VAP-F-2-05	Melakukan Filter Jadwal Damping	Pendamping dapat melihat jadwal mereka sesuai dengan tanggal yang ingin mereka lihat.
7	VAP-F-2-06	Logout Pendamping	Sistem dapat mengeluarkan pengguna dari sistem sebagai pendamping.

Setelah evaluasi dilakukan pada *prototype* 1 maka didapatkan hasil yang menyatakan perlu dilakukan penambahan pada daftar kebutuhan fungsional seperti pada tabel 4.4.

**Tabel 4.4 Kebutuhan Fungsional Pendamping *Prototype* 2**

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	VAP-F-2-07	Melihat Profil Difabel	Pendamping dapat melihat detail profil mahasiswa difabel serta detail jadwal yang akan mereka dampingi.

Perubahan pada VAP-F-2-07 dikarenakan 80% pendamping kurang mengetahui informasi data pribadi dari mahasiswa difabel yang akan pendamping dampingi. Data – data pribadi yang akan ditampilkan yaitu nama lengkap, jenis ketunaan, no *handphone*, jurusan, fakultas, angkatan masuk universitas.

#### 4.3.1.3 Mahasiswa Difabel

**Tabel 4.5 Kebutuhan Fungsional Mahasiswa Difabel *Prototype* 1**

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	VAP-F-3-01	Melihat Jadwal Pendampingan	Sistem dapat menampilkan jadwal damping yang sesuai dengan jadwal damping mahasiswa difabel. Sehingga mahasiswa difabel mengetahui siapa pendamping mereka dan mengetahui status jadwal mereka. Warna biru menandakan status jadwal terjadwal, kuning

			menandakan status jadwal dikonfirmasi, hijau menandakan status jadwal selesai.
2	VAP-F-3-02	Menelpon Pendamping	Sistem dapat membuka aplikasi telepon beserta no pendamping sehingga mahasiswa difabel dapat melakukan panggilan kepada pendamping yang di tuju.
3	VAP-F-3-03	Mengirim Pesan Pendamping	Sistem dapat membuka aplikasi pesan beserta no telepon pendamping sehingga mahasiswa difabel dapat mengirimkan pesan kepada pendamping yang dituju.
4	VAP-F-3-04	Menampilkan QR Code	Mahasiswa difabel dapat menampilkan QR Code yang dimiliki mahasiswa difabel di setiap pendamping agar pendamping dapat melakukan konfirmasi kehadiran sesuai dengan jadwal yang sudah ditentukan.
5	VAP-F-3-05	Melakukan Filter Jadwal Pendampingan	Mahasiswa Difabel dapat melihat jadwal mereka sesuai dengan tanggal yang ingin mereka lihat
6	VAP-F-3-06	Memberi <i>Review</i> Pendamping	Mahasiswa Difabel dapat memberi <i>review</i> kepada pendamping yang sudah melakukan konfirmasi kehadiran.
7	VAP-F-3-07	Logout Mahasiswa Difabel	Sistem dapat mengeluarkan pengguna dari sistem sebagai mahasiswa difabel

Setelah evaluasi dilakukan pada *prototype* 1 maka didapatkan hasil yang menyatakan perlu dilakukan penambahan pada daftar kebutuhan fungsional seperti pada tabel 4.6.

**Tabel 4.6 Kebutuhan Fungsional Mahasiswa Difabel *Prototype* 2**

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	VAP-F-3-08	Melihat Profil Pendamping	Mahasiswa difabel dapat melihat detail profil pendamping serta detail jadwal yang sudah terjadwal mendampingi mereka.

Perubahan pada VAP-F-3-08 dikarenakan 80% pendamping kurang mengetahui informasi data pribadi dari mahasiswa difabel yang akan pendamping dampingi. Data – data pribadi yang akan ditampilkan yaitu nama lengkap, no *handphone*, jurusan, fakultas, angkatan masuk universitas.

#### 4.3.2 Kebutuhan Non Fungsional

Selain kebutuhan fungsional, terdapat kebutuhan non fungsional untuk meningkatkan kepuasan pengguna dalam menggunakan aplikasi V-Apps. Kebutuhan non fungsional akan menggunakan kode huruf NF dan disajikan pada tabel 4.7.

**Tabel 4.7 Kebutuhan Non Fungsional *Prototype* 1**

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	VAP-NF-1-01	<i>Usability</i>	Aplikasi harus didesain sedemikian rupa sehingga pengguna merasa mudah dalam menggunakan aplikasi.

#### 4.4 Analisis Data

Analisis data bertujuan untuk mendapatkan struktur data penyimpanan yang akan diimplementasikan pada aplikasi V-Apps, data yang akan digunakan dalam aplikasi V-Apps ini berdasarkan analisis sebagai berikut:

- a. Pengguna umum memiliki beberapa atribut *username*, dan *password*.
- b. Data pendamping yang merupakan representasi data dari pendamping yang memiliki beberapa atribut nama, fakultas, no hp, nim, jurusan, angkatan.

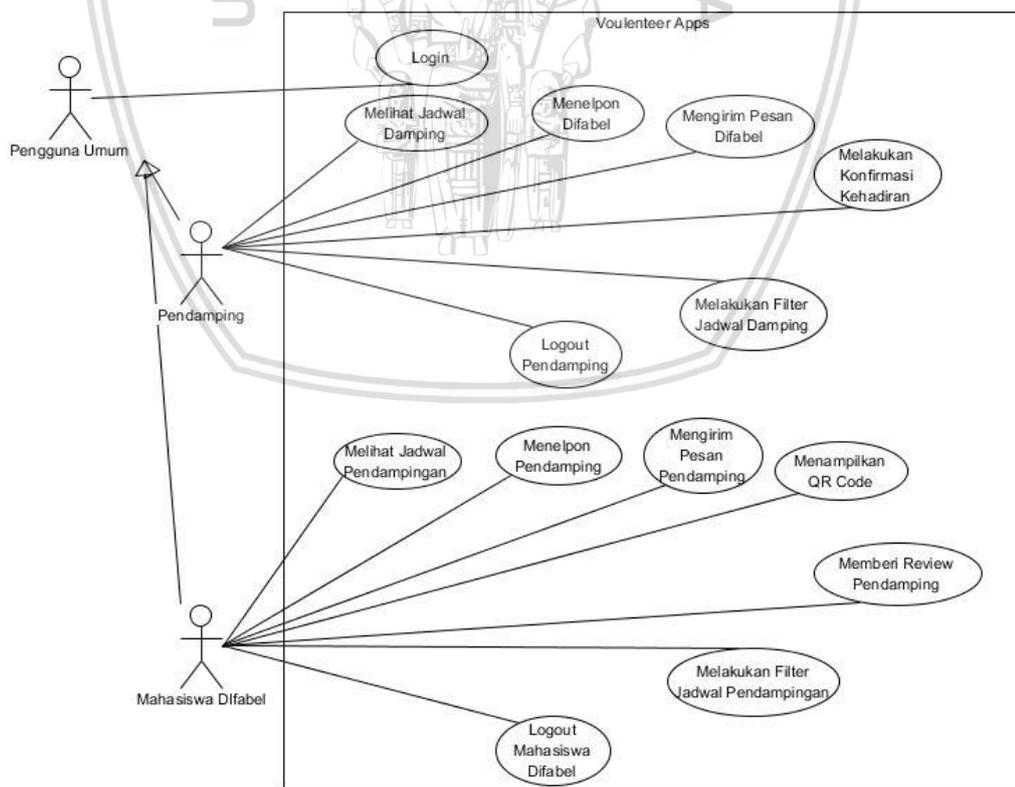
- c. Data Mahasiswa Difabel merupakan representasi data dari mahasiswa difabel yang memiliki beberapa atribut nama, fakultas, no hp, nim, jurusan, angkatan, jenis disabilitas.
- d. *Schedule* merupakan data jadwal pendampingan yang memiliki beberapa attribute hari, jam, kelas, matakuliah, nama pendamping / nama mahasiswa difabel.

#### 4.5 Pemodelan Kebutuhan

Untuk mempermudah dalam memahami kebutuhan sistem, maka dibuat suatu pemodelan kebutuhan. Dalam pemodelan kebutuhan ini terdapat *use case diagram* dan *use case scenario*. *Use case diagram* dibuat agar dapat menjelaskan apa saja yang dapat dilakukan oleh aktor dalam sistem V-Apps. Setelah *use case diagram* dibuat maka selanjutnya akan dibuat *use case scenario* untuk memperjelas rincian tiap *use case diagram* yang sudah dibuat sebelumnya.

##### 4.5.1 Use Case Diagram

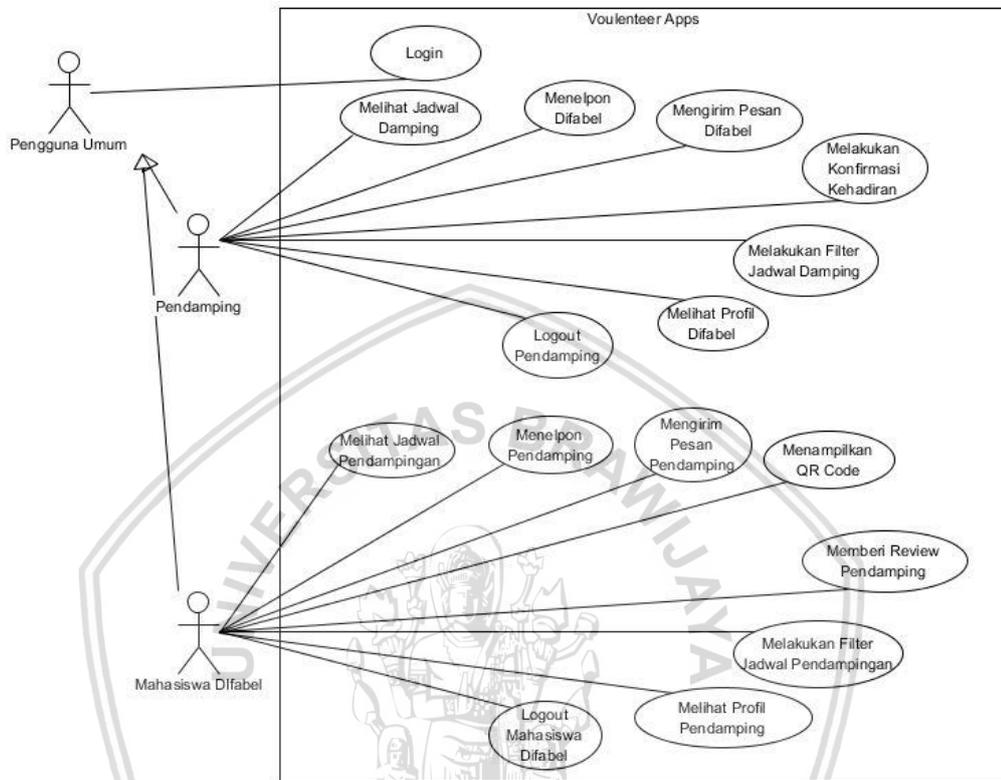
*Use case diagram* merupakan diagram yang akan berfungsi untuk menunjukan hal-hal apa saja yang bisa dilakukan oleh aktor berdasarkan kebutuhan fungsional yang telah dideskripsikan, maka dibuat *use case diagram* pada Gambar 4.1.



Gambar 4.1 Use Case Diagram V-Apps Prototype 1



Use case diagram yang ditampilkan pada Gambar 4.1 mendeskripsikan interaksi aktor dan fungsionalitas yang dapat dilakukan berdasarkan hasil analisis awal. Pada analisis kebutuhan kedua, terdapat kebutuhan tambahan yang dimasukkan kedalam sistem. Kebutuhan yang ditambahkan dapat dilihat pada use case diagram yang ditampilkan pada Gambar 4.2.



Gambar 4.2 Use Case Diagram V-Apps Prototype 2

#### 4.5.2 Use case Scenario

Berikut merupakan Use Case yang menguraikan tentang alur kegiatan yang dilakukan yang dilakukan saat menggunakan sistem dan dapat dikategorikan menurut use case yang sudah dibuat sebelumnya. Use case scenario tahap kedua dapat dilihat ketika pembuatan use case scenario pada tahap pertama telah selesai dibahas.

##### 4.5.2.1 Use Case Scenario Pengguna Umum

###### a. Use Case Scenario : Login (VAP-F-1-01)

Use Case Scenario login Prototype 1 dapat dilihat pada tabel 4.8.

Tabel 4.8 Use Case Scenario Login Prototype 1

Use Case ID	VAP-F-1-01
Nama Use Case	Login

Tujuan	Mengautentifikasi aktor pengguna biasa menjadi pendamping atau mahasiswa difabel
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk masuk ke dalam sistem sebagai pendamping atau mahasiswa difabel.
Aktor	Pengguna Umum
Kondisi Awal	Aktor berada di halaman login
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor memasukkan data <i>username</i> dan <i>password</i> dan menekan tombol login	2. Sistem memasukkan aktor menjadi pendamping atau mahasiswa difabel dan menampilkan tampilan awal yaitu jadwal damping yang dimiliki pendamping atau mahasiswa difabel.
<b>Skenario Alternatif 1 : Jika <i>username</i> dan <i>password</i> tidak terdaftar pada database (Alternatif untuk tahap no. 1)</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
	1. Sistem memberikan notifikasi bahwa <i>username</i> dan <i>password</i> tidak terdaftar.
Kondisi Akhir	Aktor masuk ke dalam sistem menjadi pendamping atau mahasiswa difabel.

#### 4.5.2.2 Use Case Scenario Pendamping

##### a. Use Case Scenario : Melihat Jadwal Damping (VAP-F-2-01)

Use Case Scenario Melihat Jadwal Damping Prototype 1 dapat dilihat pada tabel 4.9.

**Tabel 4.9 Use Case Scenario Melihat Jadwal Damping Prototype 1**

Use Case ID	VAP-F-2-01
Nama Use Case	Melihat Jadwal Damping
Tujuan	Untuk melihat jadwal damping pendamping yang sudah <i>login</i> , sehingga tahu siapa mahasiswa difabel yang akan mereka dampingi dan mengetahui status jadwal mereka.



Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk melihat jadwal damping aktor dan status jadwal aktor.
Aktor	Pendamping
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai pendamping dan berada di menu utama
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor menekan tombol <i>navigation bar</i>	2. Sistem menampilkan menu – menu yang ada di <i>navigation bar</i> .
3. Aktor menekan tombol lihat jadwal	4. Sistem menampilkan jadwal damping dan status jadwal sesuai dengan <i>database</i> aktor
Kondisi akhir	Sistem menampilkan jadwal damping dan status jadwal aktor

**b. Use Case Scenario : Menelpon Difabel (VAP-F-2-02)**

Use Case Scenario Menelpon Difabel Prototype 1 dapat dilihat pada tabel 4.10.

**Tabel 4.10 Use Case Scenario Menelpon Difabel Prototype 1**

Use Case ID	VAP-F-2-02
Nama Use Case	Menelpon Difabel
Tujuan	Untuk mengarahkan aktor ke aplikasi telepon beserta no telepon mahasiswa difabel yang dituju agar aktor dapat melakukan panggilan kepada mahasiswa difabel yang dituju
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk mengarahkan aktor ke aplikasi telepon agar dapat melakukan panggilan kepada mahasiswa difabel yang dituju
Aktor	Pendamping



Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai pendamping dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor menekan tombol telepon pada salah satu mahasiswa difabel atau jadwal	2. Sistem akan mengarahkan aktor ke aplikasi telepon dan menampilkan no telpon mahasiswa difabel yang siap untuk di telepon
Kondisi akhir	Sistem membuka aplikasi telepon dan menampilkan no telpone mahasiswa difabel

**c. Use Case Scenario : Mengirim Pesan Difabel (VAP-F-2-03)**

Use Case Scenario Mengirim Pesan Difabel Prototype 1 dapat dilihat pada tabel 4.11.

**Tabel 4.11 Use Case Scenario Mengirim Pesan Difabel Prototype 1**

Use Case ID	VAP-F-2-03
Nama Use Case	Mengirim Pesan Difabel
Tujuan	Untuk mengarahkan aktor ke aplikasi pesan beserta no telpon mahasiswa difabel yang dituju agar aktor dapat melakukan pengiriman pesan kepada mahasiswa difabel yang dituju
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk mengarahkan aktor ke aplikasi pesan agar dapat melakukan pengiriman pesan <i>text</i> kepada mahasiswa difabel yang dituju
Aktor	Pendamping
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai pendamping dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>



1. Aktor menekan tombol pesan pada salah satu mahasiswa difabel atau jadwal	2. Sistem akan mengarahkan aktor ke aplikasi pesan dan menampilkan no telepon mahasiswa difabel yang dituju
Kondisi akhir	Sistem membuka aplikasi pesan dan menampilkan no telepon mahasiswa difabel yang dituju oleh aktor

**d. Use Case Scenario : Melakukan Konfirmasi Kehadiran (VAP-F-2-04)**

Use Case Scenario Melakukan Konfirmasi Kehadiran *Prototype* 1 dapat dilihat pada tabel 4.12.

**Tabel 4. 12 Use Case Scenario Melakukan Konfirmasi Kehadiran *Prototype* 1**

Use Case ID	VAP-F-2-04
Nama Use Case	Melakukan Konfirmasi Kehadiran
Tujuan	Untuk melakukan konfirmasi kehadiran dengan cara <i>scan QR Code</i> yang dimiliki mahasiswa difabel
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk melakukan konfirmasi kehadiran dengan cara <i>scan QR Code</i> yang dimiliki mahasiswa difabel
Aktor	Pendamping
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai pendamping dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor menekan gambar <i>scan</i> .	2. Sistem akan mengarahkan aktor ke kamera
3. Aktor mengarahkan kamera <i>handphone</i> ke <i>qr code</i> yang dimiliki mahasiswa difabel sesuai dengan yang dijadwalkan sistem	4. Sistem akan merubah status jadwal dari terjadwal yang berwarna biru menjadi dikonfirmasi yang berwarna kuning
Kondisi akhir	Sistem merubah status jadwal damping dari terjadwal yang berwarna biru menjadi dikonfirmasi yang berwarna kuning.



**e. Use Case Scenario : Melakukan Filter Jadwal Damping (VAP-F-2-05)**

Use Case Scenario Melakukan Filter Jadwal Damping Prototype 1 dapat dilihat pada tabel 4.13.

**Tabel 4.13 Use Case Scenario Melakukan Filter Jadwal Damping Prototype 1**

Use Case ID	VAP-F-2-05
Nama Use Case	Melakukan Filter Jadwal Damping
Tujuan	Untuk melihat jadwal sesuai dengan tanggal yang ditentukan oleh aktor
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk melihat jadwal damping dengan cara memfilter sesuai dengan tanggal yang ditentukan oleh aktor
Aktor	Pendamping
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai pendamping dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor menekan gambar tanggal awal	2. Sistem menampilkan tanggal
3. Aktor memilih tanggal yang ingin ditampilkan diawal	4 Sistem menerima inputan aktor
5. Aktor menekan gambar tanggal akhir	6. Sistem menampilkan tanggal
7. Aktor memilih sampai tanggal berapa jadwal ingin ditampilkan	8. Sistem menerima inputan, dan menampilkan jadwal sesuai tanggal yang ditentukan aktor
Kondisi Akhir	Sistem menampilkan jadwal sesuai tanggal yang sudah di tentukan aktor.

**f. Use Case Scenario : Logout Pendamping (VAP-F-2-06)**

Use Case Scenario Logout Pendamping Prototype 1 dapat dilihat pada tabel 4.14.

**Tabel 4.14 Use Case Scenario Logout Pendamping Prototype 1**

Use Case ID	VAP-F-2-06
-------------	------------



Nama Use Case	<i>Logout</i> Pendamping
Tujuan	Untuk keluar dari sistem sebagai pendamping
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk keluar dari sistem sebagai pendamping
Aktor	Pendamping
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai pendamping dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor menekan tombol <i>navigation bar</i>	2. Sistem menampilkan menu – menu yang ada di <i>navigation bar</i> .
3. Aktor menekan tombol keluar	4. Sistem mengeluarkan aktor sebagai pendamping dan menampilkan halaman <i>login</i>
Kondisi akhir	Aktor keluar dari sistem sebagai pendamping dan kembali menjadi aktor pengguna umum dan sistem menampilkan halaman login.

Berdasarkan evaluasi pengguna didapatkan penambahan fitur pada aktor Pendamping. Penambahan ini dilakukan karena adanya permintaan dari pengguna saat evaluasi pengguna. Berikut adalah tambahan fitur dalam *Use case Scenario*.

**g. Use Case Scenario : Melihat Profil Difabel (VAP-F-2-07)**

*Use Case Scenario* Melihat Profil Difabel *Prototype* 1 dapat dilihat pada tabel 4.15.

**Tabel 4.15 Use Case Scenario Melihat Profil Difabel Prototype 2**

<i>Use Case ID</i>	VAP-F-2-07
Nama Use Case	Melihat Profil Difabel
Tujuan	Untuk melihat detail profil mahasiswa difabel dan detail jadwal



Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk melihat detail profil mahasiswa difabel dan detail jadwal
Aktor	Pendamping
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai pendamping dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor menekan nama salah satu mahasiswa difabel	2. Sistem akan menampilkan detail profil serta detail jadwal mahasiswa difabel yang dipilih aktor
Kondisi akhir	Sistem menampilkan detail profil serta detail jadwal mahasiswa difabel yang dipilih aktor.

#### 4.5.2.3 Use Case Scenario Mahasiswa Difabel

##### a. Use Case Scenario : Melihat Jadwal Pendampingan (VAP-F-3-01)

Use Case Scenario Melihat Jadwal Pendampingan *Prototype* 1 dapat dilihat pada tabel 4.16.

**Tabel 4.16 Use Case Scenario Melihat Jadwal *Prototype* 1**

Use Case ID	VAP-F-3-01
Nama Use Case	Melihat Jadwal Pendampingan
Tujuan	Untuk melihat jadwal damping mahasiswa difabel yang <i>login</i> sehingga tahu siapa pendamping yang akan mendampingi aktor dan status dari jadwal mahasiswa difabel tersebut.
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk melihat jadwal damping aktor serta status jadwal aktor
Aktor	Mahasiswa Difabel
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai mahasiswa difabel dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	



Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol <i>navigation bar</i>	2. Sistem menampilkan menu – menu yang ada di <i>navigation bar</i> .
3. Aktor menekan tombol lihat jadwal	4. Sistem menampilkan jadwal damping dan status jadwal sesuai dengan database aktor
Kondisi akhir	Sistem menampilkan jadwal dan status jadwal aktor

**b. Use Case Scenario : Menelpon Pendamping (VAP-F-3-02)**

*Use Case Scenario* Menelpon Pendamping *Prototype* 1 dapat dilihat pada tabel 4.17.

**Tabel 4.17 Use Case Scenario Menelpon Pendamping Prototype 1**

<i>Use Case ID</i>	VAP-F-3-02
Nama Use Case	Menelpon Pendamping
Tujuan	Untuk mengarahkan aktor ke aplikasi telepon beserta no telepon pendamping yang dituju agar aktor dapat melakukan panggilan kepada pendamping yang dituju
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk mengarahkan aktor ke aplikasi telepon agar dapat melakukan panggilan kepada pendamping yang dituju
Aktor	Mahasiswa Difabel
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai pendamping dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
Aksi Aktor	Reaksi Sistem
1. Aktor menekan tombol telepon pada salah satu pendamping	2. Sistem akan mengarahkan aktor ke aplikasi telepon dan menampilkan no telepon pendamping yang dituju aktor
Kondisi akhir	Sistem membuka aplikasi telepon dan menampilkan no telepon pendamping yang dituju oleh aktor



**c. Use Case Scenario : Mengirim Pesan Pendamping (VAP-F-3-03)**

Use Case Scenario Mengirim Pesan Pendamping *Prototype* 1 dapat dilihat pada tabel 4.18.

