



PENGEMBANGAN APLIKASI PELATIHAN PENDAMPING PUSAT STUDI DAN LAYANAN DISABILITAS UNIVERSITAS BRAWIJAYA (PSLD UB) BERBASIS ANDROID

SKRIPSI

Disusun oleh:

Beni Dektos Heronimus

NIM: 165150201111056



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

2020



PENGESAHAN

PENGEMBANGAN APLIKASI PELATIHAN PENDAMPING PUSAT STUDI DAN
LAYANAN DISABILITAS UNIVERSITAS BRAWIJAYA (PSLD UB) BERBASIS ANDROID

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Beni Dektos Heronimus
NIM: 165150201111056

Skripsi ini telah diuji dan dinyatakan lulus pada
24 Juni 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

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

Dosen Pembimbing II

Dioko Pramono, S.T., M.Kom.
NIP: 19780108 200501 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Irfi Astoko Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001



PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan dengan sebenar benarnya dari pengetahuan yang saya miliki, pada naskah skripsi yang saya buat ini tidak mengandung karya ilmiah suatu perguruan tinggi lain yang pernah dilakukan pengajuan oleh seseorang guna memperoleh gelar akademik, dan tidak mengandung suatu karya atau sebuah pendapat yang ditulis atau diterbitkan oleh orang lain, melainkan dengan tertulis telah disitasi pada naskah ini serta disebutkan sebagai daftar referensi.

Apabila dikemudian hari di dalam naskah skripsi ini mengandung unsur-unsur plagiasi yang telah atau dapat dibuktikan, saya telah bersedia skripsi ini digugurkan serta gelar akademik yang diperoleh (sarjana) dibatalkan, dan dilakukan tindakan sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 06 Mei 2020



Beni Dektos Heronimus

NIM: 165150201111056



PRAKATA

Puja dan puji syukur atas kehadiran Allah Subhanahuwataala tuhan yang maha esa, yang telah melimpahkan segala karunia dan kebesarannya sehingga laporan skripsi dengan judul “Pengembangan Aplikasi Pelatihan Pendamping Pusat Studi dan Layanan Disabilitas Univeristas Brawijaya (PSLD UB) berbasis android” telah berhasil diselesaikan. Dengan segala keterbatasan Penulis menyadari bahwasanya terlaksananya penelitian serta laporan ini tidak selesai apabila tanpa bantuan oleh berbagai pihak-pihak terkait. Oleh karena ucapan terimakasih dan rasa hormat penulis sampaikan kepada:

1. Bapak Fajar Pradana, S.ST., M.Eng. serta Bapak Djoko Pramono, S.T.,M.Kom. karena telah dengan sabar dalam membimbing dan mengarahkan penulis dalam menyelesaikan skripsi,
2. Bapak Agus Widodo, S.T, M.T, Ph.D. sebagai Ketua Program Studi Teknik Informatika,
3. Bapak Tri Astoto Kurniawab, S.T, M.T, Ph.D. sebagai Ketua Jurusan Teknik Informatika,
4. Dosen Fakultas Ilmu Komputer yang telah mencurahkan segala ilmunya dalam rangka proses pembelajaran di fakultas ilmu komputer,
5. Ibu Lintang selaku koordinator bidang pelayanan *Disability Awareness* PSLD UB,yang telah memberikan izin dalam proses wawancara dan penelitian.
6. Orang Tua dan juga Seluruh saudara dari penulis yang selalu memberikan dukungan materil dan moril sehingga mendukung jalannya penelitian dengan baik.
7. Serta berbagai Pihak yang membantu dalam kelancaran jalannya penelitian.

Penulis dengan sepenuh hati menyadari dalam penyusunan skripsi ini terdapat banyak kekurangan, oleh karena itu berbagai saran dan kritik yang memiliki sifat membangun sangatlah penulis harapkan. Demikian pada akhir kata penulis memiliki harapan penyusunan serta penulisan dalam skripsi ini dapat bermanfaat bagi berbagai pihak yang telah membaca.

Malang, 06 Mei 2020

Beni Dektos Heronimus
benidh@student.ub.ac.id

ABSTRAK

Beni Dektos Heronimus, Pengembangan Aplikasi Pelatihan Pendamping Pusat Studi dan Layanan Disabilitas Universitas Brawijaya (PSLD UB) Berbasis Android
Pembimbing : Fajar Pradana, S.ST., M.Eng. dan Djoko Pramono S.T., M.Kom.

Pusat Studi dan Layanan Disabilitas atau yang biasa disingkat dengan PSLD UB, merupakan salah satu lembaga di Universitas Brawijaya yang secara umum memiliki fungsi yaitu menumbuhkan kesadaran tentang pentingnya kesetaraan hak antara orang-orang atau mahasiswa yang berkebutuhan khusus atau disabilitas dengan orang-orang biasa, memberikan pendampingan kepada mahasiswa atau mahasiswi yang memiliki kebutuhan khusus atas kesetaraan hak menerima pendidikan bagi setiap orang. Dalam melakukan pendampingan PSLD UB merekrut mahasiswa atau mahasiswi Universitas Brawijaya sebagai bentuk keterlibatan mahasiswa dalam membantu teman-temannya yang memiliki kebutuhan khusus. Setiap pendamping menerima pelatihan-pelatihan tentang bagaimana melakukan pendampingan terhadap mahasiswa dengan kebutuhan khusus dalam proses belajar dan proses kegiatan akademik di lingkungan universitas. Pada tahap setelah pendaftaran seluruh calon pendamping baru mendapatkan pelatihan mengenai kesadaran menumbuhkan kepedulian terhadap disabilitas dan tatacara melakukan pendampingan selama dua hari. Dimana para calon pendamping baru diberikan materi dan praktek mengenai jenis-jenis disabilitas yang ada di Universitas Brawijaya contohnya seperti pelatihan Bahasa isyarat. Tetapi dalam praktiknya para calon pendamping baru kurang bisa menyerap materi dengan baik dikarenakan waktu pelatihan yang cenderung singkat. Oleh karena itu diperlukan suatu sistem aplikasi pelatihan pendamping PSLD UB berbasis Android yang berfungsi sebagai media pelatihan lanjutan untuk calon pendamping dalam menyerap materi pelatihan pendamping. pada analisis kebutuhan didapat 23 kebutuhan fungsional dan 1 kebutuhan non-fungsional yang dibutuhkan aplikasi dan kemudian di implementasikan dengan Bahasa pemrograman java dengan platform android serta menggunakan metode pengembangan software yaitu *Waterfall* yang prosesnya terdiri dari rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pemeliharaan (*maintenance*) dan kemudian dilakukan pengujian validasi yang menghasilkan tingkat valid 100 % serta dilakukan pengujian *usability* berdasarkan dengan jawaban responden menggunakan skala *likert* menghasilkan nilai 75,8 sehingga masuk kategori *acceptable*.

Kata kunci : Pendamping, *Waterfall*, Android, PSLD UB



ABSTRACT

Beni Dektos Heronimus, Development of Companion Training Application for Study Centers and Disability Services of Universitas Brawijaya (PSLD UB) Based on Android

Advisor: Fajar Pradana, S.ST., M.Eng. and Djoko Pramono S.T., M.Kom.

The Center for Disability Study and Services or commonly abbreviated as PSLD UB, is one of the institutions in UB that generally has the function of raising awareness about the importance of equal rights between people or students with special needs or disabilities and ordinary people, providing assistance to students who have special needs for equal rights to receive education for everyone. In assisting PSLD UB, recruiting students of Universitas Brawijaya as a form of student involvement in helping their friends who have special needs. Each mentor receives training on how to provide assistance to students with special needs in the learning process and the process of academic activities within the university. At the stage after registration, all prospective mentors received training on awareness to raise awareness of disability and how to conduct mentoring for two days. Where new prospective companions are given material and practice on the types of disabilities that exist in UB such as sign language training. But in practice the new potential candidates are not able to absorb the material well because the training time tends to be short. Therefore we need an Android-based PSLD UB companion training application system that functions as a medium for advanced training for potential trainees in absorbing companion training materials. in the needs analysis there are 23 functional needs and 1 non-functional requirements needed by the application and then it is implemented with the Java programming language with the android platform and using a software development method that is Waterfall which consists of plans (planning), analysis (analysis), design (design), implementation (implementation), testing (testing) and maintenance (maintenance) and then conducted a validation test that produces a validation rate of 100% and usability testing is done based on the answers of respondents using a Likert scale producing a value of 75.8 so that it falls into the acceptable category.

Keywords: Companion, Waterfall, Android, PSLD UB.



DAFTAR ISI

PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS	ii
PRAKATA	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 PSDL UB	5
2.2 Kajian Pustaka	6
2.3 Rekayasa Perangkat Lunak	7
2.4 Pengembangan Perangkat Lunak	7
2.4.1 Model <i>Waterfall</i>	7
2.4.2 <i>Object Oriented Programming</i> (OOP)	9
2.4.3 Unified Modeling Language (UML)	11
2.5 Teknologi Pengembangan Sistem	17
2.5.1 Android	17
2.5.2 Firebase Database	18
2.5.3 Android Studio	19
2.5.4 JAVA	19
BAB 3 METODOLOGI	21



3.1 Analisis Kebutuhan	21
3.2 Perancangan Sistem	22
3.3 Implementasi Sistem	22
3.4 Pengujian	22
3.4.1 <i>White Box Testing</i>	22
3.4.2 <i>Black Box Testing</i>	23
3.4.3 Pengujian <i>Usability</i>	23
3.5 Kesimpulan dan Saran	23
BAB 4 analisis kebutuhan	24
4.1 Deskripsi Sistem	24
4.2 Elisitasi Kebutuhan	25
4.3 Identifikasi Aktor	25
4.4 Kebutuhan Fungsional	26
4.5 Kebutuhan Non Fungsional	29
4.6 Peemodelan Kebutuhan	29
4.6.1 <i>Use Case Diagram</i>	29
4.6.2 <i>Use Case Scenario</i>	31
BAB 5 PERANCANGAN DAN IMPLEMENTASI	42
5.1 Perancangan Sistem	42
5.1.1 Perancangan <i>Sequence Diagram</i>	42
5.1.2 Perancangan <i>Class Diagram</i>	44
5.1.3 Perancangan Basis Data	46
5.1.4 Perancangan Komponen	47
5.1.5 Perancangan Antarmuka	50
BAB 6 Implementasi	55
6.1 Implementasi Sistem	55
6.1.1 Spesifikasi Perangkat Keras	55
6.1.2 Spesifikasi Perangkat Lunak	55
6.2 Implementasi Basis Data	56
6.2.1 Tabel <i>User</i>	56
6.2.2 Tabel Jadwal	56
6.2.3 Tabel Materi	57



6.2.4 Tabel Materi	57
6.2.5 Tabel Soal	58
6.2.6 Tabel Video	58
6.2.7 Tabel Timer	59
6.3 Implementasi Kode Program	59
6.3.1 Implementasi Kode Program Register	59
6.3.2 Implementasi Kode Program Tambah Pengalaman	61
6.3.3 Implementasi Kode Program Lihat Video	62
6.4 Implementasi Antarmuka	65
6.4.1 Implementasi Antarmuka <i>Register</i>	65
6.4.2 Implementasi Antarmuka <i>Login</i>	65
6.4.3 Implementasi Antarmuka <i>Dashboard</i>	66
6.4.4 Implementasi Antarmuka <i>Pengalaman</i>	67
6.4.5 Implementasi Antarmuka <i>Video</i>	67
6.4.6 Implementasi Antarmuka soal	68
6.4.7 Implementasi Antarmuka materi	69
BAB 7 PENGUJIAN	70
7.1 Pengujian Unit	70
7.1.1 Pengujian Unit tambah soal	70
7.1.2 Pengujian Unit Edit Profil	72
7.1.3 Pengujian Unit tambah pengalaman	76
7.2 Pengujian Integrasi	80
7.3 Pengujian Validasi	83
7.3.1 Pengujian Validasi <i>Login</i>	83
7.3.2 Pengujian Validasi <i>Register</i>	84
7.3.3 Pengujian Validasi lihat materi	85
7.3.4 Pengujian Validasi lihat video	86
7.3.5 Pengujian Validasi lihat pengalaman	86
7.3.6 Pengujian Validasi komentar	86
7.3.7 Pengujian Validasi kerjakan soal	87
7.3.8 Pengujian Validasi lupa <i>password</i>	87
7.3.9 Pengujian Validasi lihat jadwal pelatihan	87



7.3.10 Pengujian Validasi tambah jadwal.....	88
7.3.11 Pengujian Validasi lihat <i>user</i>	88
7.3.12 Pengujian Validasi hapus <i>user</i>	88
7.3.13 Pengujian Validasi hapus soal.....	89
7.3.14 Pengujian Validasi tambah materi.....	89
7.3.15 Pengujian Validasi hapus materi.....	90
7.3.16 Pengujian Validasi verifikasi pengalaman.....	90
7.3.17 Pengujian Validasi hapus jadwal.....	91
7.3.18 Pengujian Validasi kerjakan <i>quiz</i>	91
7.4 Pengujian <i>Usability</i>	91
7.4.1 Hasil Pengujian <i>Usability</i>	92
BAB 8 Penutup.....	96
8.1 Kesimpulan.....	96
8.2 Saran.....	96
daftar referensi.....	97



DAFTAR TABEL

Tabel 2.1 Simbol <i>use case</i> diagram	11
Tabel 2.2 contoh Class Diagram	13
Tabel 2.3 Simbol-simbol <i>class</i> diagram	13
Tabel 2.4 simbol-simbol <i>sequence</i> diagram	15
Tabel 4.1 Identifikasi Aktor pada Sistem	25
Tabel 4.2 Identifikasi Aktor pada Sistem (lanjutan)	26
Tabel 4.3 kebutuhan Fungsional	26
Tabel 4.4 kebutuhan Fungsional (lanjutan)	27
Tabel 4.5 kebutuhan Fungsional (lanjutan 2)	28
Tabel 4.6 kebutuhan Fungsional (lanjutan 3)	29
Tabel 4.7 Kebutuhan Non Fungsional	29
Tabel 4.8 <i>Use Case Scenario</i> Login	31
Tabel 4.9 <i>Use Case Scenario</i> Register	32
Tabel 4.10 <i>Use Case Scenario</i> lihat materi	32
Tabel 4.11 <i>Use Case Scenario</i> lihat video	33
Tabel 4.12 <i>Use Case Scenario</i> lihat pengalaman	33
Tabel 4.13 <i>Use Case Scenario</i> tambah pengalaman	34
Tabel 4.14 <i>Use Case Scenario</i> komentar	34
Tabel 4.15 <i>Use Case Scenario</i> kerjakan soal	35
Tabel 4.16 <i>Use Case Scenario</i> lupa <i>password</i>	35
Tabel 4.17 <i>Use Case Scenario</i> edit profil	35
Tabel 4.18 <i>Use Case Scenario</i> lihat jadwal pelatihan	36
Tabel 4.19 <i>Use Case Scenario</i> tambah jadwal	36
Tabel 4.20 <i>Use Case Scenario</i> lihat user	37
Tabel 4.21 <i>Use Case Scenario</i> hapus user	37
Tabel 4.22 <i>Use Case Scenario</i> tambah soal	38
Tabel 4.23 <i>Use Case Scenario</i> hapus soal	38
Tabel 4.24 <i>Use Case Scenario</i> tambah materi	39
Tabel 4.25 <i>Use Case Scenario</i> hapus pengalaman	39
Tabel 4.26 <i>Use Case Scenario</i> hapus materi	40



Tabel 4.27 <i>Use Case Scenario</i> verifikasi pengalaman	40
Tabel 4.28 <i>Use Case Scenario</i> hapus jadwal	41
Tabel 4.29 <i>Use Case Scenario</i> kerjakan <i>quiz</i>	41
Tabel 5.1 <i>algoritme class register</i>	47
Tabel 5.2 <i>algoritme class</i> lihat video	48
Tabel 5.3 <i>algoritme class</i> tambah pengalaman	49
Tabel 6.1 Spesifikasi Perangkat Keras	55
Tabel 6.2 Spesifikasi Perangkat Lunak	55
Tabel 6.3 Implementasi kode program <i>register</i>	59
Tabel 6.4 Implementasi kode program tambah pengalaman	61
Tabel 6.5 Implementasi kode program lihat video	63
Tabel 7.1 <i>Pseudocode</i> Algoritma tambah soal	70
Tabel 7.2 Hasil Pengujian Unit Tambah soal	72
Tabel 7.3 <i>Pseudocode</i> Algoritma Edit Profile	73
Tabel 7.4 Hasil Pengujian Edit <i>Profile</i>	75
Tabel 7.5 Hasil Pengujian Edit <i>Profile</i> (Lanjutan)	76
Tabel 7.6 <i>Pseudocode</i> Algoritma tambah pengalaman	77
Tabel 7.7 Hasil Pengujian Tambah Pengalaman	79
Tabel 7.8 Hasil Pengujian Tambah Pengalaman (lanjutan)	80
Tabel 7.9 <i>Pseudocode method</i> onClick	80
Tabel 7.10 Hasil Pengujian Integrasi <i>method onclick</i>	82
Tabel 7.11 Pengujian validasi berhasil login	83
Tabel 7.12 Pengujian validasi <i>login</i> alternatif <i>username</i> salah	83
Tabel 7.13 Pengujian validasi login alternatif <i>password</i> salah	84
Tabel 7.14 Pengujian validasi berhasil <i>register</i>	84
Tabel 7.15 Pengujian validasi <i>register</i> alternatif salah satu kolom tidak diisi	85
Tabel 7.16 Pengujian validasi <i>register</i> alternatif <i>username</i> telah ada	85
Tabel 7.17 Pengujian validasi lihat materi	85
Tabel 7.18 Pengujian validasi lihat video	86
Tabel 7.19 Pengujian validasi lihat pengalaman	86
Tabel 7.20 Pengujian validasi komentar	86
Tabel 7.21 Pengujian validasi kerjakan soal	87



Tabel 7.22 Pengujian validasi lupa <i>password</i>	87
Tabel 7.23 Pengujian validasi lihat jadwal pelatihan.....	87
Tabel 7.24 Pengujian validasi tambah jadwal.....	88
Tabel 7.25 Pengujian validasi lihat <i>user</i>	88
Tabel 7.26 Pengujian validasi hapus <i>user</i>	88
Tabel 7.27 Pengujian validasi hapus soal.....	89
Tabel 7.28 Pengujian validasi tambah materi.....	89
Tabel 7.29 Pengujian validasi hapus materi.....	90
Tabel 7.30 Pengujian validasi verifikasi pengalaman.....	90
Tabel 7.31 Pengujian validasi hapus jadwal.....	91
Tabel 7.32 Pengujian validasi hapus jadwal.....	91
Tabel 7.33 Daftar Pertanyaan SUS.....	92
Tabel 7.34 hasil kuisioner SUS.....	94
Tabel 7.35 hasil perhitungan skor SUS.....	94
Tabel 0.1 hasil survey Bahasa isyarat.....	99



DAFTAR GAMBAR

Gambar 2.1 Metode <i>Waterfall</i>	8
Gambar 2.2 Android arsitektur.....	17
Gambar 3.1 Alur Metodologi Penelitian.....	21
Gambar 4.1 Gambaran umum sistem.....	24
Gambar 4.2 Use Case Diagram APP.....	30
Gambar 5.1 <i>sequence</i> diagram Tambah pengalaman.....	43
Gambar 5.2 <i>sequence</i> diagram lihat video.....	43
Gambar 5.3 <i>sequence</i> diagram <i>register</i>	44
Gambar 5.4 Perancangan Umum <i>Class Diagram</i>	45
Gambar 5.5 Perancangan Spesifik <i>Class Diagram</i>	45
Gambar 5.6 Perancangan Database APP.....	47
Gambar 5.7 Perancangan Antarmuka <i>register</i>	50
Gambar 5.8 Perancangan Antarmuka <i>register</i> 2.....	51
Gambar 5.9 Perancangan Antarmuka <i>Login</i>	52
Gambar 5.10 Perancangan Antarmuka <i>Video</i>	52
Gambar 5.11 Perancangan Antarmuka Dashboard.....	53
Gambar 5.12 Perancangan Antarmuka Soal.....	54
Gambar 5.13 Perancangan Antarmuka Materi.....	54
Gambar 6.1 Implementasi Tabel <i>User</i>	56
Gambar 6.2 Implemetasi Tabel jadwal.....	56
Gambar 6.3 Implementasi Tabel Materi.....	57
Gambar 6.4 Implementasi Tabel Pengalaman.....	57
Gambar 6.5 Implementasi Tabel Soal.....	58
Gambar 6.6 Implementasi Tabel Video.....	58
Gambar 6.7 Implementasi Tabel Timer.....	59
Gambar 6.8 Implementasi antar muka <i>register</i>	65
Gambar 6.9 implementasi antarmuka <i>login</i>	66
Gambar 6.10 Implementasi atarmuka <i>dashboard</i>	66
Gambar 6.11 implementasi antarmuka pengalaman.....	67
Gambar 6.12 Implementasi antarmuka lihat video.....	68



Gambar 6.13 Implementasi antarmuka soal.....	68
Gambar 6.14 Implementasi antarmuka Materi.....	69
Gambar 7.1 <i>Flowgraph</i> tambah soal.....	71
Gambar 7.2 <i>Flowgraph</i> edit <i>profile</i>	74
Gambar 7.3 <i>Flowgraph</i> tambah pengalaman.....	78
Gambar 7.4 <i>Flowgraph method onClick</i>	81
Gambar 7.5 presentil skor SUS.....	93
Gambar 7.6 Rating dan Skala Konversi SUS.....	93
Gambar 0.1 bagan presentase penguasaan Bahasa Isyarat.....	99



DAFTAR LAMPIRAN

Lampiran A Hasil wawancara.....	98
Lampiran B Hasil Survey Penguasaan Bahasa Isyarat.....	99



BAB 1 PENDAHULUAN

1.1 Latar Belakang

PSLD UB atau sebuah singkatan dari nama Pusat Studi dan layanan Disabilitas Universitas Brawijaya ialah suatu lembaga universitas yang mempunyai sebuah peran sebagai satu-satunya pusat penelitian yang membahas mengenai isu atau permasalahan yang berkaitan disabilitas, serta sebagai lembaga yang memberikan berbagai macam pelayanan kepada orang penyandang disabilitas di bumi Brawijaya. Penyandang disabilitas sendiri memiliki arti yakni individu atau seorang yang mempunyai sebuah batasan intelektual, fisik dan mental dalam waktu yang tidak sedikit, dimana pada kegiatan sehari-harinya ketika berinteraksi dengan masyarakat dan lingkungan memungkinkan memiliki hambatan-hambatan serta mampu membuat kesulitan untuk memiliki peran secara maksimal serta efektif berdasar dengan persamaan suatu hak (Undang-Undang dengan Nomor 19 pada Tahun 2011 Tentang Pengesahan Hak-Hak Penyandang Disabilitas) (Riadi, 2018).

Dalam memberi pelayanan untuk para mahasiswa disabilitas pada Universitas Brawijaya, lembaga ini bekerja sama dengan mahasiswa untuk memberikan pendampingan dalam kegiatan akademik sehari-hari. Mahasiswa penyandang disabilitas akan memperoleh pendampingan sesuai dengan kondisi disabilitas yang dimiliki mahasiswa tersebut agar dapat mengakses materi pembelajaran serta mengikuti aktivitas kegiatan akademik yang setara dengan mahasiswa-mahasiswa lainnya. Pemberian pendampingan pada dasarnya terinspirasi dari sebuah praktik *Shadow Teacher* yang memiliki definisi sebagai seorang pengajar yang terampil untuk bekerja sama dengan pengajar kelas dalam memberikan dukungan-dukungan seperti akademik, sosial, emosional dan perilaku untuk mahasiswa yang memiliki kebutuhan khusus (Pratiwi, Lintang Sari, Rizky, & Rahajeng, 2018). Rekrutmen pendamping yang dilakukan PS LD UB diadakan setiap semester sekali atau mungkin bisa lebih dari sekali, tergantung kebutuhan jumlah pendamping. Mahasiswa yang berminat untuk menjadi pendamping akan melewati tahapan seleksi, baik secara administratif dan wawancara.

Proses pelatihan pendampingan di PS LD UB dimulai dari pengenalan awal mengenai materi *Disability Awareness* kemudian dilanjutkan dengan pengenalan mahasiswa difabel tuli dan pelatihan alphabet awal Bahasa isyarat, kemudian pelatihan mengenai tunadaksa dengan mempraktekan mendampingi cara berjalan menggunakan kursi roda, selanjutnya pelatihan mendampingi mahasiswa tuna netra dan diakhiri dengan pelatihan mahasiswa autisme dalam peranannya. Mahasiswa terkadang cenderung lupa mengenai materi-materi yang telah disampaikan dalam pelatihan, terutama pelatihan yang berkaitan dengan Bahasa isyarat yang harus dilatih secara terus menerus dan mengingat kata-kata yang perlu dipelajari dalam berkomunikasi sehari-hari.



Dalam peranannya pendamping di universitas brawijaya tentunya membutuhkan banyak jam pendampingan untuk bisa menguasai penggunaan Bahasa isyarat, namun tidak setiap hari mendapatkan jadwal mendampingi mahasiswa tuli. Solusi yang ada pada PSLD UB setiap 3 minggu sekali dijadwalkan pelatihan Bahasa isyarat dengan durasi sekitar 1-2 jam yang dipimpin oleh seorang tutor Bahasa isyarat dimana calon pendamping baru diajarkan kosa kata Bahasa isyarat kehidupan sehari-hari seperti kalimat Tanya dan kalimat sapa kemudian diajarkan mengenai huru-huruf dasar alphabet dari A-Z untuk pendamping baru bisa mengeja kata. Berdasar hasil survey penguasaan Bahasa isyarat dari 21 pendamping di universitas brawijaya yang dibagi menjadi 3 kategori yakni menghasilkan 42,3 % kurang, 50 % cukup dan 7,7 % masuk kategori baik yang bisa dilihat pada lampiran B. Hal ini membuktikan sebagian besar pendamping di PSLD UB diperlukan adanya pelatihan secara lanjutan.

Berdasarkan akan permasalahan diatas dibutuhkan sebuah perangkat lunak atau aplikasi pembelajaran online yang menunjang pelatihan lanjutan bagi pendamping mengenai materi-materi yang telah disampaikan dalam proses seleksi, yang akan dikembangkan pada skripsi ini. Perangkat lunak akan dikembangkan dengan android sebagai platform. Proses pengembangan perangkat lunak menggunakan konsep pemrograman berorientasi object atau OOP yang dimodelkan dengan teknik *Unified Modeling Language* atau UML dengan menggunakan Bahasa java dan android studio sebagai IDE serta menggunakan Firebase sebagai database. Pengembangan ini juga menggunakan suatu metode SDLC (Software Development Life Cycle) atau sebuah daur hidup pengembangan *software*. SDLC yang digunakan dalam pengembangan ini yaitu *Waterfall*, SDLC ini dipilih karena setiap kebutuhan dari pengembangan *software* telah teridentifikasi jelas pada tahap awal. Tahapan proses pengembangan SDLC ini dimulai dari proses perencanaan (*planning*), tahap analisis (*analysis*), proses desain (*design*), tahap implementasi (*implementation*), pengujian (*testing*) dan terakhir pemeliharaan (*maintenance*) (Simarmata, 2010).

1.2 Rumusan Masalah

Mengacu uraian dari latar belakang dan permasalahan yang telah dijabarkan diatas, diperoleh beberapa rumusan masalah diantaranya :

1. Bagaimana hasil rekayasa kebutuhan Aplikasi Pelatihan Pendamping PSLD UB ?
2. Bagaimana hasil perancangan aplikasi Pelatihan Pendamping PSLD UB ?
3. Bagaimana implementasi aplikasi Pelatihan Pendamping PSLD UB ?
4. Bagaimana hasil pengujian aplikasi Pelatihan Pendamping PSLD UB ?