**Tabel 4.18 Use Case Scenario Mengirim Pesan Pendamping *Prototype* 1**

Use Case ID	VAP-F-3-03
Nama Use Case	Mengirim Pesan Pendamping
Tujuan	Untuk mengarahkan aktor ke aplikasi pesan beserta no telpon pendamping yang dituju agar aktor dapat melakukan pengiriman pesan kepada pendamping yang dituju
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk mengarahkan aktor ke aplikasi pesan agar dapat melakukan pengiriman pesan <i>text</i> kepada pendamping yang dituju
Aktor	Mahasiswa Difabel
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai Mahasiswa Difabel dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor menekan tombol pesan pada salah satu pendamping	2. Sistem akan mengarahkan aktor ke aplikasi pesan dan menampilkan no telepon pendamping yang dituju aktor
Kondisi akhir	Sistem membuka aplikasi pesan dan menampilkan no telepon pendamping yang dituju oleh aktor

**d. Use Case Scenario : Menampilkan QR Code (VAP-F-3-04)**

Use Case Scenario Menampilkan QR Code *Prototype* 1 dapat dilihat pada tabel 4.19.

**Tabel 4. 19 Use Case Scenario Menampilkan QR Code *Prototype* 1**

Use Case ID	VAP-F-3-04
Nama Use Case	Menampilkan QR Code



Tujuan	Untuk menampilkan <i>QR Code</i> mahasiswa difabel untuk konfirmasi kehadiran pendamping
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk menampilkan <i>QR Code</i>
Aktor	Mahasiswa difabel
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai mahasiswa difabel dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor menekan gambar <i>scan</i> pada salah satu jadwal	2. Sistem menampilkan <i>QR Code</i>
Kondisi akhir	Sistem menampilkan <i>QR Code</i>

**e. Use Case Scenario : Melakukan Filter Jadwal Pendampingan (VAP-F-3-04)**

*Use Case Scenario* Melakukan Filter Jadwal Pendampingan *Prototype 1* dapat dilihat pada tabel 4.20.

**Tabel 4.20 Use Case Scenario Melakukan Filter Jadwal Pendampingan Prototype 1**

<i>Use Case ID</i>	VAP-F-3-05
Nama Use Case	Melakukan Filter Jadwal Pendampingan
Tujuan	Untuk melihat jadwal sesuai dengan tanggal yang ditentukan oleh aktor
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk melihat jadwal pendampingan dengan cara memfilter sesuai dengan tanggal yang ditentukan oleh aktor
Aktor	Mahasiswa difabel
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai mahasiswa difabel dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>



1. Aktor menekan gambar tanggal awal	2. Sistem menampilkan tanggal
3. Aktor memilih tanggal awal	4 Sistem menerima inputan aktor
5. Aktor menekan gambar tanggal akhir	6. Sistem menampilkan tanggal
7. Aktor memilih sampai tanggal berapa jadwal ingin ditampilkan	8. Sistem menerima inputan, dan menampilkan jadwal sesuai tanggal yang ditentukan aktor
Kondisi Akhir	Sistem menampilkan jadwal sesuai tanggal yang sudah di tentukan oleh aktor.

**f. Use Case Scenario : Memberi Review Pendamping (VAP-F-3-06)**

Use Case Scenario Memberi Review Pendamping Prototype 1 dapat dilihat pada tabel 4.21.

**Tabel 4.21 Use Case Scenario Memberi Review Pendamping Prototype 1**

Use Case ID	VAP-F-3-06
Nama Use Case	Memberi Review Pendamping
Tujuan	Untuk memberi penilaian kepada pendamping yang sudah melakukan konfirmasi kehadiran
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan aktor untuk memberikan review kepada pendamping yang telah melakukan konfirmasi kehadiran untuk merubah status menjadi warna hijau yaitu selesai
Aktor	Mahasiswa difabel
Kondisi Awal	Aktor menampilkan qr code
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
	1. Sistem memberikan halaman review kepada aktor
2. Aktor memberikan review kepada pendamping.	3. Sistem menampilkan hasil review sebelum di submit



4. Aktor menekan tombol <i>submit</i>	5. Sistem menampilkan notifikasi “ <i>review</i> sudah diterima” dan merubah status jadwal menjadi selesai
Kondisi Akhir	Sistem menampilkan notifikasi dan merubah status jadwal menjadi berwarna hijau selesai

**g. Use Case Scenario : Logout Mahasiswa Difabel (VAP-F-3-07)**

Use Case Scenario Logout Mahasiswa Difabel Prototype 1 dapat dilihat pada tabel 4.22.

**Tabel 4.22 Use Case Scenario Logout Mahasiswa Difabel Prototype 1**

Use Case ID	VAP-F-3-07
Nama Use Case	Logout Mahasiswa Difabel
Tujuan	Untuk keluar dari sistem sebagai Mahasiswa difabel
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk keluar dari sistem sebagai Mahasiswa difabel
Aktor	Mahasiswa difabel
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai mahasiswa difabel dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor menekan tombol <i>navigation bar</i>	2. Sistem menampilkan menu – menu yang ada di <i>navigation bar</i> .
3. Aktor menekan tombol keluar	4. Sistem mengeluarkan aktor sebagai Mahasiswa difabel dan menampilkan halaman <i>login</i>
Kondisi akhir	Aktor keluar dari sistem sebagai Mahasiswa difabel dan kembali menjadi aktor pengguna umum.

Berdasarkan evaluasi pengguna didapatkan penambahan fitur pada aktor Mahasiswa Difabel. Penambahan ini dilakukan karena adanya permintaan dari pengguna saat evaluasi pengguna. Berikut adalah tambahan fitur dalam Use case Scenario.



#### h. Use Case Scenario : Melihat Profil Pendamping (VAP-F-3-08)

Use Case Scenario Melihat Profil Pendamping *Prototype* 1 dapat dilihat pada tabel 4.23.

**Tabel 4.23 Use Case Scenario Melihat Profil Pendamping *Prototype* 2**

Use Case ID	VAP-F-3-08
Nama Use Case	Melihat Profil Pendamping
Tujuan	Untuk melihat detail profil pendamping dan detail jadwal
Deskripsi	Use case ini adalah fungsi yang dapat dilakukan oleh aktor untuk melihat detail profil pendamping dan detail jadwal
Aktor	Mahasiswa Difabel
Kondisi Awal	Aktor sudah masuk ke dalam sistem sebagai Mahasiswa difabel dan berada di halaman utama menampilkan jadwal
<b>Skenario Utama</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor menekan salah satu jadwal	2. Sistem akan menampilkan detail profil pendamping yang dipilih aktor serta detail jadwal
Kondisi akhir	Sistem menampilkan detail profil pendamping yang dipilih aktor serta detail jadwal

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

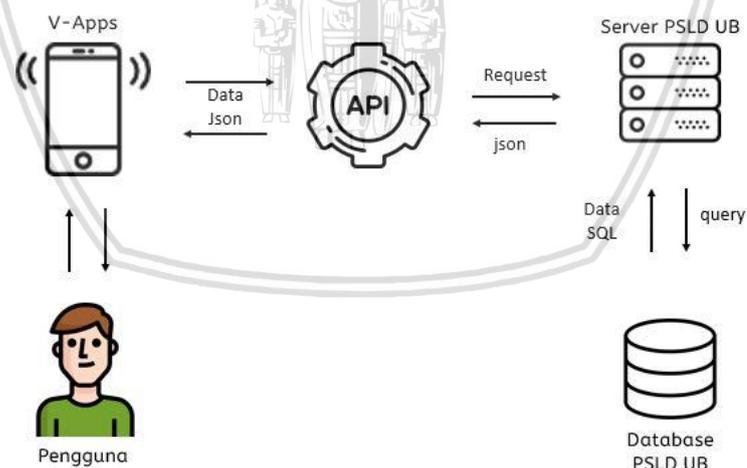
Sebelum mengimplementasikan sistem, dilakukan perancangan sistem yang akan digunakan sebagai dasar untuk mengimplementasikan sistem, dalam tahap ini perancangan yang akan dilakukan adalah perancangan arsitektur sistem, perancangan komponen, perancangan data dan perancangan antarmuka. Setelah dilakukan perancangan sistem kemudian akan dilakukan implementasi sesuai dengan hasil rancangan yang dibuat.

### 5.1 Perancangan

Setelah dilakukan proses analisis kebutuhan, maka tahap berikutnya adalah perancangan. Perancangan dilakukan berdasarkan hasil dari analisis kebutuhan. Proses perancangan V-Apps dibagi menjadi beberapa tahap yaitu perancangan arsitektur, perancangan *class diagram*, *sequence diagram*, perancangan komponen, perancangan data, dan perancangan antarmuka. Dalam perancangan ini dibagi menjadi dua tahap, tahap pertama merupakan perancangan berdasarkan daftar kebutuhan pada *prototype 1*, sedangkan perancangan tahap kedua merupakan pengembangan perancangan dengan menggunakan data identifikasi kebutuhan *prototype 2*.

#### 5.1.1 Perancangan Arsitektur Sistem

Perancangan arsitektur untuk aplikasi V-Apps dapat dilihat pada Gambar 5.1.



**Gambar 5.1 Perancangan Arsitektur Sistem *Prototype 1***

Berikut penjelasan mengenai masing-masing entitas pada Gambar 5.1.

#### a. Pengguna

Pengguna merupakan entitas yang berinteraksi dengan aplikasi secara langsung. Pengguna dapat memanfaatkan semua fitur yang ada pada aplikasi. Pengguna aplikasi yaitu pendamping dan mahasiswa difabel.

b. V-Apps

V-Apps adalah aplikasi yang merupakan entitas yang digunakan untuk menampilkan *user interface* secara langsung pada pengguna. Aplikasi juga memiliki fungsi perantara antara data dan pengguna yang ditampilkan melalui antarmuka aplikasi.

c. API

API membantu komunikasi antara aplikasi V-Apps dengan web sistem penjadwalan PSLD UB.

d. Server PSLD UB

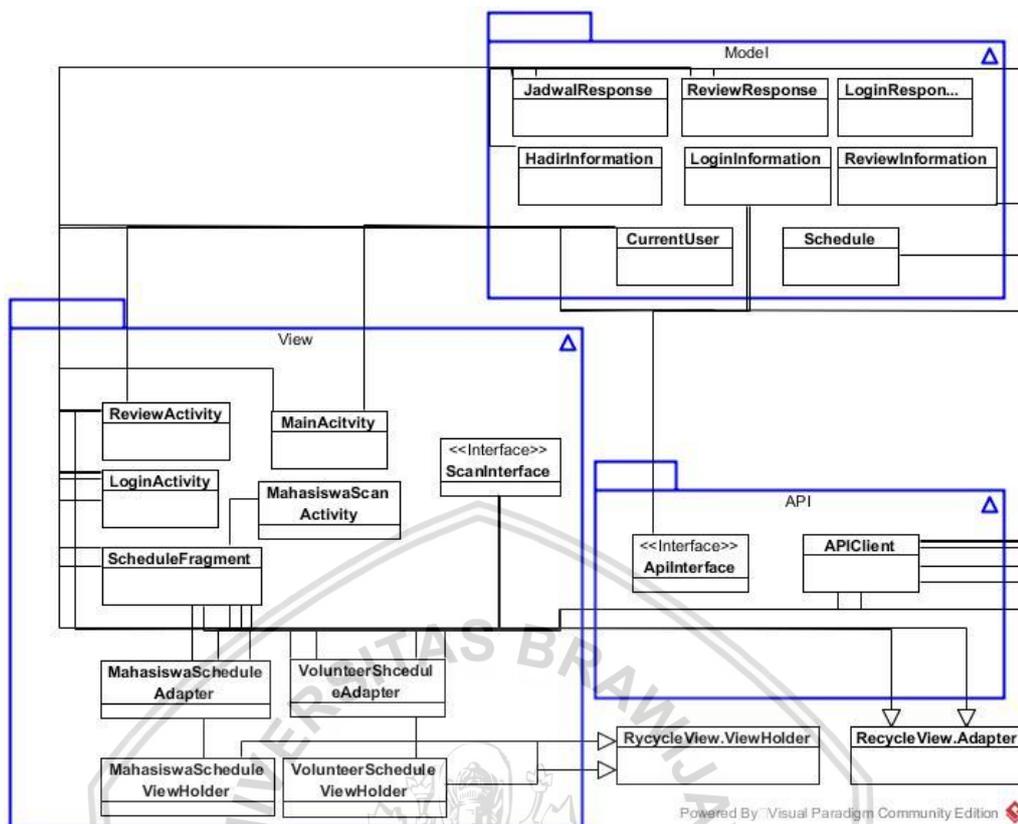
Server bertindak sebagai pemroses data yang menghubungkan *database* dan aplikasi, server dapat digunakan untuk melakukan *read*, *write*, *update*, dan *delete*.

e. *Database* PSLD UB

*Database* dibuat untuk menyimpan data yang akan ditampilkan maupun digunakan dalam aplikasi

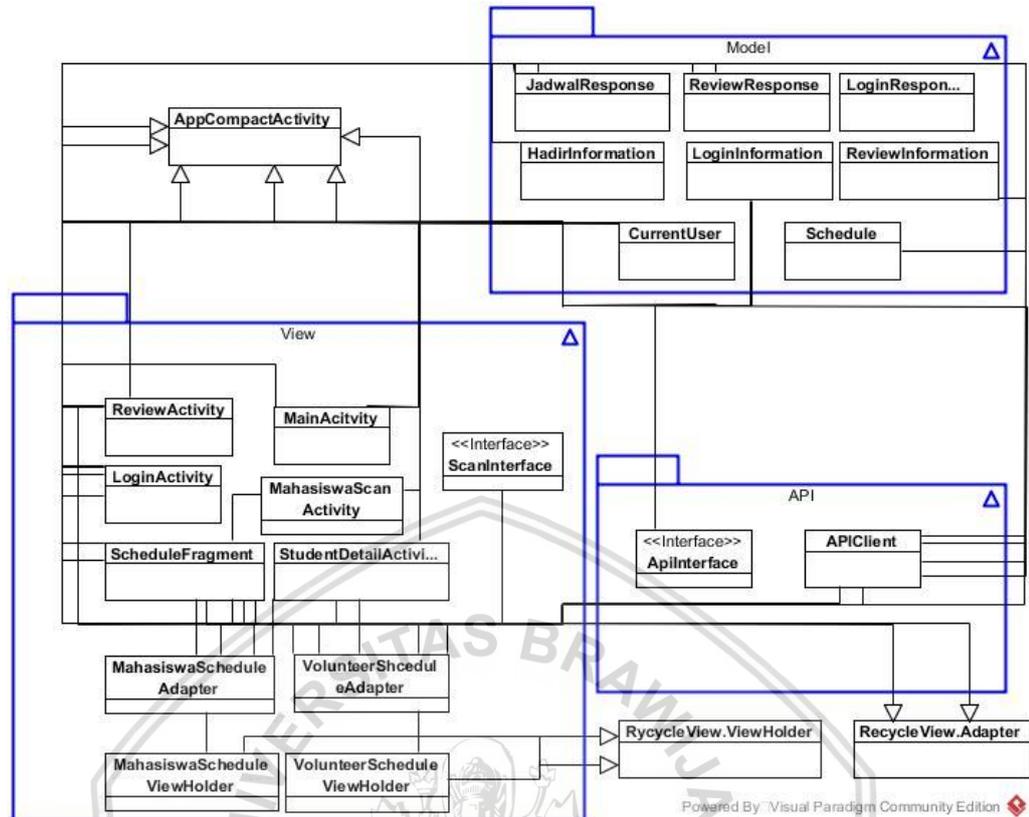
### 5.1.2 Perancangan *Class Diagram*

Perancangan *Class Diagram* dapat dilihat pada Gambar 5.2. Perancangan *class diagram* ini hanya berupa nama *class*, yang dibagi menjadi beberapa *package* sesuai dengan fungsi masing – masing *package*. *Package* yang terdapat dalam perancangan *class diagram* ini yaitu *package model*, *package view*, dan *package API*. Untuk detail *atribute* serta *operation* yang digunakan akan dijelaskan dalam sub bab setelahnya. Dalam penjelasan ini tidak semua *class* dijelaskan, hanya *class* yang dianggap penting serta dapat menjalankan sistem.



Gambar 5. 2 Perancangan Class Diagram Prototype 1

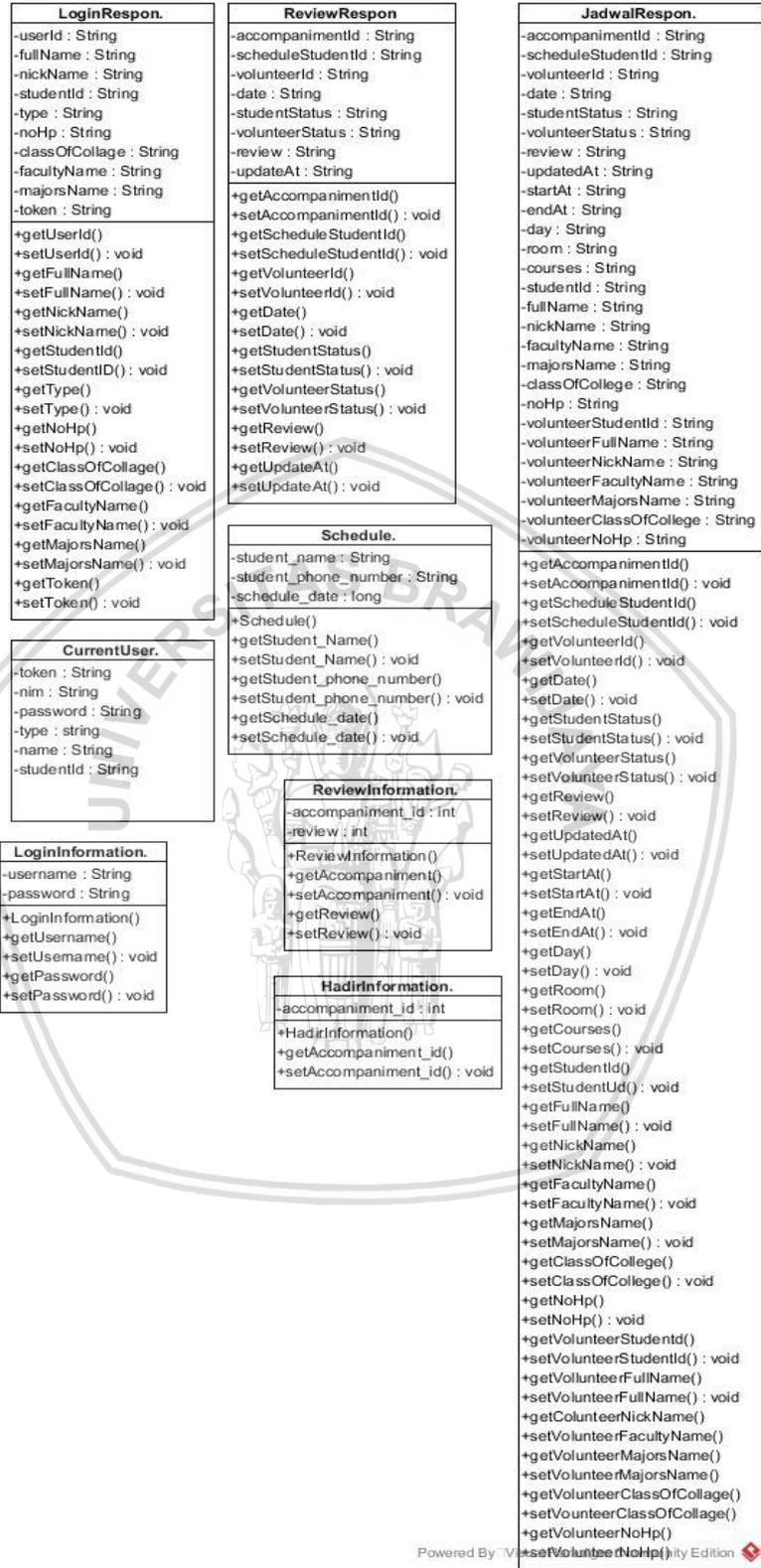
Berikut merupakan penjelasan masing – masing *package*. Penjelasan akan dibagi menjadi tiga *package* yang terdiri dari perancangan *package model* yang berguna untuk mempresentasikan data dari API, perancangan *package view* yang berguna untuk mengatur proses penampilan data kepada pengguna, dan perancangan *package API* yang berguna dalam menjalankan API di aplikasi. Setelah dilakukan evaluasi pengguna terdapat penampahan *class* yang ditunjukkan pada Gambar 5.3.



**Gambar 5.3 Perancangan Class Diagram Prototype 2**

Pada Gambar 5.3 terdapat penambahan *class* karena ada penambahan fitur melihat profil untuk pendamping dan untuk mahasiswa difabel. Fitur tersebut didapatkan berdasarkan hasil evaluasi pengguna. *Class* yang ditambahkan adalah *class StudentDetailActivity* yang terdapat pada *package view*. Penjelasan *class* pada *package* akan dijelaskan lebih detail pada penjelasan *package*.

### 5.1.2.1 Perancangan Package Model

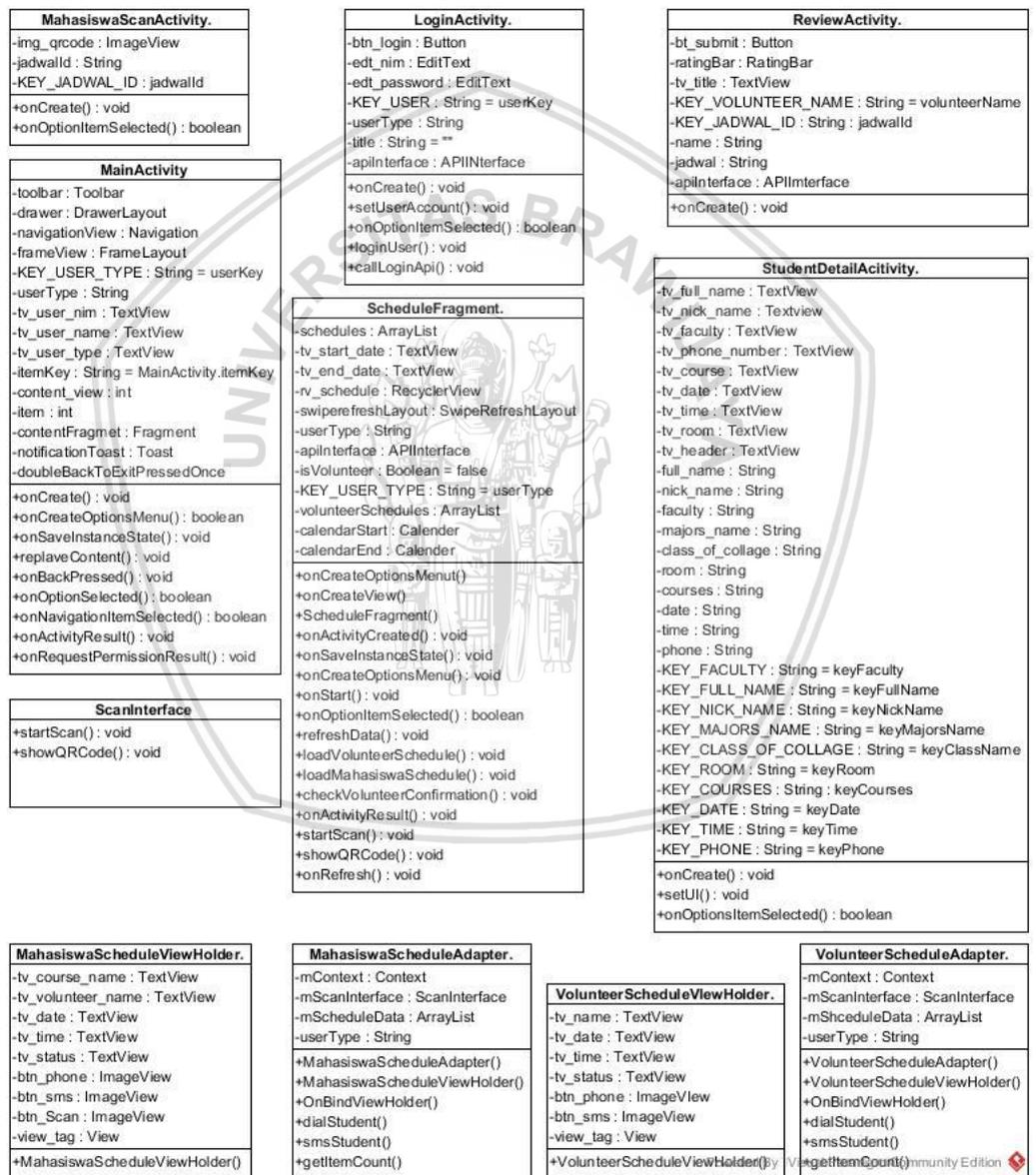


Gambar 5. 4 Perancangan Class Diagram Package Model



Terdapat enam model yakni *loginRespon* digunakan untuk mempresentasikan data *login* dari API, *ReviewRespon* digunakan untuk mempresentasikan data *review* dari API, *HadirInformation* digunakan untuk mempresentasikan data kehadiran dari API, *LoginInformation* untuk mempresentasikan data *login* dari pengguna umum, *ReviewInformation* untuk mempresentasikan data *review* dari API untu mahasiswa difabel, *Schedule* untuk mempresentasikan data jadwal dari API dari setiap pengguna, dan *jadwalRespon* untuk mempresentasikan data setiap jadwal pendamping / mahasiswa difabel.

### 5.1.2.2 Perancangan Package View

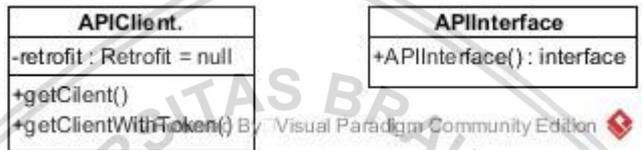


Gambar 5. 5 Perancangan Class Diagram Package View

Tedapat enam *class* yang mengatur tampilnya *view* pada aplikasi V-Apps yaitu *ScheduleFragment* yang berguna untuk tampilan halaman utama yang

berguna untuk menampilkan jadwal, *LoginActivity* untuk berguna untuk tampilan pada proses *login*, *StudentDetailActivity* berguna untuk menampilkan detail profil baik itu pendamping maupun difabel, *MainActivity* untuk membantu dalam proses penampilan, *ReviewActivity* berguna dalam membantu proses pemberian *review* pendamping, *MahasiswaScanActivity* dan *scanInterface* berguna untuk melakukan proses *scan* atau konfirmasi kehadiran pendamping. Berikutnya adalah *VolunteerSchedulAdapter* dan *VolunteerScheduleViewHolder* yang berguna untuk memproses data jadwal agar dapat ditampilkan kepada pengguna pendamping. Dan *MahasiswaScheduleAdapter* dan *MahasiswaScheduleViewHolder* berguna untuk memproses data jadal agar dapat ditampilkan kepada pengguna mahasiswa difabel.

**5.1.2.3 Perancangan Package API**



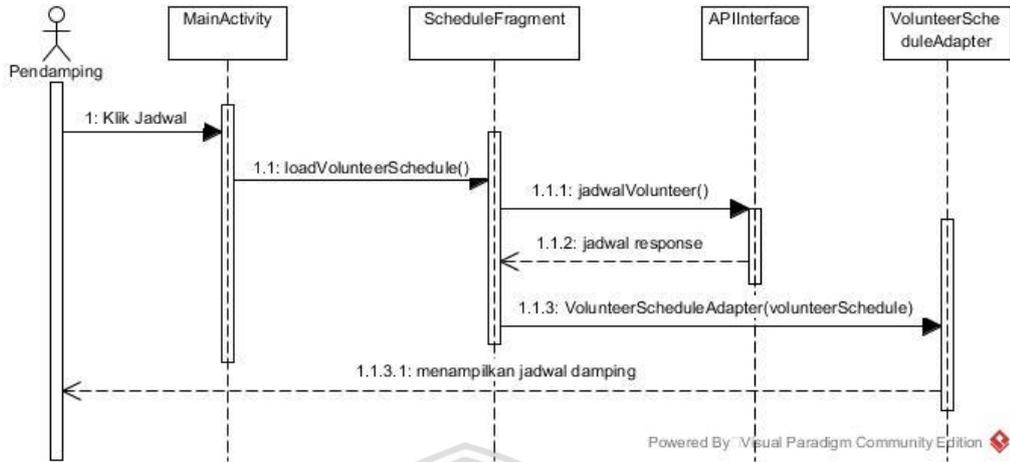
**Gambar 5. 6 Perancangan Class Diagram Package API**

Terdapat dua *class* yang membantu dalam proses penambilan data pada API. *Class* tersebut adalah *APIClient* yang memiliki *method* *getClient* serta *getClientWithToken* dan *class* *APIInterface* yang membantu proses jalannya API.

**5.1.3 Sequence Diagram**

Pada *squnce diagram* merupakan diagram yang menggambarkan alur atau jalannya proses dalam sistem. Dalam sub bab ini akan ditampilkan *squnce diagram* dari aplikasi V-Apps. *Sequence diagram* juga terbagi dalam dua tahap, tahap pertama adalah *squnce diagram* untuk *prototype 1* serta tahap kedua terdapat perubahan pada *squnce diagram* tahap pertama yang merupakan *squnce diagram* dari *prototype 2*.

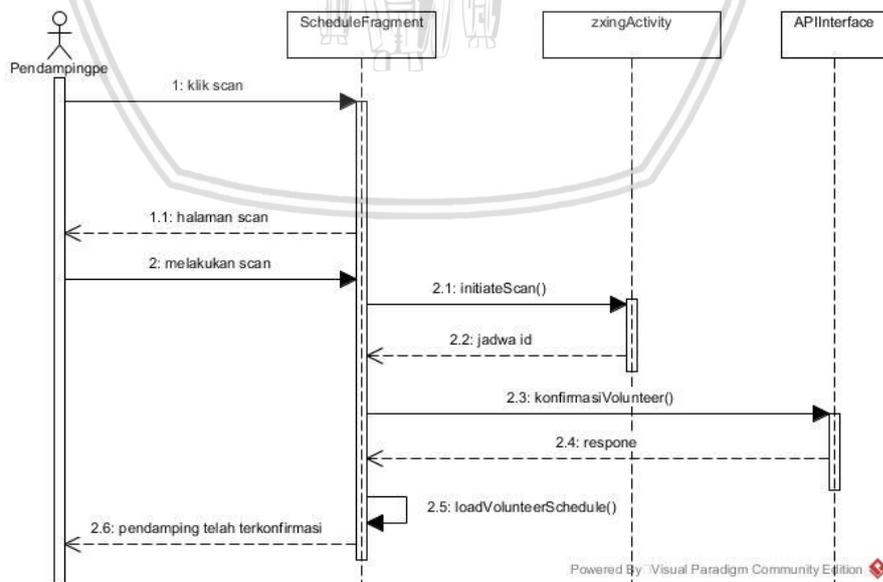
### 5.1.3.1 Sequence Diagram Melihat Jadwal Damping



Gambar 5.7 Sequence Diagram Melihat Jadwal Damping Prototype 1

Pada Gambar 5.6, terdapat interaksi yang menggambarkan skenario yang terjadi dalam menjalankan fungsi *use case* melihat jadwal damping. Pada *sequence diagram* melihat jadwal damping, aktor menekan tombol jadwal untuk melihat jadwal damping yang dimiliki aktor pendamping. Kemudian sistem akan memanggil method *loadVolunteerSchedule()* yang ada di kelas *ScheduleFragment* kemudian memanggil method *jadwalVolunteer()* yang berfungsi untuk mengambil jadwal yang dimiliki pendamping di kelas *APIInterface*. Setelah jadwal diambil kemudian akan di proses di kelas *VolunteerScheduleAdapter* kemudian sistem akan menampilkan jadwal damping yang dimiliki pendamping.

### 5.1.3.2 Sequence Diagram Melakukan Konfirmasi Kehadiran

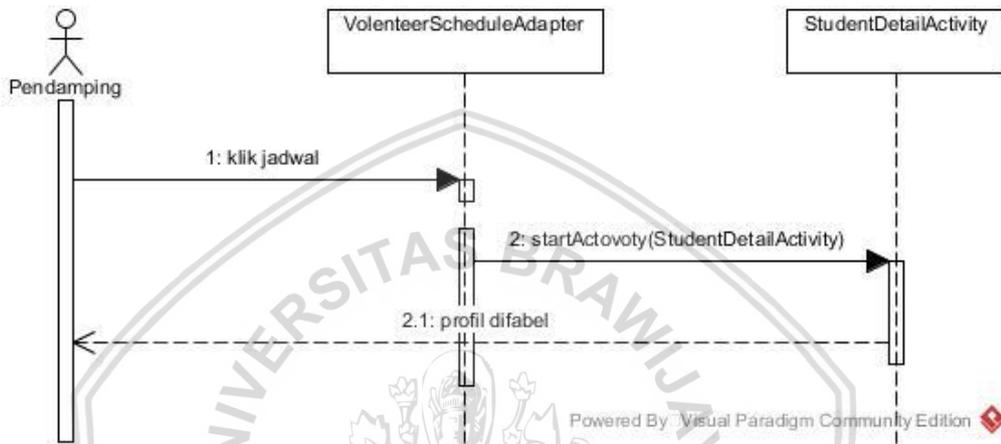


Gambar 5.8 Sequence Diagram Melakukan Konfirmasi Kehadiran Prototype 1



Pada Gambar 5.7, dijelaskan bagaimana menjalankan fungsi *use case* melakukan konfirmasi kehadiran. Aktor menekan *scan*, sistem akan membuka kamera dan aktor mengarahkan kamera ke *qr code* yang dimiliki pendamping. Setelah itu akan memanggil *method initiatescan()* yang berada di kelas *zxingActivity* dan merespon jadwal id. Kemudian memanggil *method konfirmasiVolunteer()* dan sistem akan meresponnya. Kelas *ScheduleFragment* akan memproses konfirmasi tersebut untuk kemudian ditampilkan kepada aktor bahwa jadwal telah di konfirmasi.

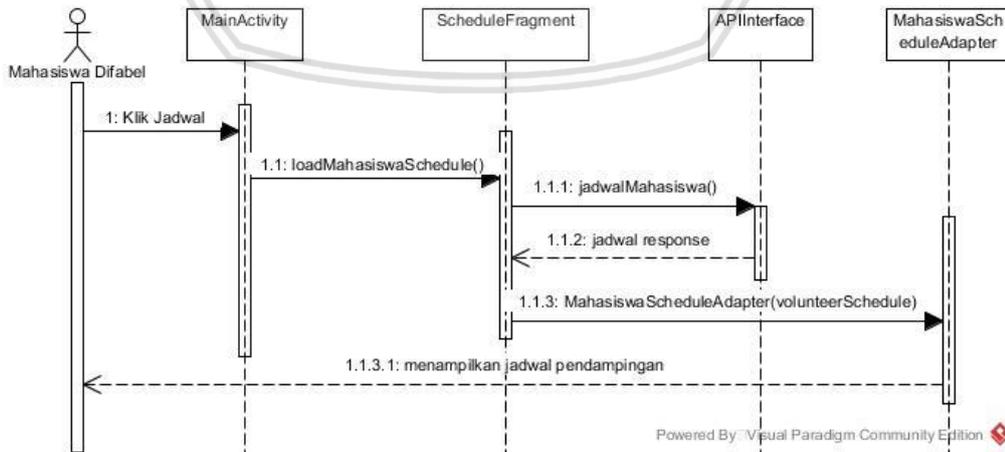
**5.1.3.3 Sequence Diagram Melihat Profil Difabel**



**Gambar 5.9 Sequence Diagram Melihat Profil Difabel Prototype 2**

Pada Gambar 5.8, dijelaskan bagaimana menjalankan fungsi *use case* melihat profil difabel. Aktor menkan salah satu jadwal kemudia di metho *VolunteerScheduleAdapter* memanggil *method startActivity* yang akan memanggil kelas *StudentDetailActivity* kemudian kelas tersebut akan menampilkan data profil yang dimiliki mahasiswa difabel.

**5.1.3.4 Sequence Diagram Melihat Jadwal Pendampingan**

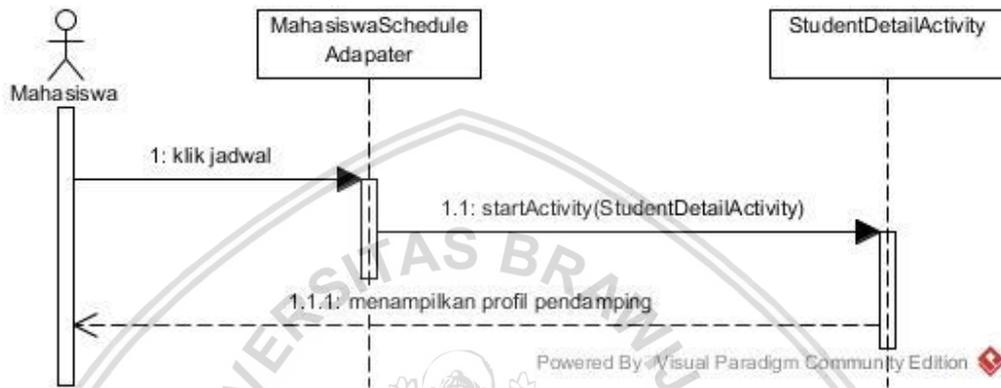


**Gambar 5.10 Sequence Diagram Melihat Jadwal Pendampingan Prototype 1**

Pada Gambar 5.9, terdapat interaksi yang menggambarkan skenario yang terjadi dalam menjalankan fungsi *use case* melihat jadwal pendampingan.

Pada *sequence diagram* melihat jadwal damping, aktor menekan tombol jadwal untuk melihat jadwal damping yang dimiliki aktor pendamping. Kemudian sistem akan memanggil method *loadMahasiswaSchedule()* yang ada di kelas *ScheduleFragment* kemudian memanggil method *jadwalMahasiswa()* yang berfungsi untuk mengambil jadwal yang dimiliki pendamping di kelas *APIInterface*. Setelah jadwal diambil kemudian akan di proses di kelas *MahasiswaScheduleAdapter* kemudian sistem akan menampilkan jadwal damping yang dimiliki mahasiswa difabel.

**5.1.3.5 Sequence Diagram Melihat Profil Pendamping**



**Gambar 5.11 Sequence Diagram Melihat Profil Pendamping Prototype 2**

Pada Gambar 5.10, dijelaskan bagaimana menjalankan fungsi *use case* melihat profil pendamping. Aktor menekan salah satu jadwal kemudian di metho *MahasiswaScheduleAdapter* memanggil method *startActivity* yang akan memanggil kelas *StudentDetailActivity* kemudian kelas tersebut akan menampilkan data profil yang dimiliki oleh pendamping.

**5.1.4 Perancangan Komponen**

Perancangan komponen menggambarkan rincian sub-sistem dari setiap komponen perangkat lunak. Untuk mencapai hal ini, perancangan komponen harus mendefinisikan struktur data untuk semua objek data lokal dan rincian algoritma untuk proses yang terjadi di dalam komponen. Dalam perancangan komponen ini dipilih tiga algoritma *method* utama yang berguna dalam menampilkan jadwal damping pendamping dan jadwal pendampingan mahasiswa difabel. *Method* yang digunakan adalah *loadVolunteerSchedule()*, *loadMahasiswaSchedule()*, dan *onActivityResult()*.

**5.1.4.1 Perancangan Komponen Melihat Jadwal Damping**

Nama Kelas : *ScheduleFragment*

Nama *Method*: *loadVolunteerSchedule*

Deskripsi : Metode ini merupakan method yang dijalankan ketika pengguna umum *login* sebagai pendamping dan menampilkan jadwal damping pendamping yang *login*.



**Algoritme :**

Pengguna melakukan *input* tanggal awal dan tanggal akhir yang ingin ditampilkan

Sistem mengambil data jadwal sesuai dengan data pengguna

Jika data jadwal tidak null

Maka sistem akan menampilkan semua data jadwal damping sesuai dengan tanggal yang sudah ditentukan

Jika tidak ada jadwal

Data tidak dapat ditemukan atau tidak ada yang Ditampilkan

#### 5.1.4.2 Perancangan Komponen Melihat Jadwal Pendampingan

Nama Kelas : *ScheduleFragment*

Nama *Method*: *loadMahasiswaSchedule*

Deskripsi : Metode ini merupakan method yang dijalankan ketika pengguna umum *login* sebagai pendamping dan menampilkan jadwal damping pendamping yang *login*.

**Algoritme :**

Pengguna melakukan *input* tanggal awal dan tanggal akhir yang ingin ditampilkan

Sistem mengambil data jadwal sesuai dengan data pengguna

Jika data jadwal tidak null

Maka sistem akan menampilkan semua data jadwal damping sesuai dengan tanggal yang sudah ditentukan

Jika tidak ada jadwal

Data tidak dapat ditemukan atau tidak ada yang Ditampilkan

#### 5.1.4.3 Perancangan Komponen Melakukan Konfirmasi Kehadiran

Nama Kelas : *ScheduleFragment*

Nama *Method*: *onAcitivityResult*

Deskripsi : Metode ini merupakan method yang dijalankan ketika pengguna pendamping melakukan konfirmasi kehadiran.

**Algoritme :**

Mengambil data *scan* dari *qr* code mahasiswa

Jika data tidak ada

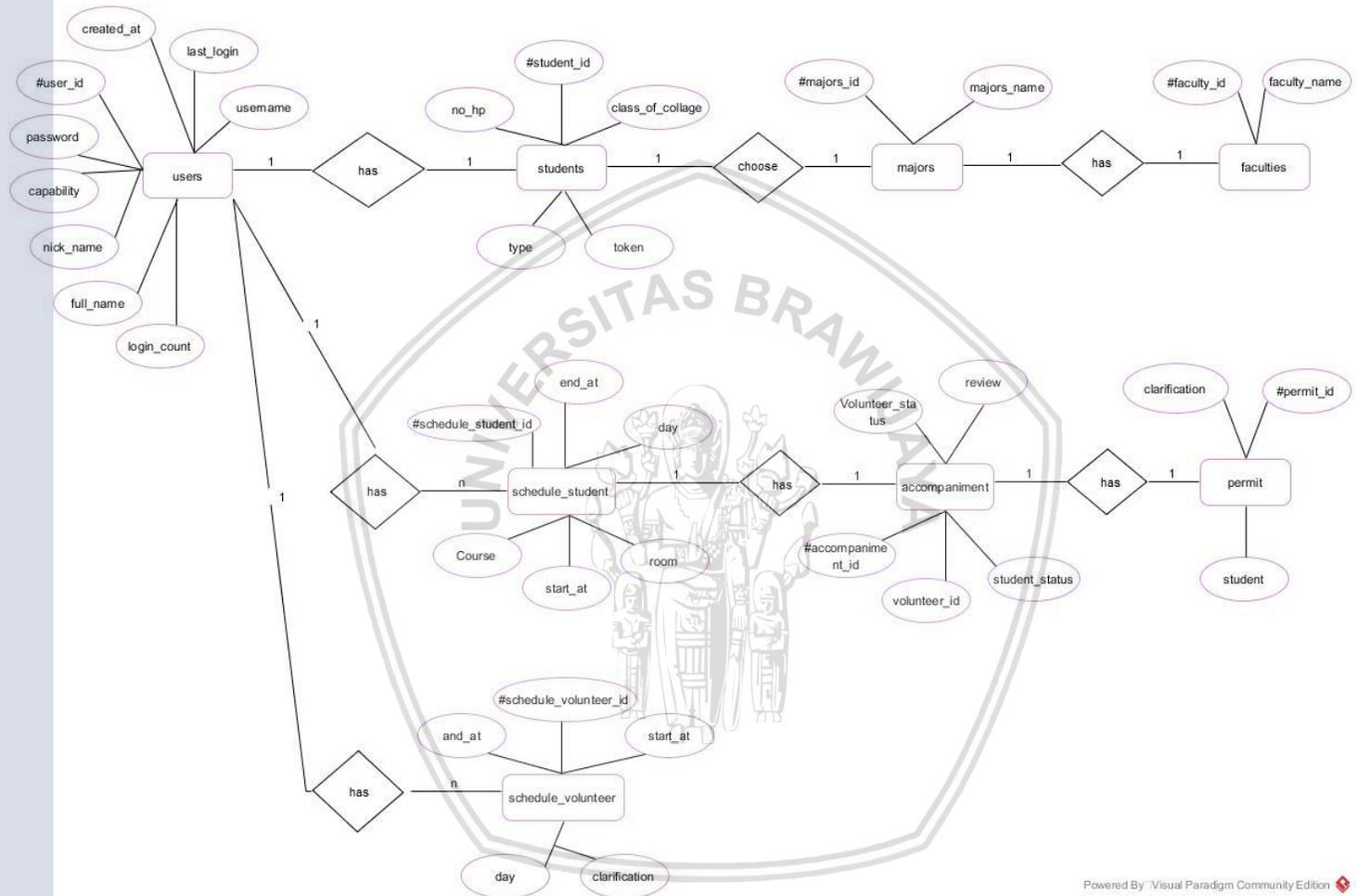
```
Maka sistem akan menampilkan pesan eror bahwa data tidak
dapat di proses
Jika data ada
Maka akan diambil id yang dimiliki mahasiswa difabel
kemudian sistem akan memanggil api untuk melakukan
konfirmasi sesuai id Jadwal yang di konfirmasi.
    Jika konfirmasi berhasil
        Maka status jadwal berubah dikonfirmasi, warna
        status jadwal berubah menjadi kuning dan
        menampilkan notif konfirmasi berhasil.
    Jika jadwal tidak berhasil / null
        Maka jadwal tidak dapat diproses
```

### 5.1.5 Perancangan Data

Aplikasi V-App menggunakan API yang berada disistem penjadwalan jadwal damping PSLD UB. Pada perancangan data kali ini akan ditulis database web sistem jadwal damping dan API yang digunakan untuk keperluan aplikasi V-Apps. API yang dapat dipakai dijelaskan pada tabel 5.9.

#### 5.1.5.1 Database Sistem PSLD UB

Database ini didapat dari sistem penjadwalan jadwal damping yang dibuat serta berada di kantor PSLD UB. Perancangan basis data pada sistem ini menggunakan *Entity Relationship Diagram* (ERD). ERD menggambarkan hubungan antara setiap *entity* atau entitas yang ada dalam sistem. Masing - masing entitas memiliki atribut yang didalamnya terdapat *primary key* (PK). Gambar 5.8 Menunjukkan perancangan ERD dalam sistem penjadwalan jadwal damping.



Powered By Visual Paradigm Community Edition

Gambar 5.12 Perancangan Entity Relationship Diagram

Terdapat 9 tabel yang digunakan untuk sistem dan akan dijelaskan rancangan secara detail pada setiap tabelnya. Tabel – tabel tersebut adalah tabel *user* dijelaskan pada tabel 5.1, tabel *student* dijelaskan pada tabel 5.2, tabel *majors* dijelaskan pada tabel 5.3, tabel *faculties* dijelaskan pada tabel 5.4, tabel *schedule\_student* dijelaskan pada tabel 5.5, tabel *schedule\_volunteer* dijelaskan pada tabel 5.6, tabel *accompaniment* dijelaskan pada tabel 5.7, dan tabel *permit* dijelaskan pada tabel 5.8.

a. Tabel *User*

**Tabel 5.1 Tabel *User***

No	Nama Field	Tipe	Panjang Nilai	Keterangan
1	<i>User_id</i>	<i>Int</i>	11	Sebagai <i>primary key</i>
2	<i>Username</i>	<i>Varchar</i>	100	Berisi <i>username user</i>
3	<i>Password</i>	<i>Varchar</i>	50	Berisi <i>password user</i>
4	<i>Full_name</i>	<i>Varchar</i>	50	Berisi nama lengkap <i>user</i>
5	<i>Nick_name</i>	<i>Int</i>	50	Berisi nama panggilan
6	<i>Capability</i>	<i>Int</i>	1	Kapasistas <i>user</i>
7	<i>Last_login</i>	<i>Datetime</i>	-	<i>Login</i> terakhir dari <i>user</i>

Tabel *user* berfungsi untuk menyimpan data pengguna baik itu pendamping maupun mahasiswa difabel. Data ini berguna untuk pengguna *login* di aplikasi V-Apps.

b. Tabel *Students*

**Tabel 5.2 Tabel *Students***

No	Nama Field	Tipe	Panjang Nilai	Keterangan
1	<i>Student_id</i>	<i>Int</i>	11	Sebagai <i>primary key</i>
2	<i>Type</i>	<i>Int</i>	11	Sebagai <i>type</i> dari <i>student</i>
4	<i>Class_of_collage</i>	<i>Year</i>	4	Berisi kelas <i>student</i>
5	<i>No_hp</i>	<i>Int</i>	13	Berisi no hp <i>student</i>
6	<i>Token</i>	<i>Int</i>	50	Berisi <i>token</i> setiap <i>student</i>

Tabel *Students* berguna untuk mendapatkan *type* dan *token* setelah pengguna *login*. Sehingga mengetahui pengguna login sebagai pendamping atau mahasiswa difabel.