1.3 Tujuan

Mengacu dari rumusan masalah di atas diperoleh tujuan dari penelitian diantaranya yaitu :



1. Mengetahui hasil rekayasa kebutuhan Aplikasi Pelatihan Pendamping PSLD UB.
2. Mengetahui hasil perancangan aplikasi Pelatihan Pendamping PSLD UB.
3. Mengetahui implementasi aplikasi Pelatihan Pendamping PSLD UB.
4. Mengetahui hasil pengujian aplikasi Pelatihan Pendamping PSLD UB.

1.4 Manfaat

Berdasar pada penelitian ini kemudian diharapkan dapat memberi manfaat atau hal positif kepada calon pendamping mahasiswa difabel universitas brawijaya sebagai media pembelajaran lanjutan mengenai materi-materi pembelajaran dan pelatihan tentang *disability awareness* sehingga calon pendamping nantinya akan lebih siap dalam melakukan pendampingan pada kegiatan-kegiatan akademik mahasiswa penyandang disabilitas Universitas Brawijaya.

1.5 Batasan Masalah

1. Penelitian hanya dilakukan pada PSLD Universitas Brawijaya.
2. Aplikasi dikembangkan untuk platform android.
3. Materi pelatihan *disability awareness* mengacu pada PSLD UB.

1.6 Sistematika Pembahasan

Dalam proses penyusunan laporan, disusun dengan tujuan sebagai media yang memberikan sebuah pandangan atau gambaran dan juga uraian detail yang terkait dengan sebuah penelitian yang sedang dijalankan, yang secara garis besar terdapat berbagai macam bab berikut ini :

BAB I PENDAHULUAN

Pada bab berikut ini akan dibahas beberapa keterangan yang berkaitan dengan latar belakang dari suatu masalah, rumusan dari permasalahan, hal yang dituju dari penelitian, kemudian manfaat yang diperoleh dari suatu penelitian, mengenai batasan-batasan pada penelitian dan juga tentang sistematika di dalam penyusunan laporan penelitian.

BAB II LANDASAN KEPUSTAKAAN

Bab berikut landasan suatu kepastakaan berisi berbagai macam penjelasan mengenai sebuah dasar berbagai teori yang dipergunakan pada penelitian yang dilaksanakan, kemudian dijadikan dasar untuk membangun aplikasi yang dikembangkan dalam penelitian.



BAB III METODOLOGI PENELITIAN

Pada pembahasan didalam bab ini berisi mengenai metode yang dilakukan dalam suatu penelitian yang dilaksanakan yang diantaranya terdiri antaranya studi literatur, dilanjutkan perancangan dan implementasi perangkat lunak, pengujian serta yang berkaitan dengan analisis pada aplikasi yang dikembangkan.

BAB IV REKAYASA KEBUTUHAN

Bab rekayasa kebutuhan memaparkan mengenai bagaimana menganalisa kebutuhan dari aplikasi yang kita kembangkan dan berisi tentang pemodelan dari alur berjalannya suatu aplikasi yang sedang dikembangkan dengan menggunakan *Unified Modeling Language* atau dikenal sebagai UML.

BAB V PERANCANGAN

Kemudian bab ini berisi mengenai *design-design* perancangan pada aplikasi yang akan dikembangkan dan juga berisi perancangan sebuah basis dari data yang akan digunakan.

BAB VI IMPLEMENTASI

Bagian pada bab ini merupakan kelanjutan dari proses perancangan yang berisi mengenai implementasi-implementasi dari hasil sebuah perancangan yang kemudian telah dilakukan sebelumnya.

BAB VII PENGUJIAN

Pada bagian terakhir ini berisi mengenai bagaimana sistem yang sudah diimplementasikan pada sebelumnya di uji, pengujian dilakukan dengan metode *blackbox* dan *whitebox* dengan berbagai tahapan pengujian yang telah disusun untuk menguji berbagai parameter kelayakan.

BAB VIII PENUTUP DAN SARAN

Bab penutup terakhir ini berisi beberapa kesimpulan yang dihasilkan dari sebuah sistem yang telah melalui proses pengujian dan berisi mengenai saran yang di tujukan untuk pengembangan selanjutnya yang dilakukan oleh pengembang lain



BAB 2 LANDASAN KEPUSTAKAAN

2.1 PSLD UB

Pusat Studi dan Layanan Disabilitas Universitas Brawijaya atau biasa dikenal dengan singkatan PSLD UB memiliki pengertian suatu lembaga yang memiliki fungsi untuk sebuah pusat dari penelitian mengenai isu yang berkaitan dengan *disabiliy* serta pemberian layanan untuk penyandang disabilitas yang berada di bumi Brawijaya. Sejarah PSLD berdiri pada tanggal 19 Maret 2012. Hal tersebut memiliki latar belakang suatu peristiwa tidak adanya mobilitas sebuah PTN untuk seluruh penderita disabilitas, meskipun mengenai hak tentang pendidikan kategori non-diskriminatif untuk penderita disabilitas terdapat perlindungan oleh Undang-Undang serta Konvensi Internasional.

Karena terjadinya keterbatasan akses ke pendidikan dengan jenjang S1 atau perguruan tinggi untuk penderita disabilitas, akibatnya kurang dari 1% berbagai pemilik disabilitas yang memiliki ijazah Sarjana. Di negara kita sendiri, bentuk pengajaran bagi orang pemilik disabilitas masihlah melalui cara segregatif yakni memberikan sebuah pengajaran yang berjalan dengan cara tertentu yang umumnya disebut SLB (Sekolah Luar Biasa). Cara tersebut melakukan perbedaan antara mahasiswa penderita disabilitas pada mahasiswa non-disabilitas yang terdapat pada lingkungan yang berbeda-beda, oleh karena itu dalam peranannya setelah masa studi berakhir kebanyakan mahasiswa pemilik disabilitas sangat tidak siap guna menyesuaikan diri atau beradaptasi pada lingkungan yang berada disekitarnya. Untuk lulus PTN juga menjadi sulit untuk mahasiswa pemilik disabilitas hal ini karena terdapat berbagai syarat untuk diterima di PTN “tidak mempunyai kelainan atau cacat anggota tubuh” dan yang berkaitan dengan kekurangan-kekurangan lainnya. Permasalahan ini juga terdapat pada UB yang pada masa lampau tidak memiliki sarana dan macam prasarana yang dapat dijangkau oleh para pemilik disabilitas. Keilmuan penduduk tentang bumi Brawijaya juga amat kurang mengenai isu yang berkaitan disabilitas, meskipun salah satu filosofi UB yang tertulis pada lambang serta logo UB, yakni universitas yang dinamis, bersifat universal, dan berkeadilan.

Berasal sebuah proses audiensi yang terjadi tanggal 31 Januari pada tahun 2012, pimpinan Universitas Brawijaya setuju untuk dapat melakukan penerimaan terhadap rancangan kampus inklusif yang merupakan bagian dari kebijakan kampus . hal-hal kesepakatan tersebut , yaitu :

1. Menjadikan PSLD UB sebagai suatu lembaga yang memberi fasilitas penyandang disabilitas dan menjalankan kegiatan dalam ranah *grand design*;
2. Merealisasi penerimaan SPKPD dengan memberi 20 kuota, menjadi salah satu implementasi secara afirmatif guna memberi suatu akses bagi pemilik disabilitas;
3. Pimpinan atau rektor UB juga memberi beasiswa untuk mahasiswa/i pemilik disabilitas yang masuk kategori tidak berkecukupan.



2.2 Kajian Pustaka

Kajian pustaka merupakan sebuah bagian yang menjadi acuan sebuah penelitian yang sedang berjalan. Kajian pustaka ini menjadi penting karena dijadikan sebuah dasar dalam pengembangan sebuah sistem atau aplikasi. Pada penelitian tersebut kajian pustaka yang akan berguna berkaitan E-learning berbasis mobile PSLD Universitas Brawijaya. Ada berbagai macam penelitian terkait E-learning namun dari penelitian tersebut, terdapat 3 penelitian yang diacu menjadi bahan referensi. Dari ketiga hasil penelitian yang menjadi referensi tersebut dianalisis beberapa fitur-fitur yang akan menjadi inspirasi dalam pembangunan sistem.

Penelitian pertama yang memiliki judul *App Based-Term Care with E-Learning* yang ditulis oleh Anthony Y.H. Liao dan Shun-Pin Huang yang berasal dari Asia Univ., Taichung Taiwan. Penelitian tersebut berisi mengenai sebuah pelatihan-pelatihan yang berisi tatacara mempersiapkan seorang perawat dalam melakukan perawatan terhadap orang-orang di Taiwan yang berusia diatas 65 tahun yang dikembangkan dengan menggunakan platform mobile yang memiliki mobilitas tinggi sehingga bisa untuk latihan dimana saja dan kapan saja. Pada aplikasi tersebut juga diharapkan mampu mempersiapkan tenaga kerja keperawatan lansia jangka panjang.

Penelitian kedua ditulis oleh Li Ling Hsu dari Nat. Taipei Univ. of Nursing & health Sci Taipei yang memiliki judul *The Design of Simulation-Based e-learning Mobile Application Software*. Pada penelitian ini berisi mengenai e-learning yang berkaitan dengan pemeriksaan fisik dan kursus pelatihan penilaian kesehatan pada pengembangan kompetensi klinis sekelompok mahasiswa dengan studi keperawatan. E-learning ini dibuat karena sangat sedikit suatu penelitian yang melakukan eksplorasi terhadap hasil dari proses belajar mahasiswa keperawatan yang melakukan pelatihan penilaian kesehatan dan pemeriksaan fisik. Pemeriksaan fisik dan penilaian kesehatan berbasis simulasi dapat memberikan lingkungan yang nyaman.

Kemudian untuk penelitian ketiga yang memiliki judul “ Pengembangan aplikasi *Mobile Learning* berbasis platform Android pada Pelajaran RPL pada SMK Sultan Trenggono Kota Semarang ” yang ditulis oleh Wulandari Dania Ayu, Wibawanto Hari, Suryanto Agus, Murnomo Agus. Pada penelitian ini berisi bagaimana memanfaatkan media pembelajaran menggunakan aplikasi mobile android pada mata pembelajaran Rekayasa Perangkat Lunak dengan dikembangkan dengan 4-D (Four D Models) yaitu *define* (pendefinisian), *design* (desain), *develop* (pengembangan), *disseminate* (penyebaran). Dari penelitian-penelitian tersebut menunjukkan pembelajaran dengan *platform mobile* semakin sering digunakan karena berbagai kemudahan dan metode pembelajaran yang lebih fleksibel.



2.3 Rekayasa Perangkat Lunak

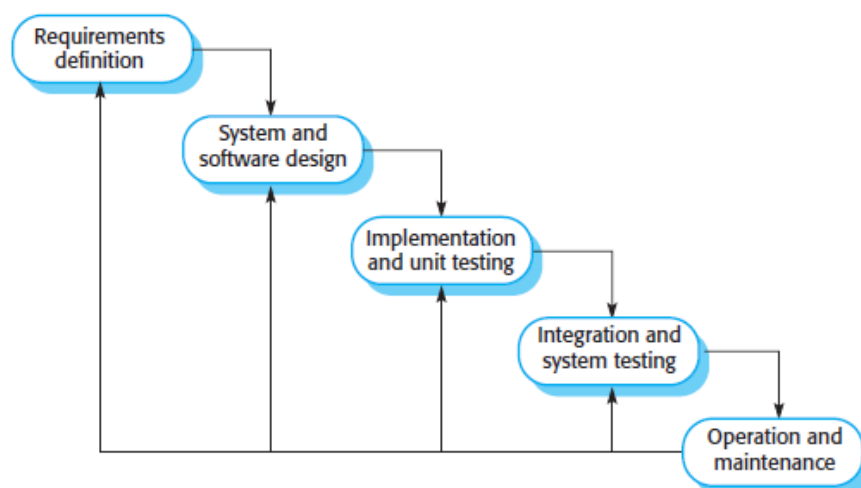
Rekayasa atau sebuah teknik memiliki pengertian pengimplementasian sebuah teknologi dan ilmu dengan tujuan menyelesaikan suatu permasalahan yang dialami seseorang. Hal tersebut dapat diselesaikan melalui matematika dan pengetahuan, serta pengetahuan secara praktis yang di implementasikan untuk melakukan desain suatu objek ataupun proses yang memiliki manfaaat. Sementara Rekayasa Perangkat Lunak atau yang biasa disingkat dengan RPL adalah suatu bidang pekerjaan yang memperdalam cara-cara melakukan pengembangan perangkat lunak yang didalamnya terdapat proses yang terdiri dari pembuatan, manajemen dari organisasi pengembangan pada perangkat lunak, pemeliharaan dan lain sebagainya (Simarmata, 2010). Dalam sebuah pemerintahan rekayasa perangkat lunak memiliki suatu peran yang cukup penting hal ini disebabkan karena tanpa adanya rekayasa perangkat lunak, sebuah negara tidak bisa menjalankan pemerintahannya secara modern hal ini juga karena infrastruktur nasional dan perawatannya dikontrol berdasarkan software atau perangkat lunak (Sommerville, 2016).

2.4 Pengembangan Perangkat Lunak

Pengembangan suatu *software* atau perangkat lunak adalah sebuah proses yang dilakukan dengan tujuan mengembangkan sebuah perangkat dengan tujuan tertentu juga dengan pengembangan perangkat lunak pelatihan pendamping PSLD UB. Dalam proses pengembangannya diperlukan ketelitian dan kesabaran dalam merancang dan juga menjalankan guna mencapai tujuan yang diinginkan. Ada berbagai macam bahasan mengenai proses pengembangan perangkat lunak diantaranya seperti model dari terciptanya perangkat lunak tersebut dengan memakai platform mobile android dengan metode pengembangan *waterfall* dikembangkan dengan bahasa pemrograman JAVA dengan *Object Oriented Programming* (OOP).