## c. Tabel Majors

Tabel 5.3 Tabel *Majors*

No	Nama Field	Tipe	Panjang Nilai	Keterangan
1	<i>Majors_id</i>	<i>Int</i>	11	Sebagai <i>primary key</i>
3	<i>Majors_name</i>	<i>Int</i>	11	Berisi nama jurusan

Tabel yang berisikan nama jurusan pada setiap fakultas yang sudah dibuat dan ditentukan fakultasnya oleh pihak admin PSLD UB.

## d. Tabel Faculties

Tabel 5.4 Tabel *Faculties*

No	Nama Field	Tipe	Panjang Nilai	Keterangan
1	<i>Faculty_id</i>	<i>Int</i>	11	Sebagai <i>primary key</i>
2	<i>Faculty_name</i>	<i>Int</i>	11	Berisi nama fakultas

Tabel yang berisikan data nama – nama fakultas yang sudah dibuat oleh admin PSLD UB.

## e. Tabel Schedule\_Student

Tabel 5.5 Tabel *Schedule\_Student*

No	Nama Field	Tipe	Panjang Nilai	Keterangan
1	<i>Schedule_student_id</i>	<i>Int</i>	11	Sebagai <i>primary key</i>
2	<i>Start_id</i>	<i>Time</i>		Awal melihat jadwal
3	<i>End_at</i>	<i>Time</i>		Akhir melihat jadwal
4	<i>Day</i>	<i>Int</i>	11	Berisi hari pada jadwal
5	<i>Room</i>	<i>Varchar</i>	10	Berisi nama ruangan
6	<i>Courses</i>	<i>Varchar</i>	100	Berisi nama mata kuliah

Tabel yang berisikan data – data jadwal pendampingan apabila pengguna berhasil *login* sebagai mahasiswa difabel.

## f. Tabel Schedule\_Volunteer

Tabel 5.6 Tabel *Schedule\_Volunteer*

No	Nama Field	Tipe	Panjang Nilai	Keterangan
1	<i>Schedule_volunteer_id</i>	<i>Int</i>	11	Sebagai <i>primarry key</i>
2	<i>Start_id</i>	<i>Time</i>	-	Awal melihat jadwal
3	<i>End_at</i>	<i>Time</i>	-	Akhir melihat jadwal
4	<i>Day</i>	<i>Int</i>	11	Berisi hari pada jadwal

Tabel yang berisikan data – data jadwal pendampingan apabila pengguna berhasil *login* sebagai pendamping.

## g. Tabel Accompaniment

Tabel 5.7 Tabel *Accompaniment*

No	Nama Field	Tipe	Panjang Nilai	Keterangan
1	<i>Accompaniment_id</i>	<i>Int</i>	11	Sebagai <i>primarry key</i>
2	<i>Schedule_student_id</i>	<i>Int</i>	11	Sebagai <i>primarry key</i> mahasiswa
3	<i>Volunteer_id</i>	<i>Int</i>	11	Sebagai <i>primarry key</i> volunteer
4	<i>Student_status</i>	<i>Int</i>	1	Berisi status jadwal milik mahasiswa difabel
5	<i>Volunteer_status</i>	<i>Int</i>	1	Berisi status jadwal milik pendamping
6	<i>Review</i>	<i>Int</i>	1	Berisi <i>review</i> pendamping

Tabel yang berguna dalam proses konfirmasi kehadiran pendamping dan pemberian *review* kepada pendamping.

## h. Tabel Permit

Tabel 5.8 Tabel *Permit*

No	Nama Field	Tipe	Panjang Nilai	Keterangan
1	<i>Permit_id</i>	<i>Int</i>	11	Sebagai <i>primary key</i>
3	<i>Student</i>	<i>Int</i>	1	Sebagai pendamping atau mahasiswa difabel
4	<i>Clarification</i>	<i>Text</i>	-	Sebagai klarifikasi jadwal

Tabel yang berfungsi untuk menyimpan data kehadiran dari pendamping. Sehingga admin mengetahui siapa saja pendamping yang tidak hadir.

## 5.1.5.2 API V-Apps

Aplikasi V-Apps memerlukan rancangan API dari sistem penjadwalan jadwal damping yang dibuat oleh pihak PSLD UB. Di sini akan dituliskan *method*, *endpoint*, *body*, dan *user* yang dapat menggunakan API. Pengujian menggunakan aplikasi Postman yang hasilnya ditampilkan pada implementasi database.

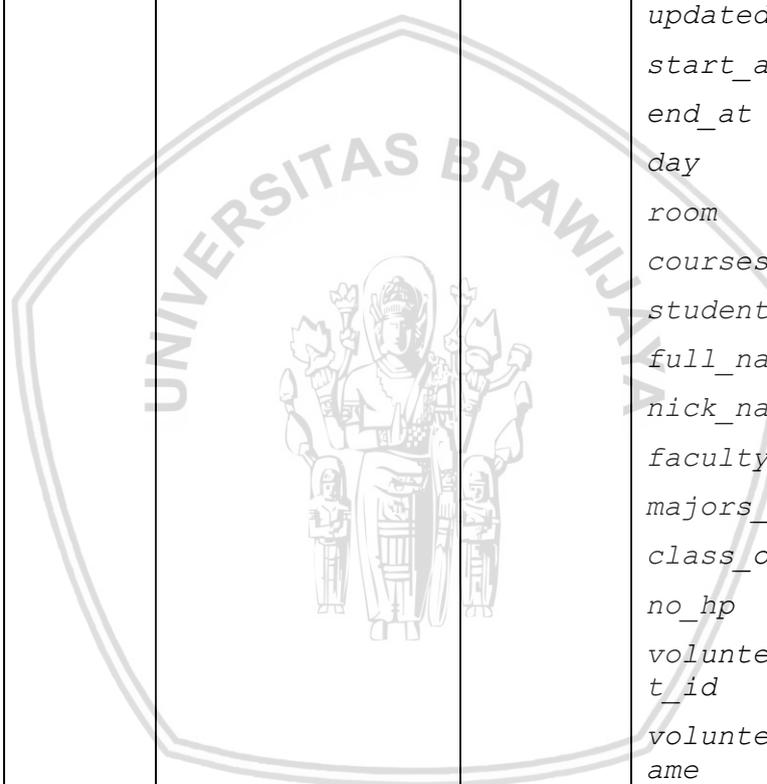
Tabel 5.9 Tabel API V-Apps

No	Method	Endpoint	Body	Result
1	<i>post</i>	<i>http://penjadwalan.orangeslab.com/api/login</i>	<i>Username, password</i>	<i>user_id</i> <i>full_name</i> <i>nick_name</i> <i>student_id</i> <i>type</i> <i>no_hp</i> <i>class_of_college</i> <i>faculty_name</i> <i>majors_name</i> <i>token</i>
2	<i>post</i>	<i>http://penjadwalan.orangeslab.com/api/logout</i>		<i>String</i>
3	<i>Get</i>	<i>http://penjadwalan.orangeslab.com/api/mahasiswa/jadwal</i>	<i>Start, end (date: 2018-04-13)</i>	<i>accompaniment_id</i> <i>schedule_student_id</i> <i>volunteer_id</i> <i>date</i> <i>student_status</i>

				<i>volunteer_status</i> <i>review</i> <i>updated_at</i> <i>start_at</i> <i>end_at</i> <i>day</i> <i>room</i> <i>courses</i> <i>student_id</i> <i>full_name</i> <i>nick_name</i> <i>faculty_name</i> <i>majors_name</i> <i>class_of_college</i> <i>no_hp</i> <i>volunteer_studen</i> <i>t_id</i> <i>volunteer_full_n</i> <i>ame</i> <i>volunteer_nick_n</i> <i>ame</i> <i>volunteer_facult</i> <i>y_name</i> <i>volunteer_majors</i> <i>_name</i> <i>volunteer_class</i> <i>of_college</i> <i>volunteer_no_hp</i>
4	Post	<a href="http://penjadwalan.orangeslab.com/api/mahasiswa/hadir">http://penjadwalan.orangeslab.com/api/mahasiswa/hadir</a>	<i>accompaniment_id(int)</i>	<i>boolean</i>
5	Post	<a href="http://penjadwalan.orangeslab.com/api/mahasiswa/review">http://penjadwalan.orangeslab.com/api/mahasiswa/review</a>	<i>accompaniment_id(int), review(int)</i>	<i>accompaniment_id</i> <i>schedule_student_id</i> <i>volunteer_id</i> <i>date</i> <i>student_status</i> <i>volunteer_status</i>



				<i>review</i> <i>updated_at</i>
6	Get	<a href="http://penjadwalan.orangeslab.com/api/volunteer/jadwal">http://penjadwalan.orangeslab.com/api/volunteer/jadwal</a>	Start, end (date: 2018-04-13)	<i>accompaniment_id</i> <i>schedule_student_id</i> <i>volunteer_id</i> <i>date</i> <i>student_status</i> <i>volunteer_status</i> <i>review</i> <i>updated_at</i> <i>start_at</i> <i>end_at</i> <i>day</i> <i>room</i> <i>courses</i> <i>student_id</i> <i>full_name</i> <i>nick_name</i> <i>faculty_name</i> <i>majors_name</i> <i>class_of_college</i> <i>no_hp</i> <i>volunteer_student_id</i> <i>volunteer_full_name</i> <i>volunteer_nick_name</i> <i>volunteer_faculty_name</i> <i>volunteer_majors_name</i> <i>volunteer_class_of_college</i> <i>volunteer_no_hp</i>



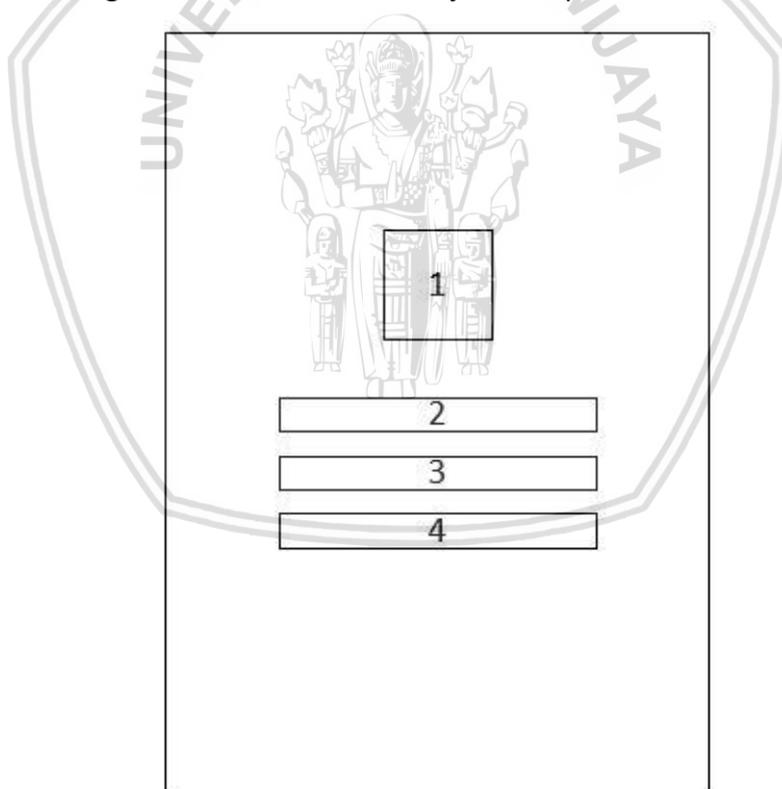
7	Post	<code>http://penjadwalan.orangeslab.com/api/volunteer/hadir</code>	<code>accompaniment_id(int)</code>	<code>boolean</code>
---	------	--	------------------------------------	----------------------

### 5.1.6 Perancangan Antarmuka

Pada perancangan antarmuka ini akan ditunjukkan beberapa rancangan antarmuka aplikasi yang ditampilkan agar mempermudah interaksi antara pengguna dan sistem. Pada proses perancangan antarmuka terdapat dua tahap yang dilakukan untuk evaluasi pengguna. Tahap pertama dalam perancangan antarmuka diambil dari identifikasi kebutuhan tahap pertama sebagai *prototype 1*. Sedangkan untuk tahap kedua diambil dari hasil evaluasi pengguna yang telah dilakukan dan menjadikan perancangan antarmuka sebagai *prototype 2*.

#### 5.1.6.1 Perancangan Antarmuka Halaman *Login*

Perancangan antarmuka *login* pada gambar 5.12, dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada tabel 5.9.



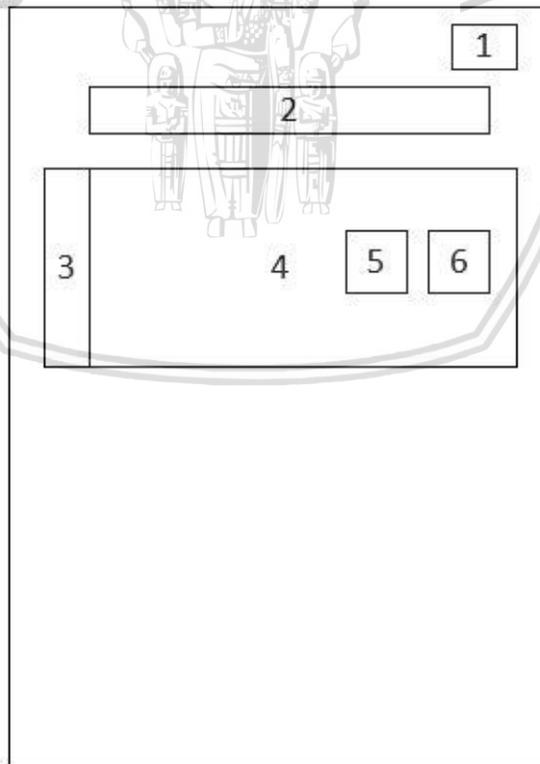
Gambar 5.13 Perancangan Antarmuka Halaman *Login Prototype 1*

Tabel 5.10 Penjelasan Antarmuka Halaman *Login Prototype 1*

No	Nama Objek	Tipe	Keterangan
1	Gambar	<i>Image View</i>	Menampilkan logo aplikasi V-Apps
2	<i>Username</i>	<i>Edit Text</i>	Memasukkan NIM sebagai <i>username</i> untuk masuk ke aplikasi
3	<i>Password</i>	<i>Edit Text</i>	Memasukkan <i>password</i> untuk masuk ke aplikasi
4	<i>Login</i>	<i>Button</i>	Tombol untuk mengautentifikasi <i>username</i> dan <i>password</i> agar bisa masuk ke aplikasi

**5.1.6.2 Perancangan Antarmuka Melihat Jadwal Damping**

Perancangan antarmuka Melihat Jadwal Damping pada gambar 5.13, dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada tabel 5.10.



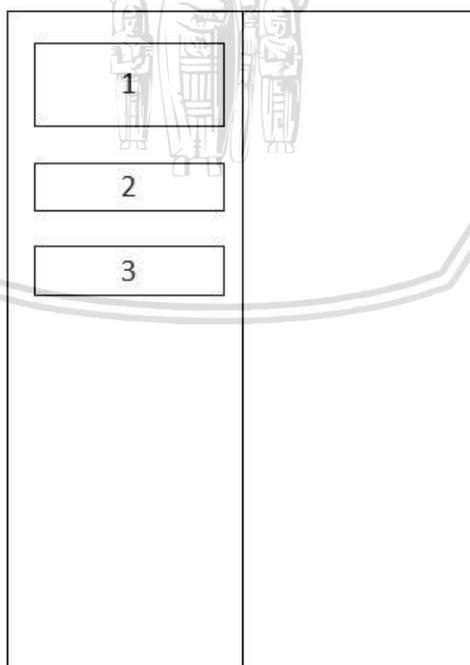
Gambar 5.14 Perancangan Antarmuka Melihat Jadwal Damping *prototype 1*

Tabel 5.11 Antarmuka Melihat Jadwal Damping *prototype 1*

No	Nama Objek	Tipe	Keterangan
1	<i>Scan</i>	<i>Image View</i>	Mengaktifkan kamera untuk <i>scan qr code</i>
2	<i>Filter</i>	<i>Text View</i>	Melakukan filter tanggal untuk melihat jadwal
3	<i>Status</i>	<i>View</i>	Menunjukkan gambar sebagai status jadwal
4	Jadwal	<i>Text View</i>	Menampilkan jadwal damping yang berisi nama, tanggal, jam, <i>status</i>
5	Telepon	<i>Image View</i>	Tombol untuk menuju aplikasi telepon
6	Pesan	<i>Image View</i>	Tombol untuk menuju ke aplikasi <i>message</i>

### 5.1.6.3 Perancangan Antarmuka Halaman *Navigation Bar* Pendamping

Perancangan antarmuka Halaman *Navigation Bar* pada gambar 5.14, dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada tabel 5.11.



Gambar 5.15 Perancangan Antarmuka Halaman *Navigation Bar* Pendamping *Prototype 1*

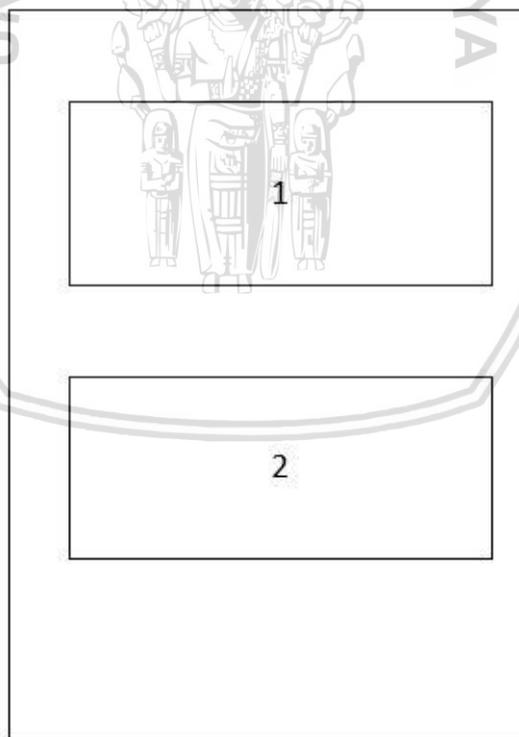
**Tabel 5.12 Antarmuka Halaman *Navigation Bar* Pendamping *Prototype 1***

No	Nama Objek	Tipe	Keterangan
1	Profil	<i>Text View</i>	Menampilkan pengguna yang <i>login</i>
2	Jadwal	<i>Text View</i>	Menampilkan jadwal sesuai dengan kebutuhan pengguna
3	<i>Logout</i>	<i>Text View</i>	Mengeluarkan pengguna dari sistem aplikasi V-Apps

Setelah dilakukan evaluasi pengguna, terdapat penambahan fitur yaitu melihat profil difabel. Pendamping dapat melihat lebih detail mahasiswa difabel yang pendamping dampingi dan jadwal yang lebih lengkap.

#### 5.1.6.4 Perancangan Antarmuka Melihat Profil Difabel

Perancangan antarmuka Halaman Melihat Profil Difabel pada gambar 5.15, dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada tabel 5.12.

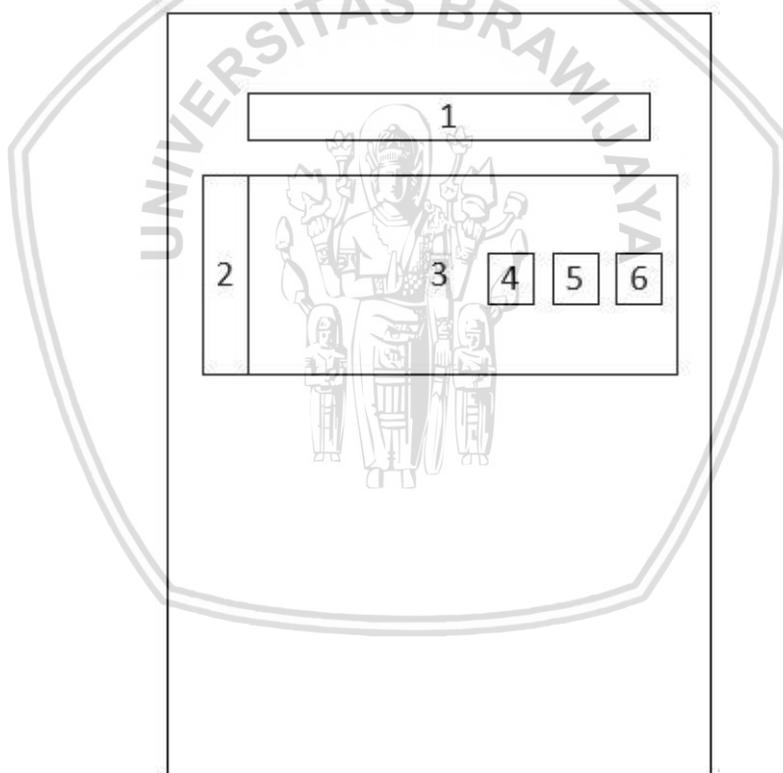
**Gambar 5.16 Perancangan Antarmuka Melihat Profil Difabel *prototype 1***

**Tabel 5.13 Antarmuka Melihat Profil Difabel *Prototype 1***

No	Nama Objek	Tipe	Keterangan
1	Data Mahasiswa	<i>Text View</i>	Menampilkan data mahasiswa yang didampingi
2	Jadwal	<i>Text View</i>	Menampilkan detail jadwal sesuai dengan kebutuhan pengguna

**5.1.6.5 Perancangan Antarmuka Melihat Jadwal Pendampingan**

Perancangan antarmuka Halaman Melihat Profil Difabel pada gambar 5.16, dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada tabel 5.13.



**Gambar 5.17 Perancangan Antarmuka Melihat Jadwal Pendampingan *Prototype 1***

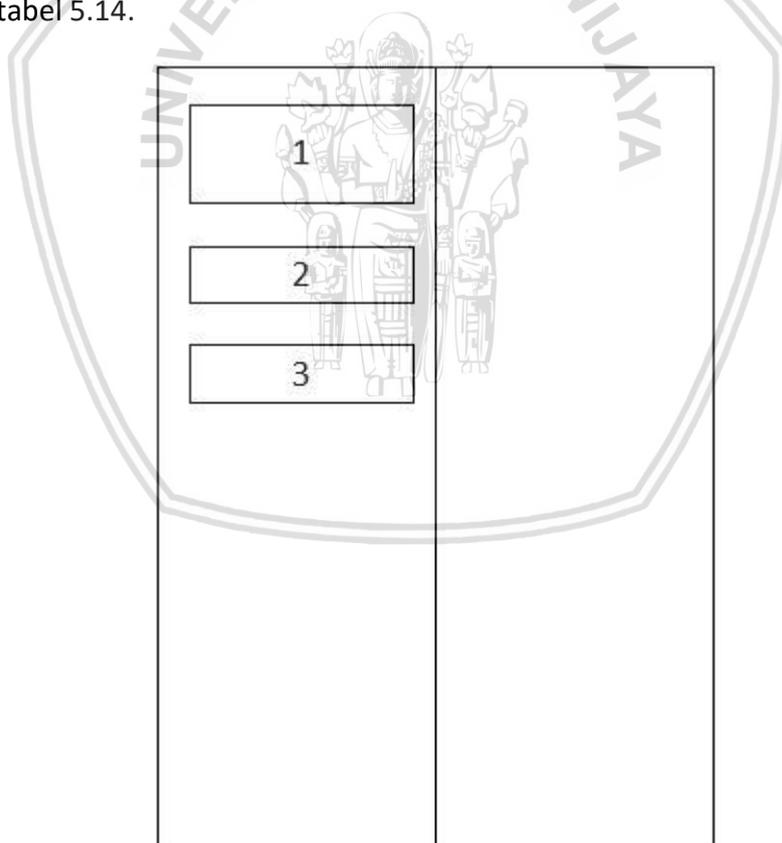
**Tabel 5.14 Antarmuka Melihat Jadwal Pendampingan *Prototype 1***

No	Nama Objek	Tipe	Keterangan
1	Filter	<i>Text View</i>	Melakukan filter tanggal untuk melihat jadwal

2	<i>Status</i>	<i>View</i>	Menunjukkan warna sebagai status jadwal
2	Jadwal	<i>Text View</i>	Menampilkan jadwal pendampingan yang berisi nama, tanggal, jam, <i>status</i>
4	Telepon	<i>Image View</i>	Tombol untuk menuju aplikasi telepone
5	Pesan	<i>Image View</i>	Tombol untuk menuju ke aplikasi <i>message</i>
6	<i>Scan</i>	<i>Image View</i>	Menampilkan <i>qr code</i> yang dimiliki mahasiswa difabel

#### 5.1.6.6 Perancangan Antarmuka Halaman *Navigation Bar* Difabel

Perancangan antarmuka Halaman Melihat Profil Difabel pada gambar 5.17, dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada tabel 5.14.