2.4.1 Model Waterfall

Pada proses pengembangan aplikasi pelatihan pendamping digunakan model *Waterfall* dikarenakan sudah didefinisikan secara jelas kebutuhan dari pengembangan aplikasi pelatihan pendamping. dimana SDLC ini juga sebagai sebuah metodologi penelitian yang langkah-langkahnya dibahas pada gambar 2.1. metode *waterfall* juga disebut sebagai metode air terjun model ini merupakan suatu contoh dari proses yang digerakan oleh sebuah rencana setidaknya inti pada pemodelan ini adalah anda melakukan perencanaan dan melakukan penjadwalan terhadap semua proses kegiatan sebelum dilakukan tahapan pembuatan perangkat lunak. Pada umumnya berbagai tahapan yang terkandung pada metode *waterfall* yaitu seperti *requirement analysis* dan *definition*, kemudian *system* dan *software design*, selanjutnya *implementation* dan *unit testing, integration and testing* dan terakhir *operation and maintenance* (Sommerville, 2016).



Gambar 2.1 Metode Waterfall

(Sumber : Sommerville,2016)

1. Requirement Analysis

Pada tahapan awal ini merupakan suatu tahapan penetapan fitur, mendefinisikan dengan detail fitur pada sistem, permasalahan serta tujuan konsultasi yang berasal dari pengguna sistem berfungsi menjadi kelengkapan sistem. Selanjutnya terdapat pada pengembangan sistem digunakan permintaan kebutuhan atau *requirement analysis* dengan melakukan wawancara untuk memperoleh kebutuhan yang dimaksudkan. Pada studi kasus pendampingan pada PS LD UB langkah *requirement analysis* dilakukan dengan melakukan wawancara kepada salah satu koordinator yang bertanggung jawab mengenai *disability awareness*. Beberapa pertanyaan yang diajukan mulai dari proses dari pendaftaran pendamping hingga proses pembekalan yang diberikan kepada calon pendamping. Kemudian kebutuhan disusun berdasar hasil wawancara yang kemudian dijadikan dasar dalam pengembangan sistem lebih lanjut.

2. System and Software Design

Pada proses tahapan berikut ini kebutuhan yang diperoleh dari tahapan sebelumnya kemudian didesain sesuai dengan yang dibutuhkan. Desain ini berfungsi sebagai bahan pertimbangan dalam menentukan perangkat keras atau *hardware* yang dipergunakan dalam pengembangan sistem serta dapat membantu dalam mendetailkan arsitektur dari sistem secara menyeluruh. Pada pengembangan aplikasi pelatihan pendampingan perangkat keras yang digunakan untuk membangun aplikasi pelatihan menggunakan laptop dengan RAM minimal 8 GB dan untuk softwarenya



menggunakan IDE Android studio dengan menggunakan Bahasa pemrograman java. Kemudian layoutnya menggunakan XML.

3. Implementation dan Unit Testing

Setelah sistem di desain sesuai dengan kebutuhan yang didapat, pada tahapan ini terjadi pengembangan awal dari sistem melalui program-program yang dikenal dengan nama unit. Yang kemudian terintegrasi pada tahap setelahnya. Pada proses ini semua unit akan diterapkan verifikasi yang sesuai dengan spesifikasi yang ada pada sebelumnya. Proses pengembangan aplikasi pelatihan pendamping dibangun dengan berbagai fungsi yang saling terintegrasi dan membangun sebuah sistem yang bertujuan menghasilkan suatu output tertentu. Diantaranya fungsi untuk menyimpan sebuah materi, menambahkan soal dan fitur untuk sharing antar pendamping sehingga bisa saling berbagi pengalaman.

4. Integration dan Testing

Pada tahapan ini setelah seluruh unit yang telah dikembangkan kemudian dilakukan integrasi setelah seluruh unit melewati tahap pengujian. Setelah semua sistem dilakukan pengujian untuk mengetahui tiap kegagalan dan kesalahan pada sistem. Tahap ini dilakukan ketika setiap fungsi pada aplikasi pelatihan pendamping telah diimplementasikan menjadi sebuah fungsi dan kemudian akan dilakukan sebuah pengujian dengan menggunakan metode yang umum dilakukan, diantaranya *white box testing* dengan melakukan proses pengujian terhadap unit serta memakai *black box testing* dengan melakukan pengujian tipe validasi yang bertujuan agar setiap fungsi sistem berjalan sesuai dengan keinginan.

5. Operation and maintenance

Tahapan ini adalah proses tahapan akhir dari pengembangan model *waterfall* perangkat lunak akan dioperasikan dan dilakukan perawatan. Perawatan termasuk dalam melakukan perbaikan terhadap kesalahan yang kemudian tidak terdeteksi pada tahapan yang telah dilalui sebelumnya. Pada proses dalam kegiatan penelitian yang dilakukan ini pengembangan memakai metode *waterfall* dan hanya sampai tahapan integrasi dan *testing*. Pada tahap akhir pengembangan aplikasi pelatihan pendamping yang dilakukan adalah memeriksa semua fungsi telah berjalan sesuai apa yang dibutuhkan.

2.4.2 Object Oriented Programming (OOP)

Object Oriented Programming ataupun yang biasa terkenal dengan OOP ialah suatu metode pemrograman yang orientasinya berdasarkan pada objek. OOP diciptakan dengan tujuan guna memberi kemudahan dalam penciptaan suatu program dengan mengadopsi dalam kehidupan manusia pada umumnya.



Pada pemrograman ini menganggap setiap permasalahan adalah objek, yang dimana objek adalah penyatuan yang berasal dari bermacam objek yang skalanya lebih kecil lagi. OOP memiliki beberapa karakteristik diantaranya (Clark, 2013).

A. *Inheritance*

Pada pemrograman OOP karakter *inheritance* dipergunakan guna memisahkan objek pada program yang sesuai dengan fungsi dan karakteristik yang pada umumnya. Hal tersebut menyebabkan bekerja menggunakan orientasi objek menjadikan lebih tertata serta intuitif. Serta menyebabkan implementasi pemrograman akan menjadi tertata dan lebih efisien, karena terdapat kemungkinan terjadinya penggabungan berbagai macam objek yang umum pada objek kategori *parent* dan bisa melakukan pewarisan karakter objek tersebut pada sebuah *child*.

B. *Abstraction*

Abstraksi pada OOP yaitu melakukan penyaringan terhadap sifat-sifat benda yang tidak dikenal sehingga hal tersebut hanya melakukan pemrosesan sebagian konsentrasi dan informasi yang tergolong penting dalam tugas-tugas yang dalam pengerjaan.

C. *Encapsulation*

Enkapsulasi didalam pemrograman OOP merupakan proses yang berguna untuk melakukan manajemen *class* yang berada didalam tidak memiliki akses secara langsung ketika yang dikirim sebuah data, tetapi memiliki sifat tersembunyi. Oleh karenanya pada hal ini ketika suatu objek ingin mengakses suatu data dari objek lain, harus melalui atau berinteraksi dengan objek lain yang memiliki tanggung jawab dalam hal pertukaran data tersebut.

D. *Aggregation*

Pada pemrograman OOP agregasi memiliki pengertian yaitu suatu keadaan ketika suatu objek adalah penggabungan dari beberapa komponen atau bagian-bagian lain yang bekerja secara bersama-sama. Sebagai contoh motor yang merupakan gabungan dari berbagai onderdil dan sperpart yang saling bekerja sama untuk menggerakkan satu tujuan. Kemampuan dalam penggunaan agregasi pada OOP merupakan salah satu fitur yang canggih yang memiliki kemungkinan implementasi pada proses bisnis serta model yang secara tepat /akurat pada sebuah program.



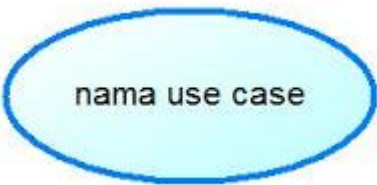
2.4.3 Unified Modeling Language (UML)

Unified Modeling Language atau yang biasa memiliki sebutan UML memiliki arti merupakan sebuah standar sebuah bahasa didalam pemodelan yang dipergunakan untuk melakukan pembangunan *software* atau perangkat lunak dimana cara pengembangannya atau dibuat dengan menggunakan sebuah teknik pemrograman dengan menggunakan objek sebagai orientasi. (A.S & Shalahuddin, 2015). UML lahir karena sebuah kebutuhan akan Bahasa yang digunakan untuk memodelkan perangkat lunak, sehingga diperlukan sebuah standarisasi yang dapat menyatukan pemikiran dari bermacam-macam orang sehingga akan mudah dimengerti oleh orang banyak dari berbagai negara. UML ialah sebuah Bahasa penglihatan atau visual yang berguna sebagai pemodelan serta aktivitas komunikasi yang berkaitan dengan sistem yang memakai media diagram serta teks-teks sebagai pendukung. UML sendiri terdiri atas berbagai macam diagram diantaranya.

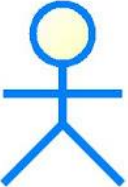
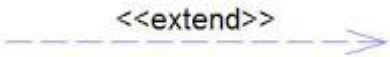
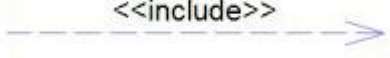
a. Use Case Diagram

Use case dikenal dengan nama diagram *use case* adalah suatu bentuk pemodelan yang berguna untuk mengidentifikasi tingkah laku (*behavior*) sistem yang terencana akan dibangun. Diagram ini menjelaskan sebuah komunikasi antara berbagai aktor beserta sebuah sistem yang terencana akan dibangun, mendefinisikan satu atau lebih aktor yang saling berinteraksi dalam sebuah keterlibatan pada sistem. Gambaran kasarnya *use case* memiliki fungsi untuk dapat mengetahui apa-apa saja fitur yang berada dalam sebuah sistem serta siapa saja yang memiliki wewenang untuk memakai fitur-fitur tersebut. Komponen-komponen pada diagram akan dijelaskan lebih pada Tabel 2.1.

Tabel 2.1 Simbol *use case* diagram

Simbol	Deskripsi
	<i>Use case</i> merupakan fungsionalitas yang tersedia oleh sistem berbentuk unit yang satu sama lain saling bertukar sebuah pesan antara unit ataupun aktor. biasa penamaannya menggunakan kata kerja yang diberikan pada awal frasa penamaan <i>use case</i>.



Aktor / actor  nama aktor	Aktor merupakan proses, individu , atau sistem yang lain saling melakukan interaksi dengan sebuah sistem terkait yang akan dibuat.
Asosiasi / association 	Asosiasi merupakan sebuah interaksi antar aktor dengan <i>use case</i> yang sedang terlibat dengan diagram ataupun <i>usecase</i> yang sedang mempunyai komunikasi dengan suatu aktor terkait. Asosiasi berguna untuk melakukan hubungan link antara element
Ekstend / extend 	Relasi pada <i>use case</i> memiliki fungsi tambahan pada suatu <i>use case</i> dimana sebuah <i>case</i> yang ditambah dapat memiliki kemampuan berdiri mandiri meskipun tanpa sebuah <i>use case</i> tambahan tersebut arah panah memiliki arah pada sebuah fitur yang ditambah.
Include 	Relasi <i>use case</i> mempunyai fungsi tambahan kepada suatu <i>use case</i> yang <i>use case</i> tersebut telah ditambahkan membutuhkan suatu <i>use case</i> tersebut guna menjalankan suatu fungsinya atau berguna sebagai suatu syarat untuk dijalankan <i>use case</i> ini arah panah include tersebut mengarah ke <i>use case</i> yang sedang dipakai (dibutuhkan) atau mengarah kepada <i>use case</i> tambahan.
Generalisasi / generalization 	Hubungan suatu generalisasi dan spesialisasi (umum - khusus) merupakan suatu hubungan antara dua <i>use case</i> yang dimana memiliki suatu fungsi yang merupakan sebuah fungsi yang memiliki sifat umum dari yang lain. arah panah mengarah kepada <i>use case</i> yang menjadi suatu generalisasinya (umum)

(Sumber : Sommerville,2016)

b. Class Diagram

Class diagram atau yang biasa dikenal dengan diagram kelas memiliki fungsi untuk penggambaran struktur sistem dari sebuah segi penjelasan atau definisi berbagai kelas yang dilakukan pembuatan pembangunan



suatu sistem. Kelas mempunyai yang disebut metode atau operasi dan atribut. Atribut memiliki pengertian merupakan suatu variabel terkait yang dipunyai oleh sebuah kelas. Sedangkan metode merupakan suatu fungsi yang kemudian dipunyai oleh sebuah kelas terkait. Diagram kelas biasanya dibuat dengan tujuan supaya pembuat program atau yang biasa disebut *programmer* akan membuat kelas tersebut memiliki kesesuaian dengan rancangan pada kelas diagram agar antara suatu dokumen dengan perancangan serta perangkat lunak bisa menjadi sinkron. Berikut contoh kelas diagram pada Tabel 2.2. Serta komponen dijelaskan pada Tabel 2.3.

Tabel 2.2 contoh Class Diagram

Class Diagram	
Anggota Id_agt Nama_agt Alamat_agt Email_agt getIdAngt() setIdAngt() getNamaAngt() setNamaAngt() getAlamatAngt() setAlamatAngt() setEmailAngt() getEmailAngt()	Telepon Id_anggota Telepon_angt getId_angt() setId_angt() getTeleponAgt() setTeleponAgt() queryMemasukanTelepon() queryMengubahTelepon() queryMencariTelepon() queryMelihatTelepon()

Tabel 2.3 Simbol-simbol *class* diagram

Simbol	Deskripsi
Kelas Nama_kelas + atribut + operasi()	Kelas pada struktur sistem
Antarmuka / <i>interface</i> ○ Nama_interface	Memiliki makna yang sama konsep <i>interface</i> pada pemrograman dengan orientasi objek



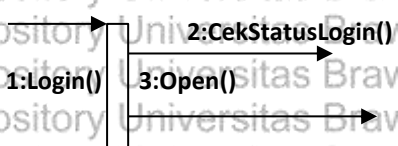
Asosiasi / <i>association</i> 	Memiliki pengertian sebuah relasi antara kelas dengan memiliki sebuah makna yang umum, biasanya disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Sebuah relasi antara kelas-kelas yang memiliki arti kelas yang satu dipergunakan oleh beberapa kelas yang lain, asosiasi juga disertai dengan <i>multiplicity</i>
Generalisasi 	Memiliki makna relasi antara kelas dengan makna generalisasi-spesialisasi (umum-khusus)
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan arti/makna Ketergantungan antar kelas
Agregasi / <i>agregation</i> 	Relasi antar kelas yang mempunyai makna semua-sebagian (<i>whole-part</i>)

(Sumber : Sommerville, 2016)

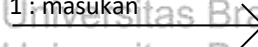
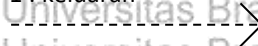
c. Sequence Diagram

Sequence diagram atau yang biasa disebut dengan diagram sekuen memiliki fungsi untuk menggambarkan objek terhadap *use case* dengan melakukan pendeskripsian waktu hidup suatu objek serta *pesan* yang dikirimkan dan diterima antara objek. Karenanya guna menggambarkan suatu sequen diagram tentunya mengetahui berbagai macam objek yang dilibatkan pada suatu *use case* serta metode yang telah dimiliki oleh sebuah kelas yang di instansiasikan menjadi bentuk objek tersebut. Diagram sequen dibutuhkan dalam penglihatan skenario yang terdapat pada *use case*. Berikut beberapa symbol yang terdapat pada komponen diagram sequen yang dijelaskan pada Tabel 2.4 berikut.