Gambar 5.18 Perancangan Antarmuka Halaman *Navigation Bar* Prototype 1

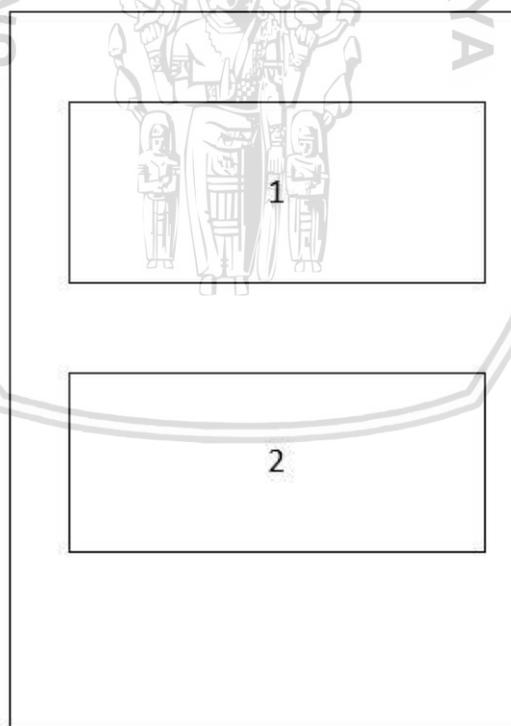
**Tabel 5.15 Antarmuka Halaman *Navigation Bar Prototype 1***

No	Nama Objek	Tipe	Keterangan
1	Profil	<i>Text View</i>	Menampilkan pengguna yang <i>login</i>
2	Jadwal	<i>Text View</i>	Menampilkan jadwal sesuai dengan kebutuhan pengguna
3	<i>Logout</i>	<i>Text View</i>	Mengeluarkan pengguna dari sistem aplikasi V-Apps

Setelah dilakukan evaluasi pengguna, terdapat penambahan fitur yaitu melihat profil difabel. Pendamping dapat melihat lebih detail mahasiswa difabel yang pendamping dampingi dan jadwal yang lebih lengkap.

#### 5.1.6.7 Perancangan Antarmuka Melihat Profil Pendamping

Perancangan antarmuka Halaman Melihat Profil Difabel pada gambar 5.18, dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada tabel 5.15.

**Gambar 5.19 Perancangan Antarmuka Melihat Profil Pendamping *Prototype***

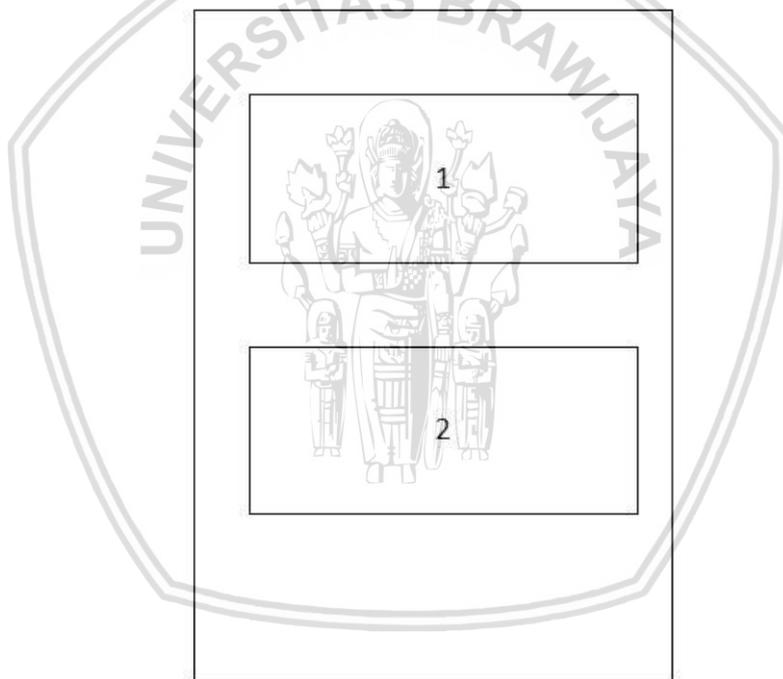
2

**Tabel 5.16 Antarmuka Melihat Profil Pendamping *Prototype 2***

No	Nama Objek	Tipe	Keterangan
1	Data Mahasiswa	<i>Text View</i>	Menampilkan data mahasiswa yang didampingi
2	Jadwal	<i>Text View</i>	Menampilkan detail jadwal sesuai dengan kebutuhan pengguna

**5.1.6.8 Perancangan Antarmuka Halaman *Review***

Perancangan antarmuka Halaman *Review* pada gambar 5.19, dan penjelasan detail dari perancangan antarmuka tersebut dijelaskan pada tabel 5.16.



**Gambar 5.20 Perancangan Antarmuka Halaman *Review Prototype 1***

**Tabel 5.17 Perancangan Antarmuka Halaman *Review Prototype 1***

No	Nama Objek	Tipe	Keterangan
1	<i>RatingBar</i>	<i>RatingBar</i>	Menampilkan rating yang dapat diberi oleh mahasiswa difabel
2	<i>Submit</i>	<i>Button</i>	Sebagai tombol untuk memproses data <i>review</i> yang diberikan

## 5.2 Implementasi

Pada bab implementasi akan dijelaskan mengenai implementasi yang dilakukan dalam pembuatan sistem. Sebelum implementasi dijelaskan spesifikasi yang digunakan dalam implementasi sistem. Spesifikasi tersebut berupa spesifikasi sistem yang digunakan, serta batasan dalam implementasi. Implementasi yang dilakukan adalah implementasi basis data, implementasi kode program, dan implementasi antarmuka.

### 5.2.1 Spesifikasi Sistem

Dalam mengimplementasikan hasil dan analisis kebutuhan, perancangan, dan hasil dari evaluasi pengguna dibutuhkan perangkat keras serta perangkat lunak. Perangkat keras dan perangkat lunak perlu dispesifikasikan agar dapat mengetahui standar yang diperlukan untuk pembuatan aplikasi selama dalam penelitian

#### 5.2.1.1 Spesifikasi Perangkat Keras

Dalam implementasi aplikasi V-Apps, dibutuhkan perangkat keras berupa komputer dan *smartphone*. Komputer digunakan untuk melakukan implementasi atau pembuatan aplikasi mulai dari implementasi antarmuka hingga implementasi algoritme. Sementara *smartphone* digunakan untuk pengujian atau perangkat penguji aplikasi V-Apps. Spesifikasi perangkat keras komputer yang digunakan dalam implementasi dapat dilihat pada pada Tabel 5.16.

**Tabel 5.18 Spesifikasi Perangkat Keras Komputer**

No	Nama Komponen	Spesifikasi
1	Model	ASUS A43S
2	<i>Processor</i>	Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz
3	<i>Memory</i>	12GB RAM
4	<i>Display</i>	NVIDIA GeForce GT 630M

Spesifikasi perangkat keras *smartphone* yang digunakan dapat dilihat pada tabel 5.17

**Tabel 5.19 Spesifikasi Perangkat Keras Smartphone**

No	Nama Komponen	Spesifikasi
1	Model	Xiaomi Redmi Note 3
2	<i>Processor</i>	Qualcom MSM8956 Snapdragon 650, Hexa-Core Max 1.85GHz

3	<i>Memory</i>	2GB RAM, 16GB internal memory
4	<i>Display</i>	5 inch

### 5.2.1.2 Spesifikasi Perangkat Lunak

Dalam implementasi aplikasi V-App, dibutuhkan perangkat lunak yang terdapat dalam komputer dan *smartphone*. Spesifikasi perangkat lunak komputer yang digunakan dalam implementasi dapat dilihat pada tabel 5.18

**Tabel 5.20 Spesifikasi Perangkat Lunak Komputer**

No	Nama Komponen	Spesifikasi
1	Sistem Operasi	<i>Windows 10 Pro Enterprise 64 bit</i>
2	Bahasa Pemrograman	<i>Java</i>
3	Editor Pemrograman	<i>Androdi Studio</i>
4	Editor Dokumentasi	<i>Microsoft Office Word 2016</i>
5	Database Server	<i>MySql, JSON</i>

Sedangkan untuk spesifikasi perangkat lunak *smartphone* dapat dilihat pada tabel 5.19

**Tabel 5.21 Spesifikasi Perangkat Lunak Smartphone**

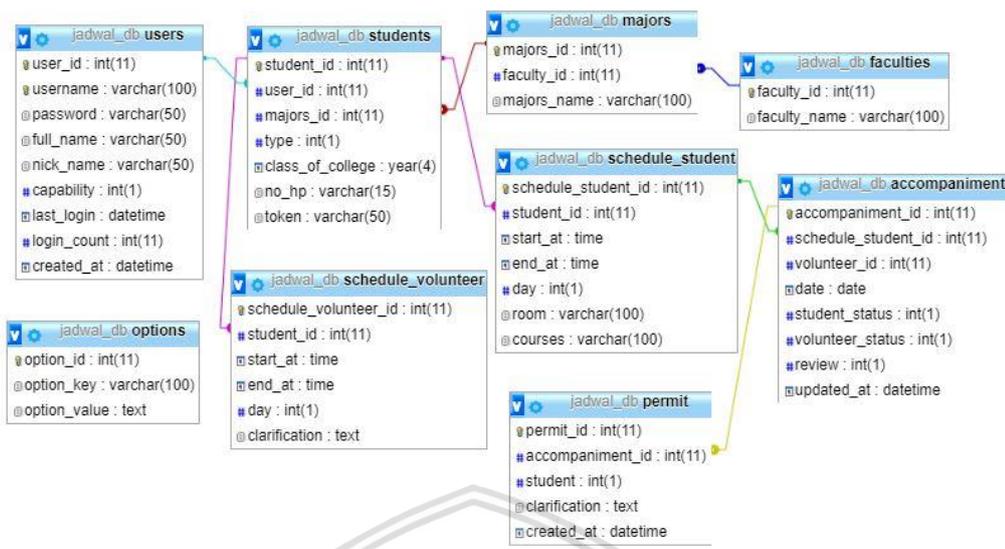
No	Nama Komponen	Spesifikasi
1	Sistem Operasi	Android 6.0.1 MMB29M

### 5.2.2 Implementasi Basis Data

Implementasi basis data memiliki dua bagian yaitu implementasi *database* yang dimiliki oleh sistem penjadwalan jadwal damping PSLD UB. Implementasi *database* diambil dari sistem penjadwalan jadwal damping PSLD UB. Serta implementasi penggunaan API untuk aplikasi V-Apps. API dibuat oleh pihak kantor PSLD UB sesuai dengan kebutuhan aplikasi.

#### 5.2.2.1 Implementasi Database Sistem PSLD UB

Pada implementasi basis data dibuat sebuah *database* sesuai dengan perancangan basis data pada gambar 5.11. Gambar 5.20 merupakan hasil implementasi basis data yang dibuat dengan MySQL.



Gambar 5.21 Implementasi Basis Data

5.2.2.2 Implementasi Penggunaan API

a. Hasil Penggunaan Login

Method : Post

Url : http://penjadwalan.orangeslab.com/api/login

Body :

- username : 11111
- password : 12345678

Hasil :

```

{
  "user_id": "44",
  "full_name": "Febrina Fitria Ningruma",
  "nick_name": "Febri",
  "student_id": "36",
  "type": "volunteer",
  "no_hp": "081228534015",
  "class_of_college": "2015",
  "faculty_name": "Fakultas Perikanan dan Ilmu Kelautan",
  "majors_name": "Teknologi Hasil Perikanan",
  "token": "d915d4fde947a2621e0a7bd9530d6f4ccb023df8"
}
    
```



**b. Hasil Penggunaan Jadwal Volunteer**

*Method* : Get

*Url* :

http://penjadwalan.orangeslab.com/api/volunteer/jadwal?start=2018-01-01&end=2018-05-05

*Headers* :

- *Studentid* : 36
- *Token* : d915d4fde947a2621e0a7bd9530d6f4ccb023df8

*Hasil* :

```
{
  "accompaniment_id": "1471",
  "schedule_student_id": "49",
  "volunteer_id": "36",
  "date": "2018-05-01",
  "student_status": "1",
  "volunteer_status": "1",
  "review": "0",
  "updated_at": "2018-04-30 14:57:32",
  "start_at": "11:10:00",
  "end_at": "12:55:00",
  "day": "2",
  "room": "MP 2.5",
  "courses": "Basis Data",
  "student_id": "31",
  "full_name": "Aditya Ilham Pratama",
  "nick_name": "Tuna Rungu",
  "faculty_name": "Fakultas Matematika dan Ilmu
  Pengetahuan Alam",
  "majors_name": "Matematika",
  "class_of_college": "2015",
  "no_hp": "081228534015",
  "volunteer_student_id": "36",
  "volunteer_full_name": "Febrina Fitria
  Ningruma",
  "volunteer_nick_name": "Febri",
  "volunteer_faculty_name": "Fakultas Perikanan
  dan Ilmu Kelautan",
```

```

        "volunteer_majors_name": "Teknologi Hasil
        Perikanan",
        "volunteer_class_of_college": "2015",
        "volunteer_no_hp": "081228534015"
    },

```

### c. Hasil Penggunaan Hadir *Volunteer*

*Method* : Post

*Url* : <http://penjadwalan.orangeslab.com/api/volunteer/hadir>

*Body* :

- *Accompaniment\_id* : 1471

*Headers* :

- *Studentid* : 36
- *Token* : 5e205c7b159619326dbe29819c488f9f8ef6f58d

*Hasil* :

```
False
```

### d. Hasil Penggunaan Jadwal Mahasiswa

*Method* : Get

*Url* :

<http://penjadwalan.orangeslab.com/api/mahasiswa/jadwal?start=2018-01-05&end=2018-05-05>

*Headers* :

- *Student\_id* : 31
- *Token* : badda60703c91a4cc7984c79f769e4e5f007686f

*Hasil* :

```

{
    "accompaniment_id": "1472",
    "schedule_student_id": "50",
    "volunteer_id": "40",
    "date": "2018-05-01",
    "student_status": "1",

```

```

"volunteer_status": "0",
"review": "0",
"updated_at": "2018-04-30 14:57:32",
"start_at": "13:00:00",
"end_at": "14:45:00",
"day": "2",
"room": "MC 3.5",
"courses": "Kewarganegaraan",
"student_id": "31",
"full_name": "Aditya Ilham Pratama",
"nick_name": "Tuna Rungu",
"faculty_name": "Fakultas Matematika dan
Ilmu Pengetahuan Alam",
"majors_name": "Matematika",
"class_of_college": "2015",
"no_hp": "081228534015",
"volunteer_student_id": "40",
"volunteer_full_name": "Sangidil Kurdi",
"volunteer_nick_name": "Sani",
"volunteer_faculty_name": "Fakultas
Perternakan",
"volunteer_majors_name": "Peternakan",
"volunteer_class_of_college": "2015",
"volunteer_no_hp": "081228534015"
},

```

#### e. Hasil Penggunaan Hadir Mahasiswa

*Method* : Post

*Url* : <http://penjadwalan.orangeslab.com/api/mahasiswa/hadir>

*Body* :

- *Accompaniment\_id* : 1492

*Headers* :

- *Studentid* : 31
- *Token* : badda60703c91a4cc7984c79f769e4e5f007686f

*Hasil* :

False

**f. Hasil Penggunaan Review Mahasiswa**

*Method* : Post

*Url* : http://penjadwalan.orangeslab.com/api/mahasiswa/review

*Body* :

- *Accompaniment\_id* : 1492
- *Review* : 0

*Headers* :

- *Studentid* : 31
- *Token* : badda60703c91a4cc7984c79f769e4e5f007686f

*Hasil* :

```
{
  "accompaniment_id": "1492",
  "schedule_student_id": "52",
  "volunteer_id": "37",
  "date": "2018-05-04",
  "student_status": "4",
  "volunteer_status": "4",
  "review": "0",
  "updated_at": "2018-04-30 14:57:32"
}
```

**g. Hasil Penggunaan Logout**

*Method* : Post

*Url* : http://penjadwalan.orangeslab.com/api/logout

*Headers* :

- *Studentid* : 31
- *Token* : badda60703c91a4cc7984c79f769e4e5f007686f

*Hasil* :

```
"Success logout"
```

**5.2.3 Implementasi Kode Program**

Setelah proses analisis kebutuhan selesai dilakukan, tahap berikutnya adalah implementasi kode program seperti dibawah ini :

### 5.2.3.1 Implementasi Kode Program *loadVolunteerSchedule*

Nama Kelas : *ScheduleFragment*

Nama Method: *loadVolunteerSchedule*

```

void loadVolunteerSchedule() {
    String start = new SimpleDateFormat("yyyy-MM-dd",
    Locale.ENGLISH).format(calendarStart.getTime());
    String end = new SimpleDateFormat("yyyy-MM-dd",
    Locale.ENGLISH).format(calendarEnd.getTime());

    Call<ArrayList<JadwalResponse>> volunteerScheduleCall =
    apiInterface.jadwalVolunteer(start,end);
    volunteerScheduleCall.enqueue(new
    Callback<ArrayList<JadwalResponse>>() {
        @Override
        public void
    onResponse(Call<ArrayList<JadwalResponse>> call,
    Response<ArrayList<JadwalResponse>> response) {
            if(response.body() != null){
                volunteerSchedules = response.body();
                LinearLayoutManager layoutManager = new
                LinearLayoutManager(getActivity());
                VolunteerScheduleAdapter
                volunteerScheduleAdapter = new
                VolunteerScheduleAdapter(getActivity(), volunteerSchedules,
                userType, ScheduleFragment.this);
                rv_schedule.setLayoutManager(layoutManager);

                rv_schedule.setAdapter(volunteerScheduleAdapter);
                swipeRefreshLayout.setRefreshing(false);
            } else{
                Toast toast = Toast.makeText(getActivity(),
                response.message(),Toast.LENGTH_LONG);
                toast.show();
            }
        }
    }
}

```

### 5.2.3.2 Implementasi Kode Program *loadMahasiswaSchedule*

Nama Kelas : *ShceduleFragment*

Nama Method: *loadMahasiswaSchedule*

```

void loadMahasiswaSchedule() {
    String start = new SimpleDateFormat("yyyy-MM-dd",
    Locale.ENGLISH).format(calendarStart.getTime());
    String end = new SimpleDateFormat("yyyy-MM-dd",
    Locale.ENGLISH).format(calendarEnd.getTime());

    final Call<ArrayList<JadwalResponse>>
    volunteerScheduleCall =
    apiInterface.jadwalMahasiswa(start,end);
    volunteerScheduleCall.enqueue(new
    Callback<ArrayList<JadwalResponse>>() {
        @Override

```

```

        public void
onResponse(Call<ArrayList<JadwalResponse>> call,
Response<ArrayList<JadwalResponse>> response) {
    if(response.body() != null){
        volunteerSchedules = response.body();
        checkVolunteerConfirmation(volunteerSchedules);
        LinearLayoutManager layoutManager = new
LinearLayoutManager(getActivity());
        MahasiswaScheduleAdapter
mahasiswaScheduleAdapter = new
MahasiswaScheduleAdapter(getActivity(), volunteerSchedules,
userType, ScheduleFragment.this);

rv_schedule.setLayoutManager(layoutManager);

rv_schedule.setAdapter(mahasiswaScheduleAdapter);
        swipeRefreshLayout.setRefreshing(false);

        } else{
            Toast toast =
Toast.makeText(getActivity(),
response.message(), Toast.LENGTH_LONG);
            toast.show();
        }
    }
}

```

### 5.2.3.3 Implementasi Kode Program *onActivityResult*

Nama Kelas : *ScheduleFragment*

Nama Method: *onActivityResult*

```

super.onActivityResult(requestCode, resultCode, data);
    IntentResult result =
IntentIntegrator.parseActivityResult(requestCode, resultCode,
data);
    if(result != null) {
        if(result.getContents() == null) {
            Toast.makeText(getActivity(), "Maaf data tidak
dapat diproses.", Toast.LENGTH_LONG).show();
        } else {
            String jadwalId = result.getContents();

            Call<String> volunteerKonfirmasiCall =
apiInterface.konfirmasiVolunteer(new
HadirInformation(Integer.valueOf(jadwalId)));
            volunteerKonfirmasiCall.enqueue(new
Callback<String>() {
                @Override
                public void onResponse(Call<String> call,
Response<String> response) {
                    if(response.body() != null){
                        Toast toast =
Toast.makeText(getActivity(), "Pendampingan telah
dikonfirmasi", Toast.LENGTH_LONG);
                        toast.show();
                        loadVolunteerSchedule();
                    } else{

```

```

        Toast toast =
        Toast.makeText(getActivity(),
        response.message(), Toast.LENGTH_LONG);
        toast.show();
    }
}

@Override
public void onFailure(Call<String> call,
Throwable t) {
    call.cancel();
}
});
}
} else {
    super.onActivityResult(requestCode, resultCode,
data);
}
}
}

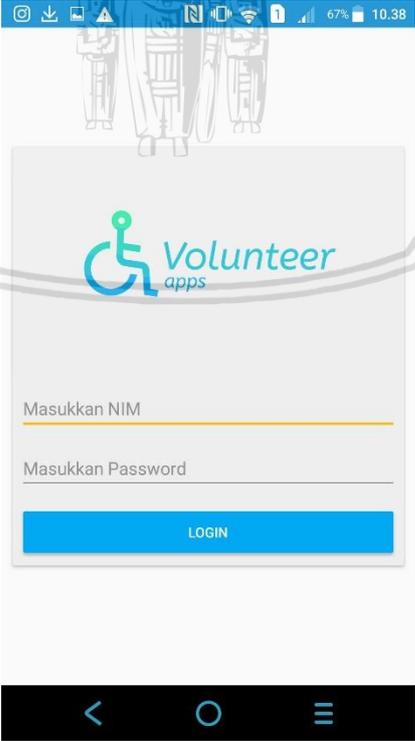
```

**5.2.4 Implementasi Antarmuka**

Pada implementasi antarmuka ini menggambarkan antarmuka yang sudah di implementasikan ke sistem.

**5.2.4.1 Implementasi Antarmuka Halaman Login**

Gambar 5.21 merupakan hasil implementasi dari perancangan antarmuka halaman *login prototype* 1 pada gambar 5.12.



**Gambar 5.22 Implementasi Antarmuka Halaman Login.**

**5.2.4.2 Implementasi Antarmuka Halaman Melihat Jadwal Damping**

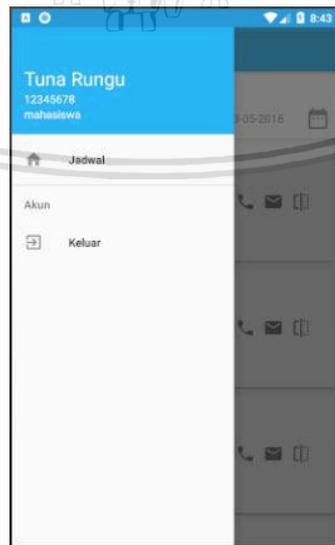
Gambar 5.22 merupakan hasil implementasi dari perancangan antarmuka halaman melihat jadwal *prototype* 1 pada gambar 5.13.