Tabel 2.4 simbol-simbol *sequence diagram*

Simbol	deskripsi
Aktor  Nama_aktor	Aktor dapat berupa individu, sebuah proses ataupun sistem yang lain yang melakukan interaksi terhadap sistem atau aplikasi, satu informasi yang kemudian dibuat yang tertelak pada eksternal sistem yang akan dibuat sendiri.
Garis hidup / <i>lifeline</i> 	Menyatakan kondisi dimana suatu objek hidup.
Objek Nama objek : nama kelas	Menyatakan suatu objek yang sedang berinteraksi pesan.
Waktu aktif 	menjabarkan kondisi dimana objek hidup dan sedang melakukan interaksi, semua yang sedang terhubung dengan status waktu aktif tersebut adalah suatu proses yang dilakukan di dalamnya, contohnya  Oleh karenanya cekstatuslogin() dan open() dijalankan pada metode login()
Pesan tipe create <<create>> 	Menyatakan keadaan suatu objek menciptakan sebuah objek yang lain, sedangkan ujung arah panah mengacu pada objek yang telah diciptakan.



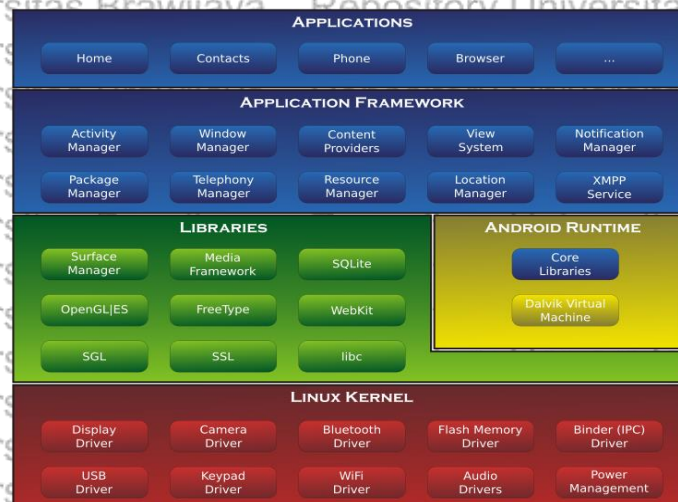
Pesan tipe call 1 : nama_metode() 	Menjabarkan keadaan/kondisi suatu objek sedang melakukan panggilan operasi/metode yang terdapat pada objek yang lain atau diri sendiri.  1 : nama_metode() Arah anak panah terlihat mengacu pada objek yang mempunyai metode/operasi, disebabkan karena hal ini sedang memanggil metode oleh suatu metode yang sedang dipanggil harus terdapat pada diagram kelas sesuai dengan kelas objek yang sedang melakukan interaksi
Pesan tipe send 1 : masukan 	Suatu keadaan dimana suatu objek mengirim suatu inputa atau data informasi kepada objek yang lainnya, arah anak panah menunjukan pada objek yang sdang dikirim data
Pesan tipe return 1 : keluaran 	Suatu keadaan dimana menjelaskan bahwa suatu objek yang telah menjalankan sebuah operasi atau sebuah metode menghasilkan suatu pengembalian atau nilai balik ke objek bersifat tertentu. Arah anak panah menuju pada objek yang dituju
Pesan tipe destroy 	Suatu keadaan dimana menjelaskan sebuah objek sudah mematikan hidup objek lain, lalu arah panah menuju pada sebuah objek yang akan diakhiri, dan sebaiknya jika terdapat create maka ada juga destroy.

(Sumber : Sommerville,2016)

2.5 Teknologi Pengembangan Sistem

2.5.1 Android

Android menjadi salah satu dari sistem informasi yang dirancang secara khusus untuk pengguna smartphone dan android. Android sendiri dikembangkan secara langsung oleh lembaga Android Inc. disertai dengan dukungan penuh dari google tapi kemudian pada tahun 2005 dibeli. Pada sistem informasi ini menggunakan linux sebagai basisnya atau linux dijadikan dasar. Linux memiliki arti sebuah sistem operasi atau os yang dilakukan perancangan secara khusus guna perangkat komputer. Sistem operasi android dirancang secara khusus untuk diinstall pada perangkat bergerak dengan layar sentuh sehingga secara khusus sistem operasi yang terdapat pada perangkat mobile ini beradaptasi dari spesifikasi *low-end* hingga mencapai *high-end*. Sistem operasi android tergolong pada sistem operasi yang bersifat terbuka atau biasa dikenal *open source* (Hermawan, 2019). Berikut arsitektur dari android pada Gambar 2.2.



Gambar 2.2 Android arsitektur

(Sumber : Source.android.com)

Android memakai sebuah kernel yang memiliki basis kernel OS linux dengan versi 3.x (versi 2.6 pada tipe android 4.0 atau *ice cream sandwich* begitu juga sebelumnya) hal-hal seperti gudang perpustakaan perangkat lunak, piranti tengah dan API ditulis menggunakan Bahasa C, serta perangkat lunak aplikasinya beroperasi pada sebuah kerangka kerja aplikasi.



2.5.2 Firebase Database

Firebase merupakan suatu layanan yang berasal dari google yang dibuat dengan tujuan untuk mempermudah berbagai pengembang aplikasi untuk mengembangkan aplikasi. Dengan diciptakannya firebase para pengembang aplikasi bisa untuk fokus mengembangkan aplikasi tanpa usaha yang besar karena firebase menyediakan berbagai macam fitur yang mempermudah pengembang dalam pekerjaannya. Dalam firebase terdapat beberapa fitur yang bisa digunakan seperti *realtime database* dan *firebase storage*. Dari beberapa fitur yang ada pada firebase tentunya akan kita gunakan sesuai dengan kebutuhan pengembangan aplikasi kita. Fitur firebase yang akan dipergunakan dalam proses pengembangan aplikasi adalah *realtime database* dan *storage*.

2.5.2.1 Firebase Realtime Database

Adalah sebuah layanan firebase yang menyediakan pengembang antarmuka pemrograman aplikasi yang memungkinkan sebuah data dari aplikasi disinkronkan pada semua klien dan disimpan pada firebase cloud. Database juga bisa dilakukan akses melalui sebuah REST API. Pengembang yang menggunakan sebuah realtime database dapat melakukan pengamanan data dengan memakai aturan keamanan yang dimiliki perusahaan yang dimilikinya. Firebase realtime adalah suatu database yang dilakukan host pada sebuah cloud. Data disimpan dengan format JSON serta disinkronkan secara realtime atau waktu nyata pada semua klien yang sedang terhubung.

2.5.2.2 Firebase Storage

Firebase *storage* merupakan salah satu fitur dari firebase yang dirancang guna pengembang aplikasi yang memiliki keperluan untuk menyimpan serta menampilkan sebuah konten yang dibuat oleh pengguna. Diantaranya data seperti foto atau video dengan menambahkan keamanan google pada kegiatan unduh dan unggah data untuk aplikasi pada firebase. Meski seperti apapun kualitas jaringannya pengembang bisa menggunakannya untuk menyimpan audio, gambar, konten dan video lain yang dibuat oleh pengguna secara langsung dari sebuah firebase SDK klien.

2.5.2.3 Firebase Configuration

Untuk memakai database firebase tentunya diperlukan serangkaian tahapan yang bertujuan menghubungkan firebase pada aplikasi kita. Atau sebuah langkah untuk konfigurasi database firebase pada aplikasi kita, langkah-langkah tersebut diantaranya sebagai berikut :

1. Pertama masuk kedalam situs console.firebase.google.com

Setelah masuk, disini adalah tempat kita membuat proyek baru yang terdiri dari nama proyek dan Negara / wilayah.



2. Pilih aplikasi android dan daftarkan aplikasi kita

Pada bagian ini kita harus mendapatkan SHA 1 yang berguna untuk kunci penghubung pada aplikasi yang kita buat. Kemudian pada tahapan ini kita juga akan mendownload file json yang berfungsi untuk menggabungkan aplikasi kita dengan firebase. Kemudian lakukan *copy google-services.json* tersebut dalam direktori app pada android studio.

3. Masukan file dependencies sesuai platform firebase yang kita gunakan

Pertama masukan kode berikut kedalam build-gradle dengan classpath `"com.google.gms:google-services:3.0.0"` kemudian kita juga harus memasukan apply plugin pada app/build.gradle. Langkah terakhir adalah memasukan *dependencies* firebase pada aplikasi pada android.

2.5.3 Android Studio

Secara umum android studio merupakan sebuah aplikasi yang dibuat dengan tujuan untuk bisa membuat aplikasi yang dapat berfungsi pada perangkat smartphoone android yang memiliki ekstensi APK serta XAPK. Secara spesifik android studio adalah sebuah *Integrated Development Environment* atau biasanya disingkat dengan IDE. Dikenalkan oleh google pada acara google I/O diselenggarakan pada tahun 2013, android studio memiliki pengertian sebuah pengembangan atau evolusi dari aplikasi Eclipse serta dibuat berdasar pada IDE java yang terkenal. Android studio ialah IDE secara resmi untuk mengembangkan aplikasi dengan basis android (Komputerdia, 2019). Android studio memiliki berbagai fitur diantaranya.

- a. Penggunaan Gradle-based build system yang cukup fleksibel
- b. Dapat melakukan *build berbagai macam* APK
- c. Mendukung cetakan dengan *Google Services* serta berbagi jenis
- d. Bagian tampilan editor yang baik
- e. *Built-in support* yang diperuntukan *Google Cloud Platform*
- f. Dapat import secara langsung dari maven *repository*
- g. Mempunyai suatu emulator yang kaya fitur dan cepat
- h. Fitur instan Run yang mendukung perubahan pada suatu aplikasi yang tengah berjalan tanpa harus menciptakan aplikasi baru.

2.5.4 JAVA

Java merupakan suatu Bahasa pemrograman yang memiliki pengertian yaitu sebuah Bahasa pemrograman yang bersifat umum (*general-purpose*) yang konkuren atau suatu Bahasa yang dapat menjalankan banyak tugas dalam waktu yang bersamaan. Dengan menggunakan basis berupa *class*, memiliki orientasi berupa objek dan dirancang khusus supaya tidak memiliki sifat ketergantungan pada sistem-sistem tertentu. Java dirancang secara khusus sebagai sebuah

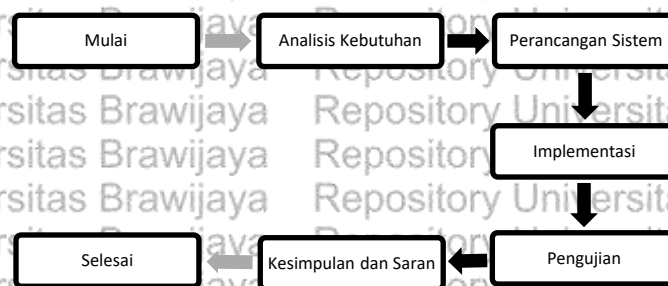


Bahasa yang umum dalam pemrograman yang dapat digunakan untuk menciptakan berbagai jenis aplikasi, seperti aplikasi website, desktop, aplikasi mobile, sampai perangkat *embedded* seperti sebuah microprocessor atau perangkat cerdas.

Pada Bahasa pemrograman java juga memiliki salah satu sifat yaitu *concurrent*, pada sebuah teori komputasi, pemrograman konkuren memiliki arti yaitu sebuah fitur dalam Bahasa pemrograman yang memiliki kemampuan untuk menjalankan banyak hal dalam waktu yang sama atau satu waktu. Pada umumnya, pemrograman konkuren dapat meningkatkan performa dari aplikasi yang dibangun. Bahasa pemrograman java mempunyai sebuah moto “*Write once, run anywhere*” yang memberikan prioritas bahwa suatu kode yang kita tulis pada Bahasa pemrograman java dapat langsung berjalan di berbagai jenis perangkat serta berbagai jenis sistem informasi tanpa harus melakukan *compile* ulang. Untuk memperoleh fitur tersebut Bahasa pemrograman java menggunakan sebuah konsep code dan JVM atau singkatan dari *Java Virtual machine* (Andre, 2019).

BAB 3 METODOLOGI

Pada bab berikut Metodologi mengandung penjelasan mengenai suatu metode/langkah penelitian yang akan digunakan di dalam proses pengembangan aplikasi pelatihan terhadap pendamping PSD di Universitas Brawijaya yang juga prosesnya terdapat pada SDLC *waterfall*. Metodologi terdiri dari bermacam tahapan yang terdapat pada penelitian ini, diantaranya yakni dimulai dari proses studi literatur, kemudian analisis kebutuhan, kemudian perancangan sistem, dilanjutkan implementasi, serta pengujian, terakhir perumusan kesimpulan dan saran, dapat dilihat lebih lanjut pada Gambar 3.1.



Gambar 3.1 Alur Metodologi Penelitian

3.1 Analisis Kebutuhan

Analisis kebutuhan memiliki fungsi untuk melakukan analisis spesifikasi atau kebutuhan yang berasal dari sistem dan kemudian akan dikembangkan (A.S & Shalahuddin, 2015). Pada penelitian berikut tahapan analisis kebutuhan dilaksanakan pada PSD UB. Tahapan dilaksanakan dengan proses elisitasi terhadap kebutuhan dengan menggunakan teknik wawancara yang bertujuan guna menghimpun data-data yang berkaitan dengan pelatihan *disability awareness*. dari wawancara yang terdapat pada lampiran A diperoleh gambaran-gambaran umum mengenai sistem dan fitur dari sistem. Setelah melakukan elisitasi kebutuhan tahapan lanjut menjalankan spesifikasi pada kebutuhan dengan memperoleh masing-masing aktor serta fitur-fitur yang bisa dijalankan oleh aktor, fitur-fitur tersebut terdiri dari kebutuhan berkategori fungsional dan juga non-fungsional yang pada sebelumnya direpresentasikan dengan tulisan yang menggambarkan kebutuhan kategori fungsional dan juga non-fungsional beserta urutannya. Tahapan terakhir dari analisis ini diimplementasikan pemodelan kebutuhan yaitu dengan Unified Modeling Language seperti menerapkan *use case diagram* beserta *use case scenario*.



3.2 Perancangan Sistem

Pada tahapan ini setelah dilakukan analisis kebutuhan, kemudian kebutuhan atau fitur yang didapatkan dari hasil elisitasi akan dimodelkan. Kebutuhan-kebutuhan tersebut didefinisikan dengan sangat dengan tujuan agar proses sebuah perancangan dari sistem dapat diimplementasikan sesuai yang pengguna butuhkan. Maka kemudian kebutuhan sistem dimodelkan dengan *Unified Modeling Language* (UML) atau suatu bahasa khusus yang dipergunakan untuk menggambarkan jalannya sistem. Proses yang terdapat pada UML tersebut antara lain terdapat dari mulai proses perancangan, *sequence diagram*, *class diagram*, perancangan komponen, perancangan antarmuka serta basis data.

3.3 Implementasi Sistem

Pada tahapan implementasi sistem ialah sebuah fase dimana pembangunan terhadap sistem yang merujuk pada hasil dari proses perancangan yang terjadi pada sistem sebelumnya. Implementasi yang dilakukan pada sistem ini dikembangkan dengan platform dengan basis mobile yakni android dengan menggunakan IDE Android Studio dengan menggunakan bahasa pemrograman JAVA. Kemudian untuk penyimpanannya menggunakan database firebase dengan menggunakan fitur *Realtime Database*.

3.4 Pengujian

Pengujian dilakukan setelah setiap fitur pada sistem telah diimplementasikan dengan menggunakan teknologi yang ada, pengujian ini dilaksanakan dengan suatu tujuan mengetahui setiap fitur dari sebuah sistem apakah sudah beroperasi atau berjalan sesuai dengan apa yang dibutuhkan. Proses pengujian yang dijalankan pada penelitian ini antara lain yaitu *black box testing* dan *white box testing* dengan menyertakan validasi dan *cyclomatic complexity*. Serta menggunakan pengujian *usability* dengan metode SUS.

3.4.1 White Box Testing

White box testing ialah suatu pengujian yang berdasarkan terhadap detail prosedur atau langkah-langkah dan urutan logika pada sebuah kode program. Pada proses *white box testing*, pemeriksa melihat kode-kode dari kumpulan program dan menjumpai *bug* dari kode program yang tengah diuji. Inti dari pengujian ini adalah melakukan pengujian yang dilakukan hingga proses yang begitu detail seperti pengecekan pada kode program. Pada pengujian ini menggunakan metode unit testing dimana pada metode ini menggunakan langkah-langkah seperti perancangan algoritma, *flowgraph*, *cyclomatic complexity*, dan jalur independent



3.4.2 Black Box Testing

Pengujian *black box testing* ialah sebuah pengujian yang berdasarkan kepada detail dari sebuah aplikasi seperti *User Interface* atau tampilan dari aplikasi. Pengujian fungsi-fungsi pada fitur yang ada pada aplikasi dengan tujuan apakah setiap fitur sesuai dengan proses bisnis yang diinginkan oleh pemesan aplikasi atau *customer*. Pada pengujian black box testing ini tidak melihat kode-kode dari program aplikasi yang sedang diuji. Pada pengujian ini setiap fitur di uji apakah sesuai dengan yang dibutuhkan customer ketika sebuah fungsi sudah sesuai dengan yang diinginkan maka statusnya menjadi valid, begitu seterusnya pengujian dilakukan. Komponen pengujiannya antara lain fungsi/kebutuhan, data input, hasil yang menjadi harapan dan hasil kenyataan yang didapat.

3.4.3 Pengujian Usability

Pengujian *usability* merupakan suatu metode pengujian yang digunakan untuk menguji kelayakan pakai dari suatu sistem dari segi penggunaan. pengujian *usability* dengan menggunakan suatu metode dengan nama SUS atau *System Usability Scale* dengan memberikan akses pada sistem yang sebelumnya telah dikembangkan untuk menjadi bahan responden. Setelah akses sistem diberikan dan telah dicoba, kemudian diberikan sebanyak sepuluh pertanyaan mengenai kemudahan dari sistem yang dibuat dan dilakukan pengukuran menggunakan skala *likert*.

3.5 Kesimpulan dan Saran

Proses ini merupakan tahapan/langkah akhir dari metodologi penelitian yang dilakukan setelah semua tahapan-tahapan sebelumnya dilakukan yaitu pengembangan perangkat lunak telah selesai dilakukan. Kesimpulan ini di dapat terutama dari proses analisis serta pengujian pad aplikasi sebelumnya yang telah dibangun. Kemudian saran ditulis dengan tujuan untuk menjalankan proses perbaikan terhadap kesalahan-kesalahan yang terdapat pada sistem dan menyempurnakan tulisan-tulisan tersebut serta memberikan suatu pertimbangan lebih lanjut mengenai pengembangan aplikasi atau perangkat lunak.

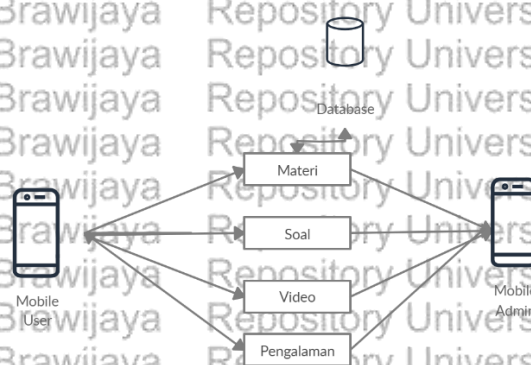


BAB 4 ANALISIS KEBUTUHAN

Analisis kebutuhan merupakan sebuah bagian awal dari pengembangan suatu sistem. Bagian ini memiliki sebuah fungsi penting yaitu untuk mengidentifikasi kebutuhan-kebutuhan yang terdapat pada aplikasi pelatihan pendamping PSLD UB. Kebutuhan tersebut didapatkan dengan menggunakan teknik wawancara. Dari hasil wawancara tersebut diperoleh data-data berupa teks yang kemudian dianalisis untuk menghasilkan kebutuhan. Dari hasil wawancara tersebut juga dianalisis untuk melakukan identifikasi aktor yang mengambil peran dalam sistem.

4.1 Deskripsi Sistem

Sistem aplikasi pelatihan pendamping PSLD UB merupakan sebuah sistem E-learning berbasis android yang bertujuan untuk memberikan sebuah pelatihan lanjutan terhadap calon pendamping di PSLD UB. Sistem ini dibangun menggunakan android sebagai basisnya, hal ini dikarenakan karena kemudahan mobilitas dan fungsionalitas pada android. Pada aplikasi ini berisi berbagai macam materi pelatihan yang dibagi menjadi beberapa kategori, kategori tersebut diantaranya tuna netra, tuna tuli, autisme dan tuna daksa. Yang setiap materinya berisi tata cara dan pembelajaran bagaimana cara mendampingi seorang mahasiswa dengan sebuah keterbatasan. Selain itu pada aplikasi ini juga memiliki sebuah fitur berbagi pengalaman yang dimana setiap pengalaman pendampingan bisa di bagikan disana. Untuk gambaran dari sistem akan digambarkan lebih detail pada Gambar 4.1.



Gambar 4.1 Gambaran umum sistem

Gambar mengenai arsitektur sistem aplikasi pelatihan pendamping diatas diantaranya terdapat beberapa komponen yaitu terdapat *database* yang menggunakan firebase kemudian pada *database* terdapat beberapa *activity* yang kemudian mengakses komponen database yang kemudian diolah oleh sistem berupa materi soal dan video yang berisi materi materi pealtihan pendampingan.



4.2 Elisitasi Kebutuhan

Elisitasi kebutuhan merupakan aktivitas yang memiliki tujuan untuk menemukan kebutuhan yang diperlukan oleh sistem dengan cara berkomunikasi pada pihak terkait. Pada topik skripsi aplikasi pelatihan pendamping ini proses elisitasi kebutuhan dilakukan dengan menggunakan teknik tradisional, teknik tradisional yang digunakan adalah wawancara yang dapat dilihat dalam Lampiran A. Jenis wawancara yang digunakan merupakan wawancara tidak terstruktur, jenis ini digunakan karena peneliti belum tau pasti tentang permasalahan-permasalahan yang terjadi pada studi kasus tersebut ataupun pemahaman yang dimiliki hanya sebatas pengetahuan umum. Kemudian kebutuhan yang telah terkumpul divalidasi menggunakan pengujian *black box* dimana pengujian ini didasarkan pada fungsi-fungsi yang ada pada aplikasi dengan melihat kesesuaian antara alur fungsi dengan bisnis proses yang diinginkan oleh customer. Wawancara pada PSLD UB dilakukan kepada coordinator disability awareness yang bertindak sebagai seorang Koordinator pelatihan *disability awareness*. kemudian dari hasil wawancara yang di dapat, didapatkan beberapa fakta-fakta yang kemudian diolah untuk menjadi sebuah kebutuhan kategori fungsional maupun kategori non-fungsional.

4.3 Identifikasi Aktor

Pada suatu bagian identifikasi aktor akan dijelaskan berbagai peranan dari aktor yang berkaitan pada pengembangan sistem. Aktor tersebut berasal dari hasil wawancara yang telah dilaksanakan pada tahapan sebelumnya. Aktor tersebut terdiri dari Pengunjung, *User* dan Admin. Setiap aktor memiliki tugas dan fungsi tersendiri. Identifikasi dari setiap aktor akan dijelaskan secara lebih lanjut pada Tabel 4.1 yang memiliki bagian tabel diantaranya nomor, aktor, nama dan berbagai penjelasan atau deskripsi mengenai berbagai macam aktor.

Tabel 4.1 Identifikasi Aktor pada Sistem

No	Nama Aktor	Deskripsi
1	Pengunjung	Pengunjung adalah seorang aktor yang dapat mengakses menu-menu aplikasi pelatihan berupa materi dan video tapi tidak untuk soal dan share pengalaman.



Tabel 4.2 Identifikasi Aktor pada Sistem (lanjutan)

No	Nama Aktor	Deskripsi
2	User	User adalah pengguna yang telah melakukan <i>login</i> pada aplikasi pelatihan pendamping, selain bisa mengakses materi dan video pelatihan, user juga dapat mengakses fitur soal dan fitur share berbagi pelatihan.
3	Admin	Admin merupakan seorang aktor yang memiliki peran untuk melakukan <i>update</i> terhadap materi yang ada pada aplikasi, admin juga memiliki peran untuk menambahkan soal-soal yang ada pada aplikasi dan admin juga bisa menghapus <i>user</i> yang terdaftar. Admin juga dapat memasukan jadwal pelatihan pendampingan.

4.4 Kebutuhan Fungsional

Kebutuhan fungsional memiliki pengertian dimana sebuah kebutuhan harus dimiliki oleh sistem dan dapat diakses oleh para pengguna. Kebutuhan ini diperoleh berdasar hasil dari elisitasi yang telah dilakukan pada tahapan sebelumnya. Kebutuhan fungsional di definisikan menggunakan kode APP_F_XX. APP adalah sebuah singkatan dari Aplikasi Pelatihan Pendamping, kemudian untuk F merupakan sebuah kode yang menunjukkan singkatan/kependekan dari Fungsional, dan untuk XX ialah suatu kode untuk urutan dari kebutuhan kategori fungsional. Dan kebutuhan fungsional kemudian dijelaskan lebih lanjut pada Tabel 4.3 sebagai berikut.

Tabel 4.3 kebutuhan Fungsional

No	Kode	Use Case	Deskripsi Kebutuhan
1	APP_F_01	<i>Login</i>	Sistem dapat menyediakan fitur <i>login</i> untuk pengunjung sehingga bisa melakukan akses pada sistem sebagai seorang <i>user</i> ataupun seorang admin.
2	APP_F_02	<i>Register</i>	Merupakan kebutuhan sistem yang memiliki fungsi untuk melakukan pendaftaran terhadap aplikasi pelatihan adapun beberapa data yang di isi berupa, <i>username</i> , <i>password</i> , <i>email</i> , <i>fullname</i> , <i>NIM</i> dan <i>faculty</i> .



Tabel 4.4 kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Deskripsi Kebutuhan
3	APP_F_03	Login	Sistem dapat menyediakan fitur <i>login</i> untuk pengunjung sehingga bisa melakukan akses pada sistem sebagai seorang <i>user</i> ataupun seorang admin.
4	APP_F_04	Logout	Sistem memiliki fitur yang digunakan untuk keluar dari aplikasi yang telah melakukan <i>Login</i> sebelumnya.
5	APP_F_05	Lihat materi	Sistem memiliki fitur untuk melihat materi pelatihan yang telah ditambahkan sebelumnya.
6	APP_F_06	Kerjakan soal	Sistem dapat menyediakan fitur untuk mengerjakan soal-soal yang berkaitan dengan disabilitas.
7	APP_F_07	Tambah soal	Sistem dapat menyediakan fitur bagi admin untuk melakukan tambah atau menghapus soal.
8	APP_F_08	Hapus user	Sistem dapat menyediakan fitur yang berguna untuk menghapus <i>user</i> yang telah terdaftar.
9	APP_F_09	Lihat user	Sistem menyediakan sebuah fitur yang berguna bagi admin untuk melihat daftar <i>user</i> yang terdaftar dalam aplikasi.