**Gambar 5.23 Implementasi Antarmuka Halaman Melihat Jadwal Damping**

**5.2.4.3 Implementasi Antarmuka Halaman *Navigation Bar* Pendamping**

Gambar 5.23 merupakan hasil implementasi dari perancangan antarmuka halaman *navigation bar* pendamping *prototype* 1 gambar 5.14.



**Gambar 5.24 Implementasi Antarmuka Halaman *Navigation Bar* Pendamping**



**5.2.4.4 Implementasi Antarmuka Halaman Melihat Profil Difabel**

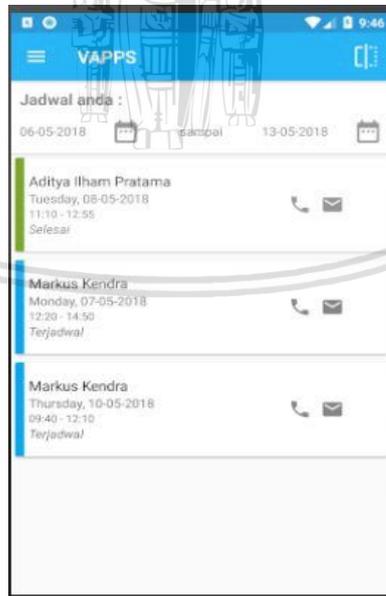
Gambar 5.24 merupakan implementasi dari hasil perancangan antarmuka halaman melihat profil difabel *prototype* 2 gambar 5.15.



**Gambar 5.25 Implementasi Antarmuka Halaman Melihat Profil Difabel**

**5.2.4.5 Implementasi Antarmuka Halaman Melihat Jadwal Pendampingan**

Gambar 5.25 merupakan implementasi dari hasil perancangan antarmuka halaman melihat jadwal pedampingan *prototype* 1 gambar 5.16.

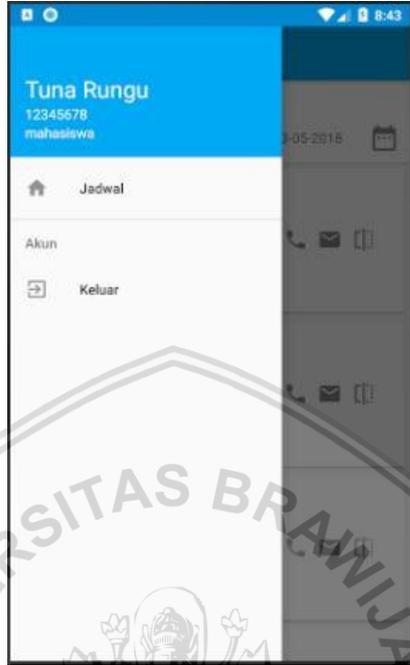


**Gambar 5. 26 Implementasi Antarmuka Halaman Melihat Jadwal Pendampingan**



**5.2.4.6 Implementasi Antarmuka Halaman *Navigation Bar* Difabel**

Gambar 5.26 merupakan hasil implementasi dari perancangan antarmuka halaman *navigation bar* difabel *prototype 1* gambar 5.17.



**Gambar 5.27 Implementasi Antarmuka Halaman *Navigation Bar* Dofabel.**

**5.2.4.7 Implementasi Antarmuka Halaman Melihat Profil Pendamping**

Gambar 5.27 merupakan hasil implementasi dari perancangan antarmuka halaman melihat profil pendamping *prototype 2* gambar 5.18.

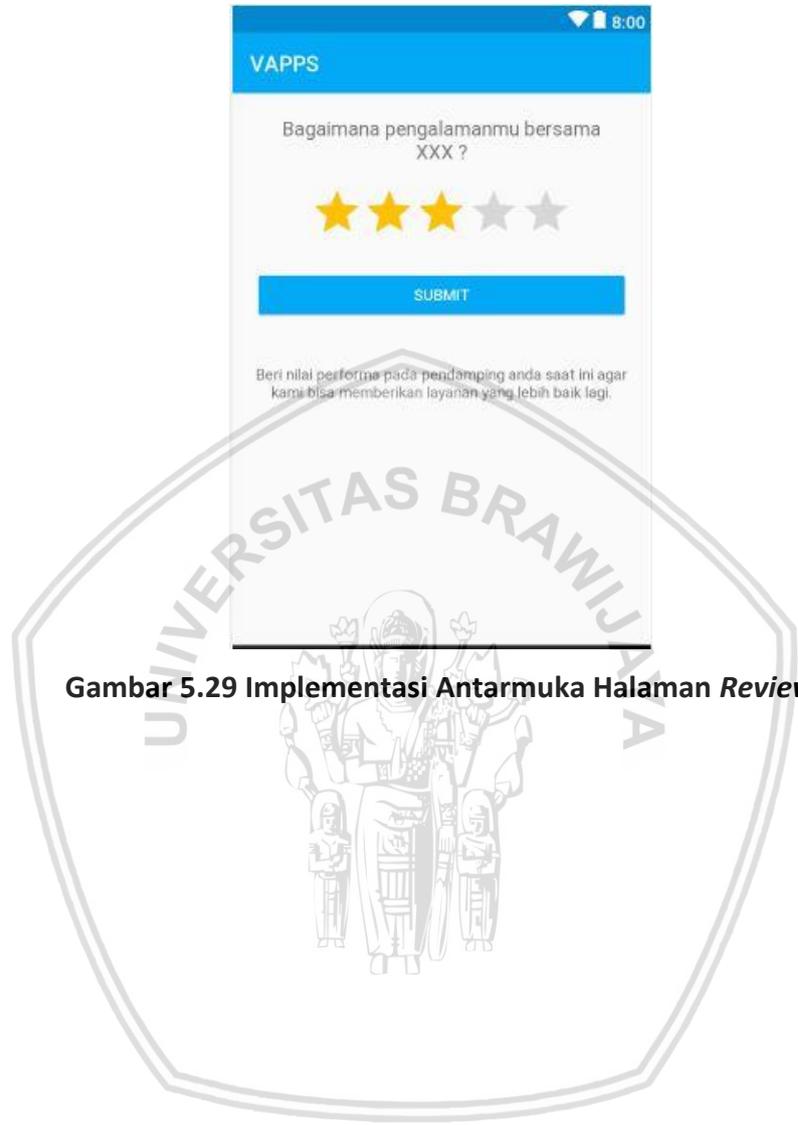


**Gambar 5.28 Implementasi Antarmuka Halaman Melihat Profil Pendamping**



### 5.2.4.8 Implementasi Antarmuka Halaman *Review*

Gambar 5.28 merupakan hasil implementasi dari perancangan antarmuka halaman *review prototype* 1 gambar 5.19.



Gambar 5.29 Implementasi Antarmuka Halaman *Review*



## BAB 6 PENGUJIAN

Tahap pengujian dilakukan setelah sudah melakukan implementasi sistem. Tujuan dari pengujian untuk memeriksa apakah implementasi sudah sesuai dengan analisis kebutuhan dan perancangan sistem atau tidak. Tahap pengujian yang akan dilakukan adalah pengujian fungsional yang terdiri atas pengujian unit dan pengujian validasi. Dan pengujian non fungsional yang diuji adalah *usability*.

### 6.1 Pengujian Fungsional

Dalam pengujian fungsional terdapat dua bagian yaitu pengujian unit dan pengujian validasi. Pengujian unit digunakan untuk menguji tiap *method* dalam fungsi utama aplikasi. Sedangkan untuk pengujian validasi adalah pengujian yang dilakukan untuk menguji kebutuhan fungsional aplikasi.

#### 6.1.1 Pengujian Unit

Pengujian unit adalah pengujian yang dilakukan untuk memastikan bahwa implementasi telah sesuai dengan yang diinginkan melalui pengujian algoritme atau method yang telah dibangun. Pengujian unit menggunakan *basis path*. *Method* yang akan diuji dalam pengujian *basis path* adalah *method loadVolunteerSchedule(), loadMahasiswaSchedule(), onActivityResult()*.

##### 6.1.1.1 Pengujian Unit loadVolunteerSchedule

###### 1. Kode Program

```

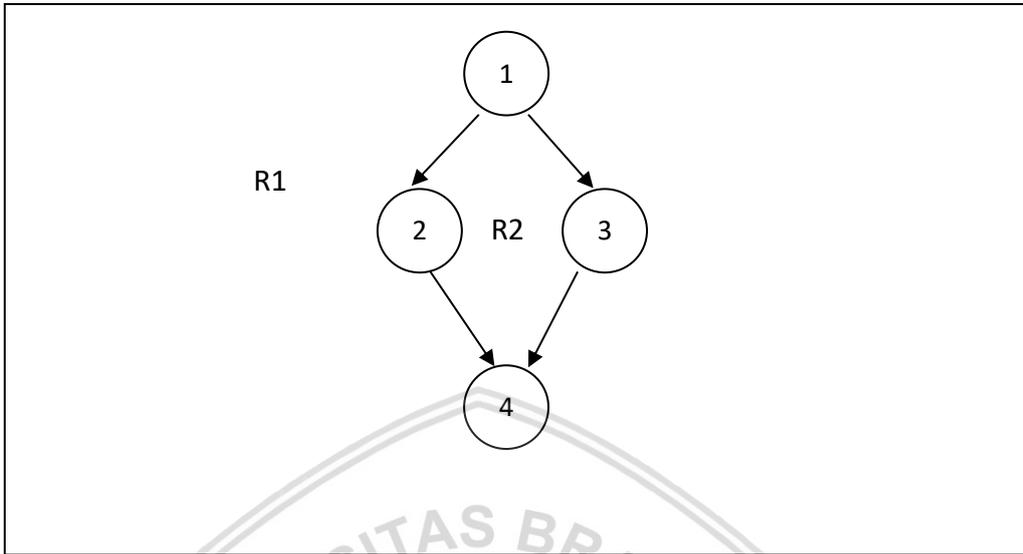
Call<ArrayList<JadwalResponse>> volunteerScheduleCall =
apiInterface.jadwalVolunteer(start,end);
volunteerScheduleCall.enqueue(new
Callback<ArrayList<JadwalResponse>>() {
public void onResponse(Call<ArrayList<JadwalResponse>>
call, Response<ArrayList<JadwalResponse>> response) {
    if(response.body() != null){
        volunteerSchedules = response.body();
        LinearLayoutManager layoutManager = new
        LinearLayoutManager(getActivity());
        VolunteerScheduleAdapter volunteerScheduleAdapter =
new VolunteerScheduleAdapter(getActivity(),
volunteerSchedules, userType, ScheduleFragment.this);
rv_schedule.setLayoutManager(layoutManager);
rv_schedule.setAdapter(volunteerScheduleAdapter);
swipeRefreshLayout.setRefreshing(false);
    } else{
        Toast toast = Toast.makeText(getActivity(),
response.message(), Toast.LENGTH_LONG);
        toast.show();
    }
}}
    
```

Gambar 6.1 Kode Program *method loadVolunteerSchedule()*



2. Basis Path Testing

a. Flow Graph



Gambar 6.2 Flow Graph method loadVolunteerSchedule()

Setelah pembuatan *flow graph*, tahap berikutnya merupakan perhitungan *cyclomatic complexity*. Berdasarkan *flow graph method loadVolunteerSchedule()* maka dapat dihitung nilai *cyclomatic complexity* sebagai berikut:

b. Cyclomatic Complexity

1.  $V(G) = \text{Jumlah Region}$   
 $= 2$
2.  $V(G) = \text{edges} - \text{node} + 2$   
 $= 4 - 4 + 2$   
 $= 2$
3.  $V(G) = \text{predicate node} + 1$   
 $= 1 + 1$   
 $= 2$

Perhitungan *cyclomatic complexity* yang telah dilakukan menghasilkan nilai 2, maka dapat ditentukan jalur independen sebagai berikut:

c. Independent

1. Jalur 1 : 1-2-4
2. Jalur 2 : 1-3-4



d. Desain *Test Case*

Berdasarkan jalur identifikasi, disusun *test case* untuk menguji unit aplikasi yang dijelaskan pada Tabel 6.1.

**Tabel 6.1 Desain *Test Case loadVolunteerSchedule***

No	No Jalur	Test Case	Expected Result
1	1	Sistem memanggil <i>method loadVolunteerSchedule</i> dengan menggunakan akun yang memiliki jadwal	Sistem menampilkan jadwal kepada pengguna
2	2	Sistem memanggil <i>method loadVolunteerSchedule</i> dengan menggunakan akun yang tidak memiliki jadwal	Sistem menampilkan jadwal kosong pada pengguna

e. Hasil Pengujian

Hasil dari setiap *test case* yang di definisikan pada Tabel 6.2

**Tabel 6.2 Hasil *Test Case loadVolunteerSchedule***

No Test Case	Result	Status
1	Sistem menampilkan jadwal kepada pengguna	Valid
2	Sistem menampilkan jadwal kosong pada pengguna	Valid

**6.1.1.2 Pengujian Unit *loadMahasiswaSchedule***

1. Kode Program

```

Call<ArrayList<JadwalResponse>> volunteerScheduleCall =
apiInterface.jadwalMahasiswa(start,end);
volunteerScheduleCall.enqueue(new
Callback<ArrayList<JadwalResponse>>() {

public void onResponse(Call<ArrayList<JadwalResponse>>
call, Response<ArrayList<JadwalResponse>> response) {
    if(response.body() != null){
        volunteerSchedules = response.body();
        checkVolunteerConfirmation(volunteerSchedules);
        LinearLayoutManager layoutManager = new
        LinearLayoutManager(getActivity());
        MahasiswaScheduleAdapter mahasiswaScheduleAdapter =
        new MahasiswaScheduleAdapter(getActivity(),
        volunteerSchedules, userType, ScheduleFragment.this);

        rv_schedule.setLayoutManager(layoutManager);
        rv_schedule.setAdapter(mahasiswaScheduleAdapter);
        swipeRefreshLayout.setRefreshing(false);
    }
}
}
    
```

1

2



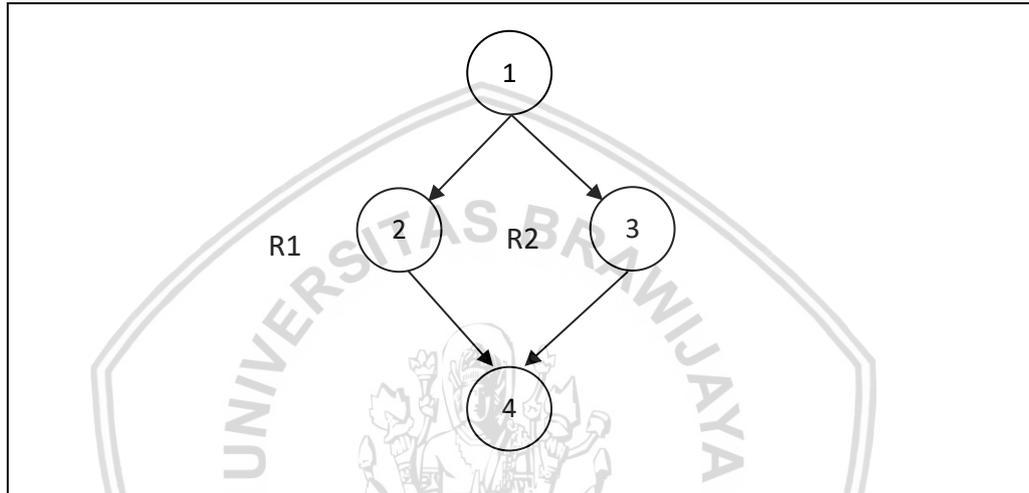
```

} else{
    Toast toast =
    Toast.makeText(getActivity(),
    response.message(), Toast.LENGTH_LONG);
    toast.show();
}
}
    
```

Gambar 6.3 Kode Program *method loadMahasiswaSchedule()*

2. Basis Path Testing

a. Flow Graph



Gambar 6.4 Flow Graph *method loadMahasiswaSchedule()*

Setelah pembuatan *flow graph*, tahap berikutnya merupakan perhitungan *cyclomatic complexity*. Berdasarkan *flow graph method loadMahasiswaSchedule()* maka dapat dihitung nilai *cyclomatic complexity* sebagai berikut:

b. Cyclomatic Complexity

1.  $V(G) = \text{Jumlah Region}$   
 $= 2$
2.  $V(G) = \text{edges} - \text{node} + 2$   
 $= 4 - 4 + 2$   
 $= 2$
3.  $V(G) = \text{predicate node} + 1$   
 $= 1 + 1$   
 $= 2$

Perhitungan *cyclomatic complexity* yang telah dilakukan menghasilkan nilai 2, maka dapat ditentukan jalur independen sebagai berikut:

c. Independent

1. Jalur 1 : 1-2-4
2. Jalur 2 : 1-3-4

d. Desain Test Case

Berdasarkan jalur identifikasi, disusun *test case* untuk menguji unit aplikasi yang dijelaskan pada Tabel 6.3.

**Tabel 6.3 Desain Test Case loadMahasiswaSchedule**

No	No Jalur	Test Case	Expected Result
1	1	Sistem memanggil <i>method loadMahasiswaSchedule</i> dengan menggunakan akun yang memiliki jadwal	Sistem menampilkan jadwal kepada pengguna
2	2	Sistem memanggil <i>method loadMahasiswaSchedule</i> dengan menggunakan akun yang tidak memiliki jadwal	Sistem menampilkan jadwal kosong pada pengguna

e. Hasil Test Case

Hasil dari tiap *test case* yang di definisikan pada Tabel 6.4

**Tabel 6.4 Hasil Test Case loadMahasiswaSchedule**

No Test Case	Result	Status
1	Sistem menampilkan jadwal kepada pengguna	Valid
2	Sistem menampilkan jadwal kosong pada pengguna	Valid

**6.1.1.3 Pengujian Unit onActivityResult**

1. Kode Program

```

super.onActivityResult(requestCode, resultCode, data);
IntentResult result =
IntentIntegrator.parseActivityResult(requestCode,
resultCode, data);
if(result != null) {
    if(result.getContents() == null) {
        Toast.makeText(getActivity(),
"Maaf data tidak dapat diproses.",
Toast.LENGTH_LONG).show();
    }
}
    
```



```

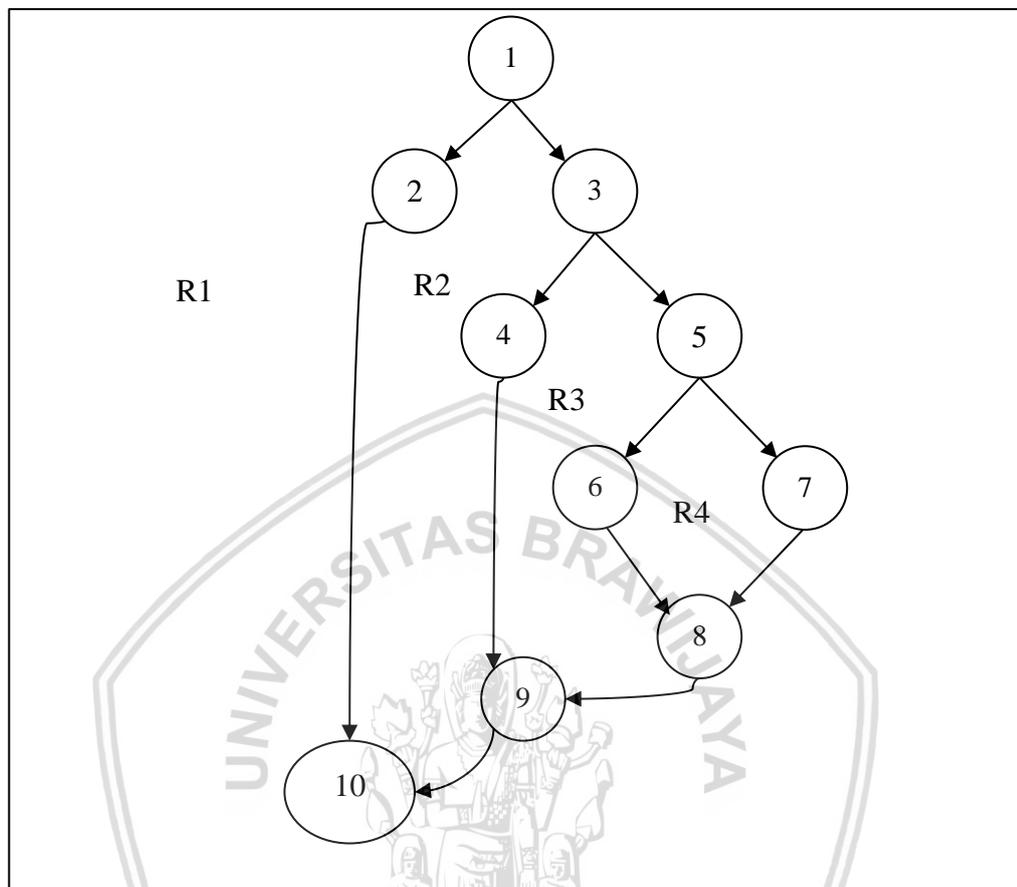
    } else {
        String jadwalId = result.getContents();
        Call<String> volunteerKonfirmasiCall =
        apiInterface.konfirmasiVolunteer(new
        HadirInformation(Integer.valueOf(jadwalId)));
        volunteerKonfirmasiCall.enqueue(new
        Callback<String>() {
            public void onResponse(Call<String> call,
            Response<String> response) {
                if(response.body() != null){
                    Toast toast = Toast.makeText(getActivity(),
                    "Pendampingan telah dikonfirmasi",Toast.LENGTH_LONG);
                    toast.show();
                    loadVolunteerSchedule();
                }
            }
        });
        else{
            Toast toast =
            Toast.makeText(getActivity(),
            response.message(),Toast.LENGTH_LONG);
            toast.show();
        }
        public void onFailure(Call<String> call, Throwable t) {
            call.cancel();
        }
    });
}
else { super.onActivityResult
(requestCode, resultCode, data);
}}

```

Gambar 6.5 Kode Program *onActivityResult()*

2. Basis Path Testing

b. Flow Graph



Gambar 6.6 Flow Graph method `onActivityResult()`

Setelah pembuatan *flow graph*, tahap berikutnya merupakan perhitungan *cyclomatic complexity*. Berdasarkan *flow graph method onActivityResult()* maka dapat dihitung nilai *cyclomatic complexity* sebagai berikut:

b. Cyclomatic Complexity

1.  $V(G)$  = Jumlah Region  
= 4
2.  $V(G)$  =  $edges - nodes + 2$   
=  $12 - 10 + 2$   
= 4
3.  $V(G)$  =  $predicate\ nodes + 1$   
=  $3 + 1$   
= 4



c. *Independent*

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

d. Desain *Test Case*

Berdasarkan jalur identifikasi, disusun *test case* untuk menguji unit aplikasi yang dijelaskan pada Tabel 6.5.

**Tabel 6.5 Desain *Test Case onActivityResult***

No	No Jalur	<i>Test Case</i>	<i>Expected Result</i>
1	1	Pengguna menekan tomnol <i>scan</i> serta tidak memberikan izin akses kamera	Sistem tidak dapat menampilkan kamera untuk melakukan <i>scan</i>
2	2	Pengguna menekan menekan tombol <i>scan</i> dan memebri izin akses kamera dan melakukan <i>scan qr code</i> yang salah	Sistem tidak dapat memproses data <i>qr code</i> dan menampilkan "maaf data tidak dapat diproses"
3	3	Pengguna menekan tombol <i>scan</i> dan memberi izin akses kamera dan melakukan <i>scan qr code</i> yang sesuai jadwal	Sistem dapat melakukan <i>scan</i> dan dapat memproses data sehingga sistem menampilkan "penampingan telah dikonfirmasi"
4	4	Pengguna menekan tombol <i>scan</i> dan memberi izin akses kamera dan melakukan <i>scan qr code</i> yang tidak sesuai jadwal	Sistem dapat memproses data tetapi API tidak dapat memproses data.

## e. Hasil Pengujian

Hasil pengujian *test case* dapat dilihat pada tabel 6.6.