10	APP_F_10	Hapus soal	Sistem dapat menyediakan fitur bagi admin untuk menghapus soal yang telah ditambahkan.
----	----------	------------	--

Tabel 4.5 kebutuhan Fungsional (lanjutan 2)

No	Kode	Use Case	Deskripsi Kebutuhan
11	APP_F_11	Penjadwalan pelatihan	Sistem dapat menyediakan fitur untuk admin menjadwalkan pelatihan Bahasa isyarat bagi <i>user</i> yang telah terdaftar.
12	APP_F_12	Hapus soal	Sistem dapat menyediakan fungsi bagi admin untuk melakukan hapus pada soal yang telah ditambahkan.
13	APP_F_13	Tambah pengalaman	Sistem dapat menyediakan sebuah fungsi yang berguna bagi <i>user</i> untuk membagikan pengalaman
14	APP_F_14	Lihat pengalaman	Sistem menyediakan suatu fitur yang berfungsi untuk <i>user</i> melihat berbagai pengalaman yang telah ditambahkan oleh <i>user</i> lain
15	APP_F_15	Komentar	Bagi <i>user</i> sistem juga menyediakan sebuah fungsi untuk bisa menambahkan komentar pada setiap pengalaman yang ditambahkan.
16	APP_F_16	Edit profil	Sistem menyediakan fungsi bagi <i>user</i> untuk melakukan edit terhadap data profil.
17	APP_F_17	Lihat video	<i>User</i> dapat melihat berbagai video pelatihan-pelatihan pendampingan yang telah disediakan oleh sistem.



18	APP_F_18	Lupa <i>password</i>	Sistem dapat menyediakan fitur yang berguna untuk memberitahukan <i>password</i> apabila <i>user</i> lupa <i>password</i> .
----	----------	----------------------	---

Tabel 4.6 kebutuhan Fungsional (lanjutan 3)

No	Kode	Use Case	Deskripsi Kebutuhan
19	APP_F_19	Edit materi	Sistem dapat menyediakan fitur untuk admin yang berfungsi sebagai melakukan pembaruan soal apa bila ditemukan kesalahan
20	APP_F_20	Hapus materi	Sistem dapat menyediakan fungsi bagi admin untuk menghapus materi yang tidak diperlukan

4.5 Kebutuhan Non Fungsional

Kebutuhan ini ialah suatu kebutuhan yang memiliki hubungan dengan suatu batasan serta kualitas dari sistem. Kemudian Kebutuhan kategori non fungsional ini terdiri dari beberapa kategori yang diantaranya *availability*, *usability*, *performance*, serta *reliability*. Kode yang digunakan pada kebutuhan ini ialah APP-NF-XX. APP memiliki pengertian singkatan dari Aplikasi Pelatihan Pendamping sementara untuk NF merupakan sebuah kode kependekan dari kebutuhan Non Fungsional. Dan untuk XX merupakan nomor dari kebutuhan Non Fungsional. kebutuhan non fungsional akan dijabarkan pada Tabel 4.7.

Tabel 4.7 Kebutuhan Non Fungsional

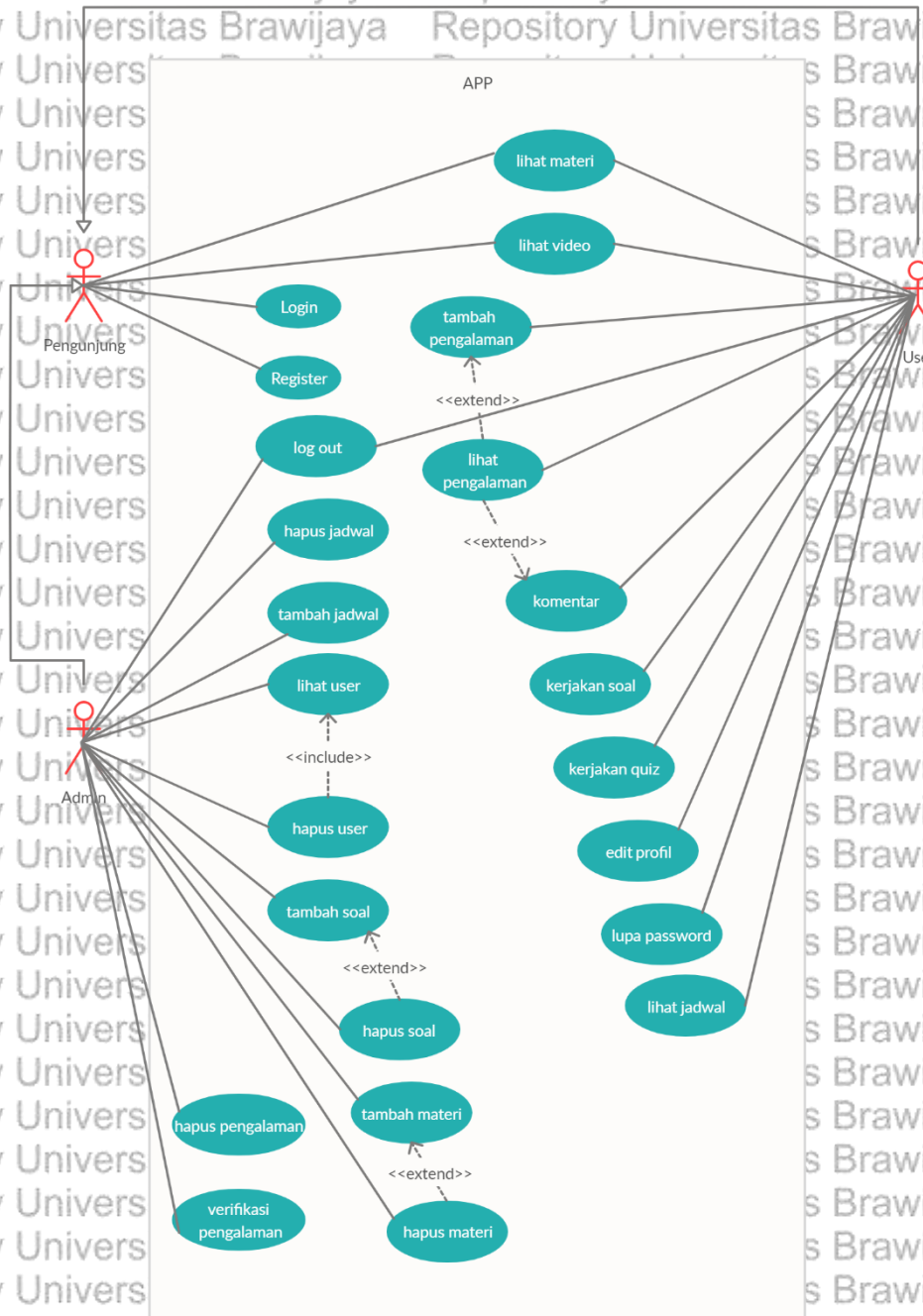
No	Kode	Parameter	Deskripsi Kebutuhan
1	APP_NF_01	Usability	Sistem dapat memberikan kemudahan penggunaan pada setiap segmen.

4.6 Peemodelan Kebutuhan

4.6.1 Use Case Diagram

Use case ialah sebuah diagram yang memiliki fungsi atau tujuan untuk melakukan pemodelan terhadap perilaku aktor pada sistem. Kemudian pada diagram ini juga mengandung gambaran apapun yang bisa dijalankan aktor pada satu sistem yang berkaitan dengannya. Pada sistem Aplikasi Pelatihan Pendamping terdapat 3 aktor yang diantaranya terdiri dari Pengunjung, *user* serta terdapat admin. Dari ketiga aktor tersebut memiliki 23 kebutuhan

berkategori fungsional serta 1 kebutuhan kategori non fungsional. Tujuan dari diagram ini agar pengguna dapat melihat gambaran mengenai fungsionalitas dari sistem yang dibangun. Kemudian akan bisa dilihat lebih lanjut pada Gambar 4.1.



Gambar 4.2 Use Case Diagram APP



4.6.2 Use Case Scenario

Use case scenario merupakan *use case* yang berfungsi memberikan sebuah penjelasan terhadap alur dari kegiatan ketika seorang pengguna menjalankan atau menggunakan sebuah sistem. merupakan *use case* yang berbentuk sebuah tabel. Tabel tersebut memiliki beberapa komponen diantaranya *main flow*, *pre-condition*, *alternative flow*, *Actor*, *objective* serta yang terakhir *post-condition*. Pada Tabel 4.8 sampai 4.26 akan dijelaskan lebih lanjut mengenai pemodelan ini.

Tabel 4.8 Use Case Scenario Login

Actor	Pengunjung
Objective	Aktor berhasil melakukan <i>login</i> sehingga memiliki hak akses yang dikenali sistem
Pre-Condition	<i>Activity login</i> ditampilkan
Main Flow	1. Aktor mengisi <i>username</i> dan <i>password</i> dengan lengkap 2. Aktor menekan tombol <i>login</i> pada <i>Activity</i>. 3. Sistem memeriksa <i>username</i> serta <i>password</i>. 4. Sistem menampilkan <i>Activity</i> utama.
Alternative Flow	1. Jika <i>username</i> yang dimasukan salah maka akan menampilkan Toast "<i>Username salah</i>" 2. Jika <i>Password</i> salah maka akan menampilkan Toast "<i>Password yang anda masukan salah</i>"
Post-Condition	Sistem menampilkan <i>Activity</i> utama.

Aktivitas login ini difungsikan sebagai portal utama dalam suatu sistem untuk mengakses fitur-fitur utama yang tersedia, pada skenario ini terdapat tahapan awal yaitu kondisi aktivitas fungsi login ditampilkan kemudian pengguna mengisi data-data terkait yang diperlukan untuk tahapan verifikasi sehingga memberikan akses pada fitur utama. Ketika pengguna menuliskan nama yang tidak benar maka sistem kemudian memunculkan peringatan yakni nama yang anda masukan salah, dan ketika memasukan kata kunci atau *password* yang salah maka akan menampilkan sebuah peringatan melalui fungsi toast kata kunci yang anda masukan salah. Dan ketika pengguna menekan tombol untuk *login* tetapi kolom dalam keadaan kosong kemudian sistem akan otomatis memunculkan peringatan form tidak boleh keadaan kosong baik ketika yang tidak diisi kolom *username* dan kolom *password*. Dan pada akhirnya pengguna yang telah *login* berhasil untuk masuk kedalam sistem sehingga dapat mengakses berbagai fitur utama.



Tabel 4.9 Use Case Scenario Register

Actor	Pengunjung.
Objective	Aktor melakukan <i>Register</i> sebagai <i>user</i> .
Pre-Condition	<i>Activity start</i> ditampilkan.
Main Flow	1. Aktor menekan "<i>Create New Account</i>". 2. Sitem menampilkan <i>Activity Register</i>. 3. Aktor mengisi <i>Username</i>, <i>Email</i> serta <i>Password</i>. 4. Aktor menekan tombol <i>Continue</i>. 5. Sistem menampilkan <i>Activity Register</i> kedua. 6. Aktor mengisi kolom <i>fullname</i>, <i>nim</i> serta <i>faculty</i>. 7. Aktor mengunggah foto profil. 8. Aktor menekan tombol <i>Register</i>.
Alternative Flow	1. Jika terdapat kolom yang tidak di isi maka akan muncul Toast kolom tidak boleh kosong. 2. Jika <i>username</i> sudah ada maka sistem menampilkan pesan "<i>username sudah ada</i>".
Post-Condition	Aktor berhasil mendaftar sebagai <i>user</i> .

Tabel 4.10 Use Case Scenario lihat materi

Actor	Pengunjung, <i>user</i>
Objective	Aktor membuka materi yang ingin dipelajari.
Pre-Condition	<i>Activity</i> materi ditampilkan
Main Flow	1. Aktor menekan tombol yang berisi materi. 2. Sistem menampilkan beberapa kategori materi. 3. Aktor memilih materi yang ingin dipelajari.
Alternative Flow	
Post-Condition	Aktor berhasil membuka materi yang ingin dipelajari.

Pada *scenario register* dimasukkan untuk para pengguna awal untuk mendaftar kepada sistem agar bisa untuk melakukan operasi *login* sehingga dapat mengakses fungsi utama. Kondisi awal yang dilakukan ialah pengguna masuk kedalam halaman *register* dan mengisi data-data yang menjadi syarat pendaftaran untuk mengakses kedalam sistem. Dan hasil yang diharapkan pengunjung telah berhasil terdaftar sebagai pengguna. Kemudian *scenario* lihat



materi berfungsi sebagai skema pagi pengguna untuk melihat materi-materi yang telah disediakan sistem untuk pelatihan lebih lanjut mengenai tatacara pendampingan. Pada kondisi akhir aktor dapat membuka seluruh materi yang ada.

Tabel 4.11 Use Case Scenario lihat video

Actor	Pengunjung, <i>user</i>
Objective	Aktor membuka video pelatihan
Pre-Condition	Activity video ditampilkan.
Main Flow	1. Aktor menekan tombol yang berisi video. 2. Sistem menampilkan beberapa video pelatihan. 3. Aktor memutar video yang ingin dipelajari.
Alternative Flow	-
Post-Condition	Aktor berhasil memutar video yang ingin dipelajari.

Tabel 4.12 Use Case Scenario lihat pengalaman

Actor	<i>User</i>
Objective	Aktor melihat pengalaman yang dibagikan
Pre-Condition	Activity pengalaman ditampilkan.
Main Flow	1. Aktor menekan tombol yang berisi pengalaman. 2. Sistem menampilkan beberapa list pengalaman. 3. Aktor memilih pengalaman yang ingin dipelajari.
Alternative Flow	-
Post-Condition	Aktor berhasil melihat pengalaman yang dibagikan.