**Tabel 6.6 Hasil Test Case onActivityResult**

No Test Case	Result	Status
1	Sistem tidak dapat menampilkan kamera untuk melakukan <i>scan</i>	<i>Valid</i>
2	Sistem tidak dapat memproses data <i>qr code</i> dan menampilkan “maaf data tidak dapat diproses”	<i>Valid</i>
3	Sistem dapat melakukan <i>scan</i> dan dapat memproses data sehingga sistem menampilkan “penampungan telah dikonfirmasi”	<i>Valid</i>
4	Sistem dapat memproses data tetapi API tidak dapat memproses data.	<i>Valid</i>

### 6.1.2 Pengujian Validasi

Pengujian Validasi adalah merupakan bagian dari pengujian *black box*. Pengujian validasi dilakukan untuk mengetahui bahwa fungsi – fungsi yang dibangun telah sesuai dengan kebutuhan fungsional atau belum. Kasus uji dibuat untuk masing- masing kebutuhan fungsional. Hasil yang didapatkan dari pengujian validasi akan dianalisis untuk mendapatkan kesimpulan apakah aplikasi telah memenuhi kebutuhan fungsional yang telah dideskripsikan atau belum.

#### 6.1.2.1 Kasus Uji

Kasus uji dibutuhkan dalam pengujian validasi karena dapat digunakan untuk memberi prosedur pengujian agar sesuai dengan kebutuhan fungsional. Kasus uji dibuat untuk masing-masing kebutuhan fungsional yang telah dideskripsikan. Berikut adalah detail kasus uji yang dapat dilihat pada tabel 6.1 sampai tabel 6.17.

##### a. Kasus Uji *Login*

**Tabel 6.7 Kasus Uji *Login* Berhasil**

Nama Kasus Uji	<i>Login</i> Berhasil
Tujuan Pengujian	Untuk membuktikan bahwa sistem dapat memasukkan pengguna ke sistem V-Apps dengan menggunakan

	<i>username</i> dan <i>password</i> yang terdaftar
Prosedur Pengujian	<ol style="list-style-type: none"> <li>1. Membuka Aplikasi V-Apps</li> <li>2. Memasukkan <i>username</i> dan <i>password</i> yang valid dan terdaftar</li> <li>3. Klik tombol <i>login</i></li> </ol>
Hasil yang Diharapkan	Sistem menampilkan halaman utama aplikasi

**Tabel 6.8 Kasus Uji *Login* Gagal**

Nama Kasus Uji	<i>Login</i> Gagal
Tujuan Pengujian	Sistem tidak memasukkan pengguna kedalam sistem V-apps yang dilakukan dengan menggunakan <i>username</i> dan <i>password</i> yang tidak terdaftar.
Prosedur Pengujian	<ol style="list-style-type: none"> <li>1. Membuka Aplikasi V-Apps</li> <li>2. Memasukkan <i>username</i> dan <i>password</i> yang tidak terdaftar</li> <li>3. Klik tombol <i>login</i></li> </ol>
Hasil yang Diharapkan	Sistem menampilkan notifikasi pesan kesalahan

b. Kasus Uji Melihat Jadwal Damping

**Tabel 6.9 Kasus Uji Melihat Jadwal Damping Berhasil**

Nama Kasus Uji	Melihat Jadwal Damping
Tujuan Pengujian	Untuk melihat jadwal damping pendamping dan status jadwal sehingga pendamping tau siapa yang akan didampingi
Prosedur Pengujian	<ol style="list-style-type: none"> <li>1. Menekan tombol <i>navigation bar</i></li> <li>2. Menekan tombol melihat jadwal</li> </ol>
Hasil yang Diharapkan	Sistem menampilkan jadwal damping milik pendamping serta status jadwal.



## c. Kasus Uji Menelpon Difabel

**Tabel 6.10 Kasus Uji Menelpon Difabel Berhasil**

Nama Kasus Uji	Menelpon Difabel
Tujuan Pengujian	Untuk membuka aplikasi telepon dan menampilkan no telpon mahasiswa difabel yang dapat digunakan untuk melakukan panggilan kepada mahasiswa difabel.
Prosedur Pengujian	1. Menekan gambar telepon pada salah satu jadwal
Hasil yang Diharapkan	Sistem menampilkan aplikasi telepon beserta no telepon mahasiswa difabel.

## d. Kasus Uji Mengirim Pesan Difabel

**Tabel 6.11 Kasus Uji Mengirim Pesan Difabel Berhasil**

Nama Kasus Uji	Mengirim Pesan Difabel
Tujuan Pengujian	Untuk membuka aplikasi pesan beserta no telepon mahasiswa difabel agar pendamping dapat mengirim pesan kepada mahasiswa difabel yang dituju.
Prosedur Pengujian	1. Menekan gambar pesan pada salah satu jadwal
Hasil yang Diharapkan	Sistem membuka aplikasi pengirim pesan beserta no telepon mahasiswa difabel

## e. Kasus Uji Melakukan Konfirmasi Kehadiran

**Tabel 6.12 Kasus Uji Melakukan Konfirmasi Kehadiran Berhasil**

Nama Kasus Uji	Melakukan Konfirmasi Kehadiran
Tujuan Pengujian	Untuk melakukan konfirmasi kehadiran dengan cara <i>Scan QR Code</i> milik mahasiswa difabel yang sudah dijadwalkan.
Prosedur Pengujian	<ol style="list-style-type: none"> <li>1. Menekan gambar <i>scan</i></li> <li>2. Mengarahkan kamera ke <i>qr code</i> yang dimiliki mahasiswa difabel</li> </ol>
Hasil yang Diharapkan	Sistem akan menampilkan notifikasi konfirmasi berhasil dan status jadwal berubah dari warna biru terjadwal menjadi warna kuning dikonfirmasi.

## f. Kasus Uji Melihat Profil Difabel

**Tabel 6.13 Kasus Uji Melihat Profil Difabel Berhasil**

Nama Kasus Uji	Melihat Profil Difabel
Tujuan Pengujian	Untuk melihat detail profil difabel pada jadwal yang sudah ditentukan
Prosedur Pengujian	<ol style="list-style-type: none"> <li>1. Klik salah satu jadwal yang tersedia</li> </ol>
Hasil yang Diharapkan	Sistem akan menampilkan data profil difabel yang akan di dampingi beserta jadwal.

## g. Kasus Uji Melakukan Filter Jadwal Damping

**Tabel 6.14 Kasus Uji Melakukan Filter Jadwal Damping Berhasil**

Nama Kasus Uji	Melakukan Filter Jadwal Damping
Tujuan Pengujian	Untuk melakukan filter melihat jadwal damping sesuai dengan tanggal yang dimasukkan
Prosedur Pengujian	<ol style="list-style-type: none"> <li>1. Klik tanggal di awal</li> <li>2. Masukkan tanggal awal untuk melihat jadwal</li> <li>3. Klik tanggal di akhir</li> </ol>

	4. Masukkan tanggal akhir untuk melihat jadwal
Hasil yang Diharapkan	Sistem akan menampilkan jadwal sesuai dengan tanggal yang sudah ditentukan.

h. Kasus Uji *Logout* Pendamping

**Tabel 6.15 Kasus Uji *Logout* Pendamping Berhasil**

Nama Kasus Uji	<i>Logout</i> Pendamping
Tujuan Pengujian	Untuk mengeluarkan pengguna dari sistem aplikasi V-Apps menuju halaman login
Prosedur Pengujian	1. Klik <i>Navigation Bar</i> 2. Klik <i>Logout</i>
Hasil yang Diharapkan	Sistem mengeluarkan pengguna dari sistem dan menampilkan halaman <i>login</i>

i. Kasus Uji Melihat Jadwal Pendampingan

**Tabel 6.16 Kasus Uji Melihat Jadwal Pendampingan Berhasil**

Nama Kasus Uji	Melihat Jadwal Pendampingan
Tujuan Pengujian	Menampilkan jadwal pendampingan serta status jadwal yang dimiliki mahasiswa difabel
Prosedur Pengujian	1. Menekan tombol <i>navigation bar</i> 2. Menekan tombol melihat jadwal
Hasil yang Diharapkan	Sistem menampilkan jadwal pendampingan milik mahasiswa difabel serta status jadwal.

## j. Kasus Uji Menelpon Pendamping

**Tabel 6.17 Kasus Uji Fungsi Menelpon Pendamping Berhasil**

Nama Kasus Uji	Menelpon Pendamping
Tujuan Pengujian	Membuka aplikasi telepon beserta no telepon pendamping sehingga mahasiswa difabel dapat menelepon pendamping yang dituju
Prosedur Pengujian	1. Menekan gambar telepon pada salah satu jadwal
Hasil yang Diharapkan	Sistem menampilkan aplikasi telepon beserta no telepon pendamping.

## k. Kasus Uji Mengirim Pesan Pendamping

**Tabel 6.18 Kasus Uji Mengirim Pesan Pendamping Berhasil**

Nama Kasus Uji	Mengirim Pesan Pendamping
Tujuan Pengujian	Membuka aplikasi pesan beserta no telepon pendamping sehingga mahasiswa difabel dapat mengirim pesan kepada pendamping yang dituju
Prosedur Pengujian	1. Menekan gambar pesan pada salah satu jadwal
Hasil yang Diharapkan	Sistem membuka aplikasi pesan beserta no telepon pendamping

## l. Kasus Uji Menampilkan QR Code

**Tabel 6.19 Kasus Uji Menampilkan QR Code Berhasil**

Nama Kasus Uji	Menampilkan QR Code
Tujuan Pengujian	Menampilkan qr code pendamping
Prosedur Pengujian	1. Menekan gambar scan pada salah satu jadwal
Hasil yang Diharapkan	Sistem menampilkan qr code

## m. Kasus Uji Memberi Review Pendamping

**Tabel 6.20 Kasus Uji Memberi *Review* Pendamping Berhasil**

Nama Kasus Uji	Memberi <i>Review</i> Pendamping
Tujuan Pengujian	Untuk memberikan <i>review</i> kepada pendamping yang sudah melakukan konfirmasi kehadiran dengan scan <i>QR Code</i>
Prosedur Pengujian	1. Memberi bintang 1-5 2. Klik tombol <i>submit</i>
Hasil yang Diharapkan	Sistem menampilkan notif terimakasih telah memberikan penilaian dan status jadwal berubah menjadi warna merah yaitu selesai

## n. Kasus Uji Melihat Profil Pendamping

**Tabel 6.21 Kasus Uji Melihat Profil Pendamping Berhasil**

Nama Kasus Uji	Melihat Profil Pendamping
Tujuan Pengujian	Untuk melihat detail profil pendamping pada jadwal yang sudah di tentukan
Prosedur Pengujian	1. Klik salah satu jadwal yang tersedia
Hasil yang Diharapkan	Sistem akan menampilkan data profil pendamping yang akan mendampingi beserta jadwal.

## o. Kasus Uji Melakukan Filter Jadwal Pendampingan

**Tabel 6.22 Kasus Uji Melakukan Filter Jadwal Pendampingan Berhasil**

Nama Kasus Uji	Melakukan Filter Jadwal Pendampingan
Tujuan Pengujian	Untuk melihat jadwal pendampingan dengan cara memfilter jadwal sesuai dengan tanggal yang di tentukan.
Prosedur Pengujian	1. Klik tanggal awal 2. Masukkan tanggal awal untuk melihat jadwal

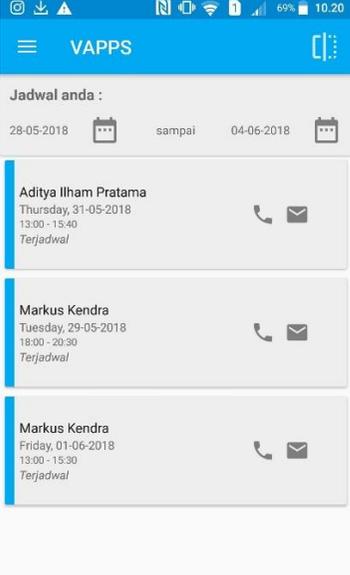
	<p>3. Klik tanggal akhir</p> <p>4. Masukkan tanggal akhir untuk melihat jadwal</p>
Hasil yang Diharapkan	Sistem akan menampilkan jadwal sesuai dengan tanggal yang sudah ditentukan.

p. Kasus Uji *Logout* Mahasiswa Difabel

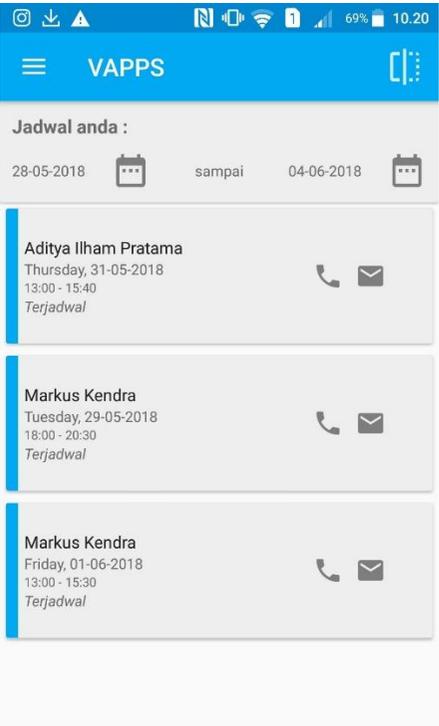
**Tabel 6.23 Kasus Uji *Logout* Pendamping Berhasil**

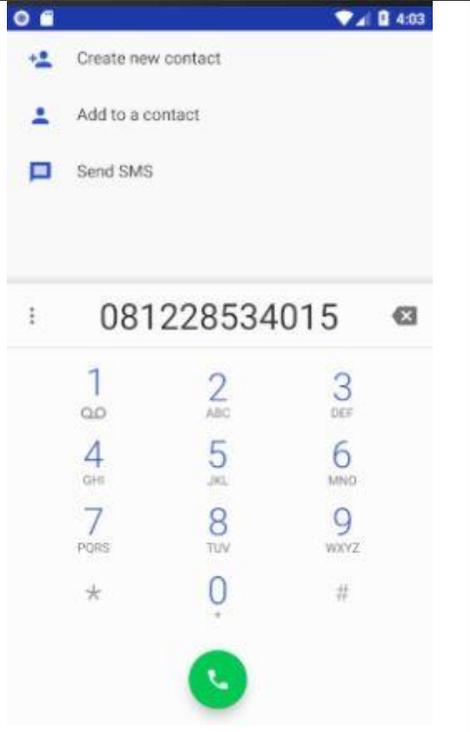
Nama Kasus Uji	Melakukan Fungsi <i>logout</i> Mahasiswa Difabel
Tujuan Pengujian	Untuk melakukan proses <i>logout</i> untuk keluar dari sistem sebagai Mahasiswa Difabel
Prosedur Pengujian	<ol style="list-style-type: none"> <li>1. Klik <i>Navigation Bar</i></li> <li>2. Klik <i>Logout</i></li> </ol>
Hasil yang Diharapkan	Sistem mengeluarkan pengguna dari sistem dan menampilkan halaman <i>login</i> .

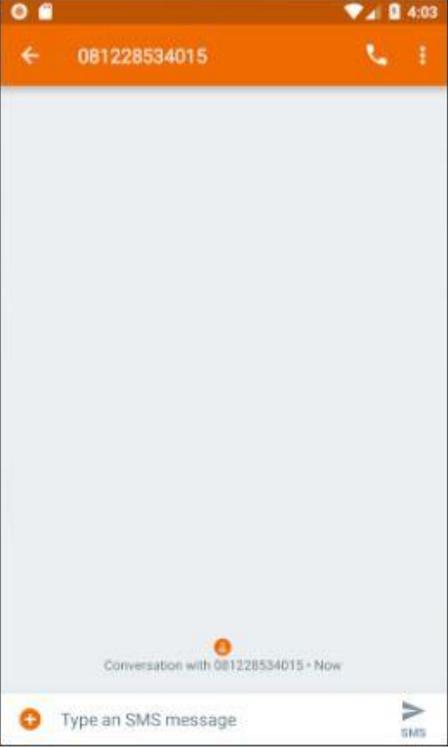
6.1.2.2 Hasil Pengujian Validasi

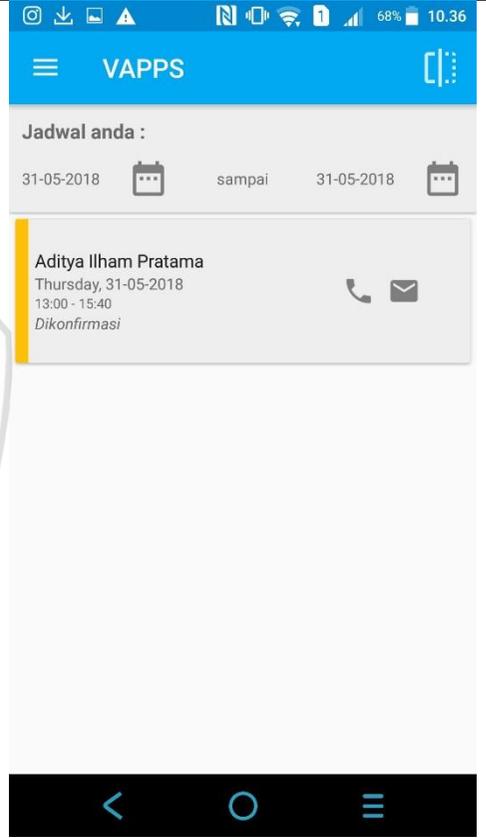
No	Kasu Uji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Hasil Uji	Screenshot Validasi
1	Login Berhasil	Sistem menampilkan halaman utama aplikasi yaitu menampilkan jadwal	Sistem menampilkan halaman utama aplikasi yaitu menampilkan jadwal	Valid	

2	Login Gagal	Sistem menampilkan notifikasi pesan kesalahan	Sistem menampilkan notifikasi pesan kesalahan	Valid	 <p>The screenshot shows the login interface of the 'Volunteer apps'. It features the app's logo at the top, followed by two input fields labeled 'Masukkan NIM' and 'Masukkan Password'. A blue 'LOGIN' button is positioned below the fields. At the bottom, a dark grey message box displays the error: 'Ada kesalahan, cek kembali kombinasi nim dan password anda'.</p>
---	-------------	---	---	-------	--

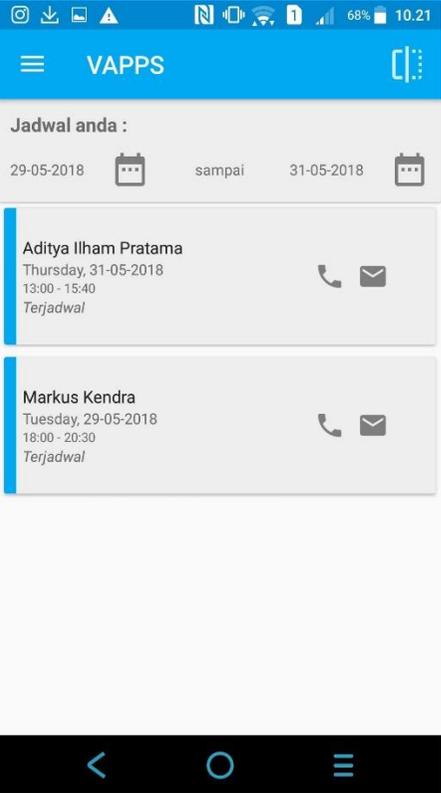
<p>3</p>	<p>Melihat Jadwal Damping Berhasil</p>	<p>Sistem menampilkan jadwal damping milik pendamping serta status jadwal.</p>	<p>Sistem menampilkan jadwal damping milik pendamping serta status jadwal.</p>	<p>Valid</p>	
----------	--	--	--	--------------	--

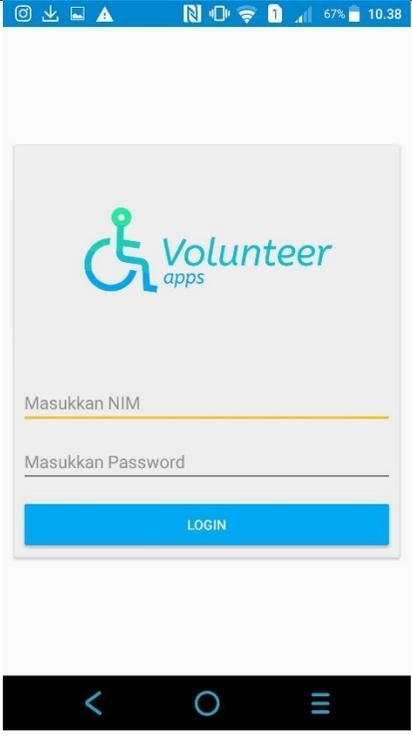
4	Menelpon Berhasil	Difabel	Sistem menampilkan aplikasi telepon beserta no telepon mahasiswa difabel.	Sistem menampilkan aplikasi telepon beserta no telepon mahasiswa difabel.	<i>Valid</i>	
---	-------------------	---------	---	---	--------------	--

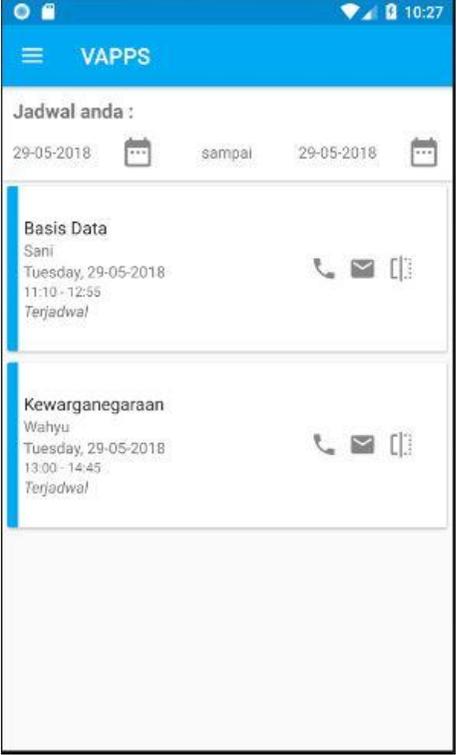
5	Mengirim Pesan Difabel Berhasil	Sistem membuka aplikasi pengirim pesan beserta no telepon mahasiswa difabel	Sistem membuka aplikasi pengirim pesan beserta no telepon mahasiswa difabel	Valid 
---	------------------------------------	---	---	---

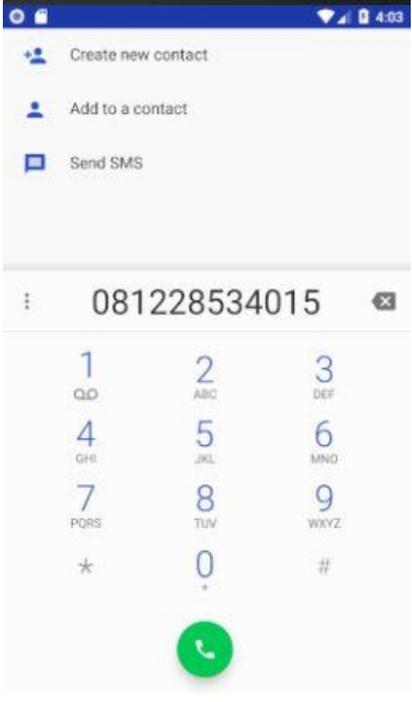
<p>6</p>	<p>Melakukan Konfirmasi Kehadiran Berhasil</p>	<p>Sistem akan menampilkan notifikasi konfirmasi berhasil dan status jadwal berubah dari warna biru terjadwal menjadi warna kuning dikonfirmasi.</p>	<p>Sistem akan menampilkan notifikasi konfirmasi berhasil dan status jadwal berubah dari warna biru terjadwal menjadi warna kuning dikonfirmasi.</p>	<p><i>Valid</i></p>	 <p>The screenshot shows the VAPPS mobile application interface. At the top, there is a blue header with the text 'VAPPS' and a menu icon on the left and a list icon on the right. Below the header, it says 'Jadwal anda :'. Underneath, there is a date range '31-05-2018' followed by a calendar icon, the word 'sampai', another calendar icon, and the date '31-05-2018'. A yellow bar highlights a specific entry for 'Aditya Ilham Pratama' on 'Thursday, 31-05-2018' from '13:00 - 15:40' with the status 'Dikonfirmasi'. To the right of this entry are icons for a phone call and an email. At the bottom of the screen, there is a black navigation bar with a back arrow, a home circle, and a menu icon.</p>
----------	--	--	--	---------------------	--

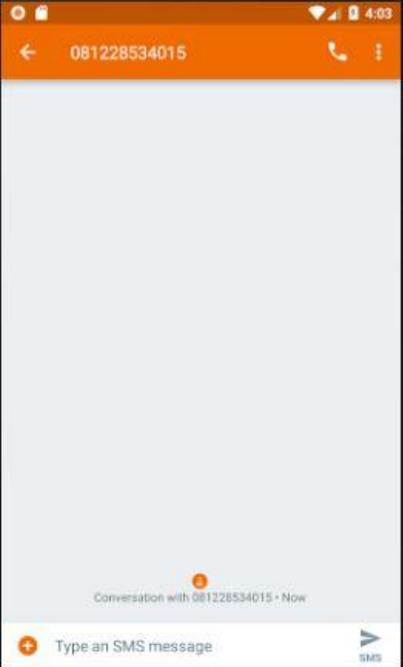
<p>7</p>	<p>Melihat Profil Difabel Berhasil</p>	<p>Sistem akan menampilkan data profil difabel yang akan di dampingi beserta jadwal.</p>	<p>Sistem akan menampilkan data profil difabel yang akan di dampingi beserta jadwal.</p>	<p><i>Valid</i></p> 
----------	--	--	--	--

<p>8</p>	<p>Melakukan Filter Damping Jadwal Berhasil</p>	<p>Sistem akan menampilkan jadwal sesuai dengan tanggal yang sudah ditentukan.</p>	<p>Sistem akan menampilkan jadwal sesuai dengan tanggal yang sudah ditentukan.</p>	<p><i>Valid</i></p>	
----------	---	--	--	---------------------	--

<p>9</p>	<p><i>Logout</i> Berhasil</p>	<p>Pendamping</p>	<p>Sistem mengeluarkan pengguna dari sistem dan menampilkan halaman <i>login</i></p>	<p>Sistem mengeluarkan pengguna dari sistem dan menampilkan halaman <i>login</i></p>	<p><i>Valid</i></p>	
----------	-----------------------------------	-------------------	--	--	---------------------	--

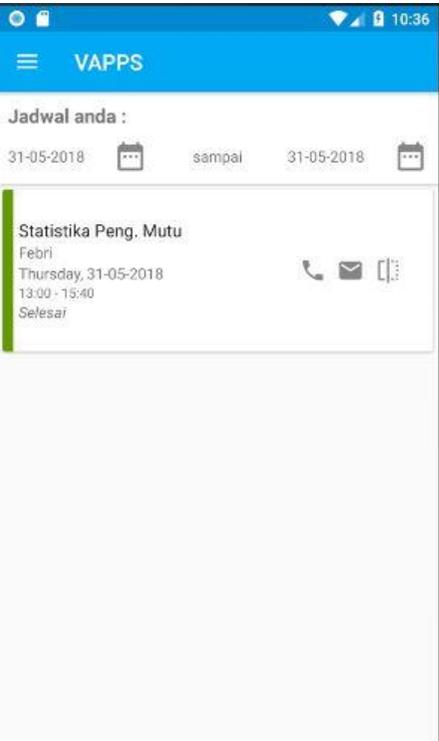
<p>19</p>	<p>Melihat Pendampingan Berhasil</p>	<p>Jadwal</p> <p>Sistem menampilkan jadwal pendampingan milik mahasiswa difabel serta status jadwal.</p>	<p>Sistem menampilkan jadwal pendampingan milik mahasiswa difabel serta status jadwal.</p>	<p>Valid</p>	
-----------	--------------------------------------	--	--	--------------	--

<p>20</p>	<p>Menelpon Pendamping Berhasil</p>	<p>Sistem menampilkan aplikasi telepon beserta no telepon pendamping.</p>	<p>Sistem menampilkan aplikasi telepon beserta no telepon pendamping.</p>	<p><i>Valid</i></p>	
-----------	-------------------------------------	---	---	---------------------	--

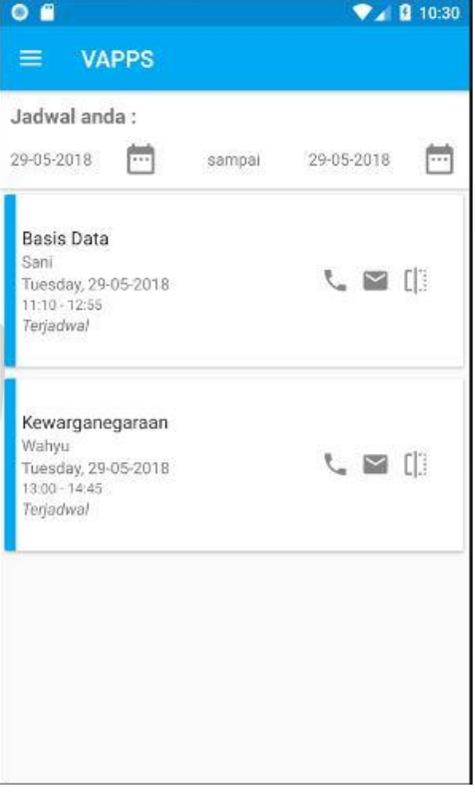
21	Mengirim Pesan Pendamping Berhasil	Sistem membuka aplikasi pesan beserta no telepon pendamping	Sistem membuka aplikasi pesan beserta no telepon pendamping	<i>Valid</i>	
----	------------------------------------	---	---	--------------	--

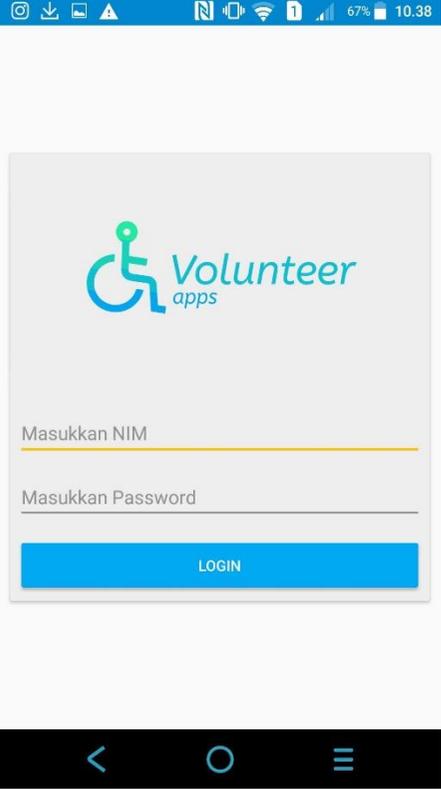


<p>22</p>	<p>Menampilkan QR Code Berhasil</p>	<p>Sistem menampilkan qr code</p>	<p>Sistem menampilkan qr code</p>	<p>Valid</p>	
-----------	-------------------------------------	-----------------------------------	-----------------------------------	--------------	--

<p>23</p>	<p>Memberi Pendamping Berhasil <i>Reveiw</i></p>	<p>Sistem menampilkan notif terimakasih telah memberikan penilaian dan status jadwal berubah menjadi warna merah yaitu selesai</p>	<p>Sistem menampilkan notif terimakasih telah memberikan penilaian dan status jadwal berubah menjadi warna merah yaitu selesai</p>	<p><i>Valid</i></p>	
-----------	--	--	--	---------------------	--

<p>24</p>	<p>Melihat Profil Pendamping Berhasil</p>	<p>Sistem akan menampilkan data profil pendamping yang akan mendampingi beserta jadwal.</p>	<p>Sistem akan menampilkan data profil pendamping yang akan mendampingi beserta jadwal.</p>	<p><i>Valid</i></p>	
-----------	---	---	---	---------------------	--

<p>25</p>	<p>Melakukan Filter Jadwal Pendampingan Berhasil</p>	<p>Sistem akan menampilkan jadwal sesuai dengan tanggal yang sudah ditentukan</p>	<p>Sistem akan menampilkan jadwal sesuai dengan tanggal yang sudah ditentukan.</p>	<p><i>Valid</i></p>	
-----------	--	---	--	---------------------	--

<p>26</p>	<p>Logout Difabel</p>	<p>Mahasiswa</p>	<p>Sistem mengeluarkan pengguna dari sistem dan menampilkan halaman <i>login</i></p>	<p>Sistem mengeluarkan pengguna dari sistem dan menampilkan halaman <i>login</i></p>	<p><i>Valid</i></p>	
-----------	---------------------------	------------------	--	--	---------------------	--

## 6.2 Pengujian Non Fungsional

### 6.2.1 Pengujian Usabilitas

Pengujian usabilitas pada penelitian ini digunakan untuk melihat tingkat kepuasan pengguna. Metode yang digunakan untuk pembuatan kuesioner adalah metode SUS atau *system usability scale*. Pengujian ini terbukti kehandalannya dalam mendeteksi perbedaan pada ukuran sampel yang kecil kuisioner dan tersedia secara komersil atau bebas digunakan. SUS dapat digunakan pada sampel yang berukuran sangat kecil namun tetap menghasilkan hasil yang dapat diandalkan (Jeff Sauro, 2011).

Metode SUS terdiri dari 10 butir pertanyaan yang terdiri dari 5 pernyataan positif dan 5 pertanyaan negatif. Kuesioner dibagikan kepada responden pendamping dan mahasiswa difabel yang terdaftar di PSLD UB. Untuk pendamping akan menguji aplikasi yang *login* sebagai pendamping dan mahasiswa difabel akan menguji aplikasi yang *login* sebagai mahasiswa difabel.

Daftar pertanyaan dari kuesioner yang diberikan kepada responden sebagai sampel yang akan diuji dapat dilihat pada tabel 6.18.

**Tabel 6.24 Kuesioner Pengujian Usabilitas**

No	Pertanyaan	Sangat Tidak Setuju					Sangat Setuju					
		1	2	3	4	5	1	2	3	4	5	
1	Saya pikir saya akan sering menggunakan aplikasi ini											
2	Saya pikir ada beberapa fitur dalam aplikasi yang membuat saya kerepotan											
3	Menurut saya aplikasi ini mudah digunakan											
4	Saya berfikir bahwa saya membutuhkan bantuan untuk menggunakan aplikasi ini											
5	Menurut saya fitur yang ada dalam aplikasi sudah terintegrasi dengan baik											
6	Menurut saya, terdapat banyak fungsi yang berjalan secara tidak konsisten pada aplikasi ini											

7	Menurut saya, banyak orang yang akan dengan mudah menggunakan aplikasi ini					
8	Menurut saya aplikasi ini sangat sulit untuk digunakan					
9	Saya merasa nyaman saat menggunakan aplikasi ini					
10	Saya harus belajar banyak hal sebelum saya dapat menggunakan aplikasi ini					

Pertanyaan positif ditunjukkan pada pernyataan yang terdapat pada nomor ganjil, sementara pertanyaan negatif terdapat pada nomor genap. Kuesioner akan dibagikan kepada 5 pendamping dan 5 mahasiswa difabel.

#### 6.2.1.1 Perhitungan Skor Kuesioner

Setiap pernyataan diisi dengan salah satu skala 1 sampai 5 (sangat tidak setuju sampai sangat setuju), kemudian dihitung skor dari setiap pertanyaan, dan skor dijumlahkan untuk mendapatkan total skor dari satu kuesioner. Perhitungan ini akan dilakukan pada masing – masing kuesioner yang telah dibagikan. Detail perhitungan yang dilakukan berdasarkan rumusan dibawah ini:

- Setiap pertanyaan dengan nomor ganjil, skala yang dipilih dikurangi 1 (skala – 1)
- Setiap pertanyaan dengan nomor genap, 5 dikurangi skala yang dipilih (5 – skala)
- Kemudian hasil dari perhitungan skor pada setiap nomor ganjil dan genap tersebut, seluruh skor dijumlahkan dan dikalikan 2,5 agar didapatkan nilai dari 0-100.

Standard kepuasan yang baik pada penilaian SUS ini adalah pada skor 68 (Sauro, 2011). Jika skor yang didapatkan kurang dari 68 maka aplikasi tidak berjalan dengan baik atau mengalami masalah dalam penggunaannya. Jika skor lebih dari 68, maka aplikasi berjalan dengan baik dalam penggunaannya atau setiap fungsi tidak terdapat kesalahan dan berjalan secara konsisten. Dapat disimpulkan bahwa jika skor kurang dari 68 pengguna tidak menyukai aplikasi dan jika skor lebih dari 68 pengguna menyukai aplikasi yang ditawarkan.

#### 6.2.1.2 Hasil Pengujian Usabilitas

Pengujian dilakukan di kantor Pusat Studi dan Layanan Disabilitas Universitas Brawijaya dengan setiap responden diberikan kuesioner dan harus menjawab setiap pertanyaan. Pertanyaan yang diberikan terdiri dari

pertanyaan positif pada nomor ganjil dan negatif pada nomor genap. Pada tabel 6. merupakan sampel dari pengisian kuesioner yang dilakukan oleh Responden 1 pada pengguna Mahasiswa Difabel.

**Tabel 6.25 Sampel Kuesioner Pengujian Usabilitas**

No	Pertanyaan	Penilaian oleh Pengguna	Perhitungan	Hasil
1	Saya pikir saya akan sering menggunakan aplikasi ini	4	4-1	3
2	Saya pikir ada beberapa fitur dalam aplikasi yang membuat saya kerepotan	2	5-2	3
3	Menurut saya aplikasi ini mudah digunakan	5	5-1	4
4	Saya berfikir bahwa saya membutuhkan bantuan untuk menggunakan aplikasi ini	1	5-1	4
5	Menurut saya fitur yang ada dalam aplikasi sudah terintegrasi dengan baik	4	4-1	3
6	Menurut saya, terdapat banyak fungsi yang berjalan secara tidak konsisten pada aplikasi ini	1	5-1	4
7	Menurut saya, banyak orang yang akan dengan mudah menggunakan aplikasi ini	4	4-1	3
8	Menurut saya aplikasi ini sangat sulit untuk digunakan	2	5-2	3
9	Saya merasa nyaman saat menggunakan aplikasi ini	3	3-1	2
10	Saya harus belajar banyak hal sebelum saya dapat menggunakan aplikasi ini	1	5-1	4
	<b>TOTAL</b>		<b>Jumlah* 2,5</b>	<b>82,5</b>

Untuk seluruh data pengisian kuesioner oleh pengguna dapat dilihat pada lampiran dokumen ini. Kemudian Tabel 6.20 dan 6.21 merupakan hasil perhitungan kuesioner yang dibagikan kepada 10 responden yang terdiri dari 5 pendamping dan 5 mahasiswa difabel.

## Pendamping

Tabel 6.26 Hasil Responden Pendamping

No	Nama Responden	Total Skor
1	Responden 1	97,5
2	Responden 2	85
3	Responden 3	87,5
4	Responden 4	82,5
5	Responden 5	80
<b>Rata – Rata Total Skor</b>		<b>86,5</b>

## Mahasiswa Difabel

Tabel 6.27 Hasil Responden Mahasiswa Difabel

No	Nama Responden	Total Skor
1	Responden 1	82,5
2	Responden 2	80
3	Responden 3	87,5
4	Responden 4	85
5	Responden 5	87,5
<b>Rata – Rata Total Skor</b>		<b>84,5</b>

## 6.3 Analisis Hasil Pengujian

Setelah semua pengujian telah selesai dilakukan, hasil dari semua pengujian dianalisis untuk dapat menarik kesimpulan bahwa hasil dari semua pengujian yang telah selesai dilakukan telah memenuhi kebutuhan yang telah dijelaskan pada analisis kebutuhan.

### 6.3.1 Pengujian Unit

Dari semua pengujian unit yang dilakukan yaitu diantaranya `loadVolunteerSchedule()`, `loadMahasiswaSchedule()`, `onActivityResult()`, dihasilkan beberapa *basis path* atau jalur dasar yaitu semua kemungkinan yang dapat terjadi tersebut sudah diuji dan mendapatkan hasil yang valid pada setiap pengujianya sehingga semua pengujian sudah sesuai dengan kebutuhan yang telah ditentukan pada analisis kebutuhan.

### 6.3.2 Pengujian Validasi

Dalam pengujian validasi semua fungsi telah diuji berdasarkan pengujian yang dilakukan dengan mencoba aplikasi yang dijalankan dan menjalankan aplikasi sesuai dengan prosedur pengujian yang sudah ditentukan, dari semua pengujian validasi tidak terdapat fungsi yang tidak valid yang berarti bahwa semua fungsi telah sesuai dengan kebutuhan yang telah ditentukan pada analisis kebutuhan.

### 6.3.3 Pengujian Usabilitas

Berdasarkan hasil pengujian pada tabel 6.20 dan 6.21 , dapat disimpulkan bahwa seluruh responden yang menggunakan aplikasi V-Apps ini dapat menjalankan aplikasi dengan mudah dalam menjalankan setiap fungsi pada aplikasi dan merasa senang dalam menggunakannya. Sehingga rata – rata skor yang didapatkan pada responden pendamping adalah 86,5 dan responden mahasiswa difabel adalah 84,5.



## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berikut adalah. Berdasarkan hasil analisis kebutuhan, perancangan, implementasi, dan pengujian yang telah dilakukan, maka kesimpulan dalam penelitian pengembangan aplikasi V-Apps sebagai aplikasi untuk menginformasikan jadwal damping mahasiswa difabel disimpulkan sebagai berikut:

1. Pada hasil analisis kebutuhan didapatkan tiga aktor yang dapat menggunakan aplikasi V-Apps. Aktor pertama yaitu pengguna umum memiliki satu kebutuhan fungsional, aktor kedua yaitu pendamping memiliki tujuh kebutuhan fungsional, dan aktor ketiga yaitu mahasiswa difabel yang memiliki delapan kebutuhan fungsional. Serta memiliki satu kebutuhan non fungsional. Kebutuhan fungsional tersebut juga dimodelkan dalam *use case diagram*, serta *use case scenario*.
2. Pada hasil perancangan didapatkan dari hasil analisis kebutuhan. Rancangan yang dibuat yaitu perancangan arsitektur, perancangan data, perancangan komponen, perancangan antarmuka. Dalam perancangan arsitektur sistem terdapat *class diagram* yang dijelaskan dengan rinci pada setiap operasi dan atributnya dan dijelaskan lima *sequence diagram*. Pada perancangan data diperoleh rancangan *database* yang didapatkan dari sistem penjadwalan kantor PSLD UB yang dibuat berupa *entity relationship diagram* dan API yang digunakan pada aplikasi V-Apps. Pada perancangan komponen terdapat tiga *algoritme* yang akan digunakan dan diimplementasikan pada sistem. Pada perancangan antarmuka berisi *mock-up* atau gambaran dari sistem yang akan dibangun. Tahap implementasi yang dilakukan menghasilkan implementasi basis data, implementasi kode program, dan implementasi antarmuka. Implementasi basis data didapatkan *database* yang disimpan dalam *MySql* dan juga didapatkan hasil implementasi dari API yang digunakan pada aplikasi V-Apps. Implementasi kode program didapatkan tiga kode program yang didapat dari hasil rancangan komponen. Sedangkan implementasi antarmuka didapatkan dari hasil perancangan antarmuka. Pada tahap pengujian dihasilkan dua pengujian yaitu pengujian unit dan pengujian validasi. Pengujian unit dilakukan dengan *white box testing* dan pengujian validasi dilakukan dengan *black box testing*.
3. Sedangkan untuk menentukan kemudahan sistem V-App menggunakan pengujian usabilitas. Pengujian usabilitas berguna untuk menilai kemudahan penggunaan aplikasi V-Apps untuk menginformasikan jadwal damping mahasiswa difabel.



## 7.2 Saran

Saran yang diberikan untuk pengembangan aplikasi V-Apps sebagai aplikasi untuk menginformasikan jadwal damping mahasiswa difabel ini selanjutnya antara lain:

1. Pendamping dapat melakukan izin lewat aplikasi. Hal ini masih belum bisa dilakukan karena proses pelaksanaan izin yang masih belum jelas dan perlu pengaturan jadwal ulang apabila ada izin. Dan aplikasi perlu menampilkan notif apabila ada izin dari setiap pendamping.
2. Untuk sekarang tidak bisa menampilkan data foto setiap pendamping dan mahasiswa difabel karena data masih belum tersedia. Oleh karena itu, diharapkan kedepan dapat menampilkan foto.
3. Sistem dapat menampilkan rating pendamping yang diperoleh.



## DAFTAR PUSTAKA

Amman, P. & Offutt, J., 2008. *Introduction to Software Testing*. New York: Cabridge University Press New York.

Android, 2018. *Content License*. [Online]  
Available at: <https://source.android.com/setup/start/licenses>  
[Diakses 20 April 2018].

Android, 23 September 2008. *Announcing The Android 1.0 SDK, release 1*. [Online]  
Available at: <https://android-developers.googleblog.com/2008/09/announcing-android-10-sdk-release-1.html>  
[Diakses 30 April 2018].

Disabilitas, P. S. d. L., 2018. *Kemahasiswaan*. [Online]  
Available at: <http://psld.ub.ac.id/in/kemahasiswaan/daftar-mahasiswa/>  
[Diakses 2 Januari 2018].

Heriyanto, A. P., 2016. *Mobile's phone Forencsics Theory*. [Online]  
Available at: [https://books.google.co.id/books?id=erg5DgAAQBAJ&pg=PA104&dq=json+adalah&hl=id&sa=X&ved=0ahUKEwjfg\\_DGi8TSAhWMvI8KHYYAnAGYQ6AEILzAE#v=onepage&q=json%20adalah&f=false](https://books.google.co.id/books?id=erg5DgAAQBAJ&pg=PA104&dq=json+adalah&hl=id&sa=X&ved=0ahUKEwjfg_DGi8TSAhWMvI8KHYYAnAGYQ6AEILzAE#v=onepage&q=json%20adalah&f=false)  
[Diakses 27 Maret 2018].

Json, 2017. *Introducing Json*. [Online]  
Available at: <http://www.json.org/>  
[Diakses 28 Maret 2018].

Kolawa, A., 2007. *Automated Defect Prevention : Best Practices in Software Management*. s.l.:Wiley-IEEE Computer Society Press.

Lintangsari, A. P., Pratiwi, A., Fatmala, U. R. & Gustomy, R., 2016. *Buku Panduan Aksebilitas Layanan*. 2nd penyunt. Malang: Pusat Studi dan Layanan Disabilitas Universitas Brawijaya.

L, W., 2008. *The Certified Software Quality Engineer Handbook*. s.l.:ASQ Quality Press.

Pressman, R., 2010. *Software Engineering*. 7nd penyunt. New York: McHraw-Hill.

Rouf, A., 2012. *Pengujian Perangkat Lunak dengan Menggunakan Metode White Box dan Black Box*. Semarang: STIMIK Himsya Semarang.

Sauro, J., 2011. *Measuring Usability With The System Usability Scale (SUS)*. [Online]  
Available at: <https://measuringu.com/sus/>  
[Diakses 25 Mei 2018].

Soeleman, 2017. *QR Code dengan ZXing.Net*. [Online] Available at: <https://www.codepolitan.com/qr-code-dengan-zxingnet-58a9bb66912ae> [Diakses 1 Juni 2018].

Sommerville, I., 2010. *Software Engineering*. 9th ed. New York: Addison-Wesley.

Sutrisno, N. A., 2017. *REST API Client Sederhana dengan Retrofit pada Android Studio*. [Online] Available at: <https://www.codepolitan.com/rest-api-client-sederhana-dengan-retrofit-pada-android-studio-58986d62c46ae> [Diakses 2 Juni 2018].

Thomas, N., 2015. *How to Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website*. [Online] Available at: <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/> [Diakses 25 Mei 2018].

Tutorialpoint, 2017. *SDLC - Overview*. [Online] Available at: [https://www.tutorialspoint.com/sdlc/sdlc\\_overview.htm](https://www.tutorialspoint.com/sdlc/sdlc_overview.htm) [Diakses 30 April 2018].

Whitten, J. L. & Bentley, L. D., 2007. *System Analysis and Design Methods*. 7th penyunt. New York: McGraw-Hill.