Pada scenario lihat video dimaksudkan sebagai sebuah aktivitas untuk melihat berbagai macam video yang mendukung proses pelatihan dimana prosesnya diawali dari pengguna berada pada menu utama yang kemudian pengguna melakukan klik pada video sehingga sistem menampilkan video untuk dapat dilihat. Kemudian pada lihat pengalaman dimaksudkan agar para pengguna dapat berbagi mengenai pengalaman pendampingan yang ia lalui setiap harinya, sehingga dapat menjadi sebuah referensi atau acuan bagi pendamping baru yang masi kurang dari segi pengalaman, di sayaratkan kondisi pertama pengguna bertempat pada menu utama dan melakukan klik pada menu pengalaman sehingga sistem menampilkan pengalaman.



Tabel 4.13 Use Case Scenario tambah pengalaman

Actor	User
Objective	Aktor menambahkan pengalaman
Pre-Condition	Activity pengalaman ditampilkan.
Main Flow	1. Aktor mengisi kolom judul dan isi pengalaman. 2. Aktor menekan tombol share. 3. Admin melakukan verifikasi pengalaman. 4. Pengalaman berhasil ditambah.
Alternative Flow	1. Muncul Toast judul/ pengalaman tidak boleh kosong apabila tombol share ditekan.
Post-Condition	Aktor berhasil menambahkan pengalaman

Tabel 4.14 Use Case Scenario komentar

Actor	User
Objective	Aktor menambahkan komentar
Pre-Condition	Activity pengalaman ditampilkan.
Main Flow	1. Aktor membuka salah satu pengalaman. 2. Aktor mengisi kolom komentar. 3. Aktor menekan tombol kirim. 4. Sistem menyimpan komentar.
Alternative Flow	1. Muncul Toast yang memunculkan kalimat "teks tidak boleh kosong" apabila ditekan tombol kirim
Post-Condition	Aktor berhasil menambahkan komentar.

Aktivitas scenario menambah pengalaman dimaksudkan agar para pengguna yang telah melalui aktivitas pendampingannya dapat membagikan ilmu-ilmu di lapangan terkait mahasiswa difabel. Dimana pengguna akan mengisi data seperti judul dan isi dari pengalaman yang dimilikinya. Disyaratkan pengguna berada pada menu pengalaman dan menekan tombol tambah kemudian data disimpan. Untuk aktivitas komentar berada pada detail pengalaman yang bertujuan menambahkan komentar terhadap pengalaman yang telah dibagikan.



Tabel 4.15 Use Case Scenario kerjakan soal

Actor	User
Objective	Aktor mengerjakan soal
Pre-Condition	Activity utama ditampilkan.
Main Flow	1. Aktor membuka menu soal. 2. Aktor mengerjakan soal-soal pilihan ganda. 3. Aktor menekan tombol selesai.
Alternative Flow	-
Post-Condition	Aktor berhasil mengerjakan soal.

Tabel 4.16 Use Case Scenario lupa password

Actor	User
Objective	Aktor mengetahui password yang lupa
Pre-Condition	Activity login ditampilkan.
Main Flow	1. Aktor membuka menu lupa password. 2. Aktor mengisi email dan username. 3. Sistem mengirimkan password ke email.
Alternative Flow	-
Post-Condition	Aktor berhasil login ulang.

Tabel 4.17 Use Case Scenario edit profil

Actor	User
Objective	Aktor melakukan edit profil
Pre-Condition	Activity utama ditampilkan.
Main Flow	1. Aktor melakukan klik terhadap foto profil. 2. Sistem menampilkan gambar profil. 3. Aktor menekan tombol edit profil. 4. Aktor mengisi kolom yang akan diubah. 5. Aktor menekan tombol simpan. 6. Sistem menyimpan perubahan.
Alternative Flow	-
Post-Condition	Aktor berhasil merubah profil.



Pada skenario mengerjakan soal dimaksudkan agar pengguna dapat mengerjakan beberapa soal dengan kategori yang bertujuan untuk mengasah kemampuan mengenai pengetahuan yang berkaitan dengan mahasiswa difabel. Kemudian pada kondisi awal pengguna harus berada pada halaman utama soal agar dapat mengerjakan soal. Untuk lupa *password* merupakan skema dari sebuah fungsi yang bertujuan ketika seorang pengguna mendadak lupa terhadap *password* pengguna bisa meminta sistem untuk mengirimkan sandi tersebut, dengan kondisi awal pengguna harus berada pada halaman lupa *password* kemudian sistem akan mengirimkannya melalui email yang telah terdaftar sebelumnya. Terakhir untuk edit profil dimaksudkan agar pengguna dapat memperbarui profil ketika terdapat kesalahan atau perubahan. Dimana kondisi awal pengguna harus berada dalam halaman edit profil dari sistem.

Tabel 4.18 Use Case Scenario lihat jadwal pelatihan

Actor	User
Objective	Aktor melihat jadwal pelatihan
Pre-Condition	Activity utama ditampilkan.
Main Flow	1. Aktor menekan tombol lihat jadwal. 2. Sistem menampilkan jadwal pelatihan. 3. Aktor melakukan klik pada jadwal. 4. Sistem menampilkan detail pada jadwal
Alternative Flow	-
Post-Condition	Aktor berhasil melihat jadwal.

Tabel 4.19 Use Case Scenario tambah jadwal

Actor	Admin
Objective	Aktor menambahkan jadwal
Pre-Condition	Activity utama admin ditampilkan.
Main Flow	1. Aktor memilih menu tambahkan jadwal. 2. Aktor mengisikan waktu, tempat dan tutor pelatihan. 3. Aktor menekan tombol tambah. 4. Sistem menyimpan jadwal.
Alternative Flow	-
Post-Condition	Aktor berhasil menambahkan jadwal pelatihan.



Tabel 4.20 Use Case Scenario lihat user

Actor	Admin
Objective	Aktor melihat user
Pre-Condition	Activity utama admin ditampilkan.
Main Flow	1. Aktor memilih menu user. 2. Sistem menampilkan user.
Alternative Flow	-
Post-Condition	Aktor berhasil melihat user yang terdaftar.

Tabel 4.21 Use Case Scenario hapus user

Actor	Admin
Objective	Aktor melihat user
Pre-Condition	Activity utama admin ditampilkan.
Main Flow	1. Aktor memilih menu user. 2. Sistem menampilkan user. 3. Aktor menekan tombol hapus pada user. 4. Sistem menghapus user.
Alternative Flow	-
Post-Condition	Aktor berhasil menghapus user yang diinginkan.

Pada tambah jadwal dan lihat jadwal merupakan fungsi dimaksudkan agar pengguna dapat melihat jadwal pelatihan yang diadakan oleh PSKD kemudian untuk fitur menambah jadwal dimaksudkan untuk *admin* untuk menambahkan jadwal pelatihan dengan mengisi data-data yang ada seperti nama tutor nama dan waktu. Kemudian pada akhirnya kondisi akhir pengguna berhasil untuk menambahkan ataupun melihat jadwal. Untuk fitur melihat *user* dan hapus *user* dimaksudkan untuk admin mengelola *user* yang telah terdaftar dimana admin dapat melakukan lihat komponen-komponen data dari pengguna tersebut dan admin juga dapat melakukan operasi hapus dimana kondisi awal harus masuk kedalam sistem sebagai *admin* dan masuk kedalam halaman profil memilih tombol user dan sistem akan menampilkan daftar user dalam sebuah bentuk list dan dimana dalam list tersebut terdapat tombol untuk menghapus user yang diinginkan untuk dihapus. Sementara untuk fitur lihat jadwal pengguna hanya masuk sebagai pengguna biasa yang kemudian memilih menu jadwal pada menu utama.



Tabel 4.22 Use Case Scenario tambah soal

Actor	Admin
Objective	Aktor dapat menambahkan soal
Pre-Condition	Activity utama admin ditampilkan.
Main Flow	1. Aktor memilih menu tambah soal. 2. Sistem menampilkan <i>activity</i> tambah soal. 3. Aktor mengisi kolom soal dan jawaban. 4. Aktor menekan tombol tambah.
Alternative Flow	1. Sistem menampilkan toast “kolom tidak boleh kosong” apa bila ada kolom yang belum di isi.
Post-Condition	Sistem berhasil menambahkan soal.

Tabel 4.23 Use Case Scenario hapus soal

Actor	Admin
Objective	Aktor dapat menghapus soal
Pre-Condition	Activity utama admin ditampilkan.
Main Flow	1. Aktor memilih menu soal. 2. Sistem menampilkan kumpulan soal. 3. Aktor memilih tombol hapus pada soal
Alternative Flow	-
Post-Condition	Aktor berhasil menghapus soal.

Skenario menambah soal diperuntukan bagi admin dimana admin tersebut bisa menambahkan soal dengan jawabannya dan juga dapat memilih kategori dari soal tersebut, untuk fitur ini kondisi awalnya harus masuk sebagai admin dan masuk pada halaman utama soal kemudian tinggal pilih tombol tambah dan kemudian tinggal mengisi soal. Untuk fitur hapus soal kondisi awal harus masuk kedalam sistem sebagai admin dan membuka soal yang ingin dilakukan hapus. Dan soal dapat dipilih dari berbagai kategori yang ada. Dari proses tersebut diharapkan pada hasil akhir admin dapat untuk menambahkan soal sesuai dengan kategori dan bisa juga untuk menghapus soal yang telah ditambahkan sesuai dengan kategori



Tabel 4.24 Use Case Scenario tambah materi

Actor	Admin
Objective	Aktor dapat menambahkan materi
Pre-Condition	Activity utama admin ditampilkan.
Main Flow	1. Aktor memilih menu materi. 2. Sistem menampilkan <i>activity</i> materi. 3. Aktor memilih tombol tambah materi. 4. Sistem menampilkan kolom tambah materi. 5. Aktor mengisi kolom dengan materi. 6. Aktor menekan tombol simpan. 7. Sistem menyimpan materi.
Alternative Flow	1. Menampilkan toast “kolom tidak boleh kosong” apabila ada kolom yang tidak terisi.
Post-Condition	Aktor berhasil menambahkan materi.

Tabel 4.25 Use Case Scenario hapus pengalaman

Actor	Admin
Objective	Aktor dapat menghapus pengalaman
Pre-Condition	<i>login</i> sebagai admin dan <i>activity</i> pengalaman ditampilkan.
Main Flow	1. Aktor memilih menu pengalaman. 2. Sistem menampilkan beberapa pengalaman. 3. Aktor memilih tombol hapus. 4. Sistem menghapus pengalaman.
Alternative Flow	
Post-Condition	aktor berhasil menghapus pengalaman

Fitur tambah materi dimaksudkan untuk penambahan materi yang hanya bisa dilakukan oleh seorang admin yang telah masuk kedalam sistem. Fitur ini akan berisi penambahan foto judul materi dan isi dari materi kemudian admin tinggal menekan tombol simpan. Untuk hapus pengalaman juga hanya bisa dilakukan oleh admin yang telah masuk kedalam sistem dengan cara memilih menu pengalaman dan menghapusnya berdasar pengalaman yang ia pilih.



Tabel 4.26 Use Case Scenario hapus materi

Actor	Admin
Objective	Aktor dapat menghapus materi
Pre-Condition	Activity utama admin ditampilkan.
Main Flow	1. Aktor memilih menu materi. 2. Sistem menampilkan kumpulan materi 3. Aktor memilih tombol hapus pada materi 4. Sistem menghapus materi.
Alternative Flow	-
Post-Condition	Aktor berhasil menghapus materi.

Tabel 4.27 Use Case Scenario verifikasi pengalaman

Actor	Admin
Objective	Aktor dapat memverifikasi pengalaman
Pre-Condition	Login sebagai admin dan menampilkan activity pengalaman
Main Flow	1. Aktor memilih menu pengalaman. 2. Sistem menampilkan kumpulan pengalaman. 3. Aktor memilih tombol verifikasi pengalamani 4. Sistem menampilkan pengalaman di sisi user.
Alternative Flow	-
Post-Condition	Aktor berhasil memverifikasi materi.

Pada skenario hapus materi merupakan suatu fitur yang dimaksudkan untuk admin yang berfungsi untuk menghapus materi yang sebelumnya ditambahkan atau materi ini bersifat materi tambahan. Kondisi ini berjalan dengan syarat pengunjung telah masuk kedalam sistem sebagai seorang admin, kemudian masuk kedalam aktivitas materi dan klik tombol materi tambahan. Kemudian terdapat list materi dan tinggal pilih materi yang diinginkan untuk dihapus. Dimana hasil yang diharapkan materi telah berhasil dihapus. Kemudian untuk skema verifikasi pengalaman dimaksudkan untuk melakukan verifikasi terhadap pengalaman yang telah berhasil ditambahkan dengan tujuan agar setiap pngalaman adalah pengalaman yang layak untuk dibagikan. Dengan kondisi awal yang diharapkan yakni telah memasuki sistem sebagai admin dan kemudian masuk pada menu pengalaman kemudian melakukan klik tombol verifikasi untuk pengalaman, dan pengalaman telah terverifikasi.



Tabel 4.28 Use Case Scenario hapus jadwal

Actor	Admin
Objective	Aktor dapat menghapus jadwal.
Pre-Condition	Login sebagai admin dan menampilkan <i>activity</i> jadwal
Main Flow	1. Aktor memilih tombol jadwal. 2. Sistem menampilkan jadwal. 3. Aktor memilih tombol hapusi jadwal. 4. Sistem menghapus jadwal.
Alternative Flow	-
Post-Condition	Aktor berhasil menghapus jadwal.

Skema menghapus jadwal dimaksudkan agar pengguna yang terdaftar sebagai admin dapat menghapus jadwal yang telah ditambahkan sebelumnya dimana kondisi awal yang harus dipenuhi pengguna memasuki halaman jadwal dan memilih jadwal yang ingin dihapus dengan memilih tombol hapus pada jadwal.

Tabel 4.29 Use Case Scenario kerjakan quiz

Actor	User
Objective	Aktor dapat mengerjakan <i>quiz</i> .
Pre-Condition	User telah <i>login</i> dan pada <i>activity</i> soal
Main Flow	1. Aktor memilih menu <i>quiz</i>. 2. Sistem menampilkan <i>activity quiz</i>. 3. Aktor memilih tombol <i>start</i>. 4. Sistem menampilkan soal.
Alternative Flow	
Post-Condition	Aktor dapat mengerjakan <i>quiz</i> .

BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Sistem

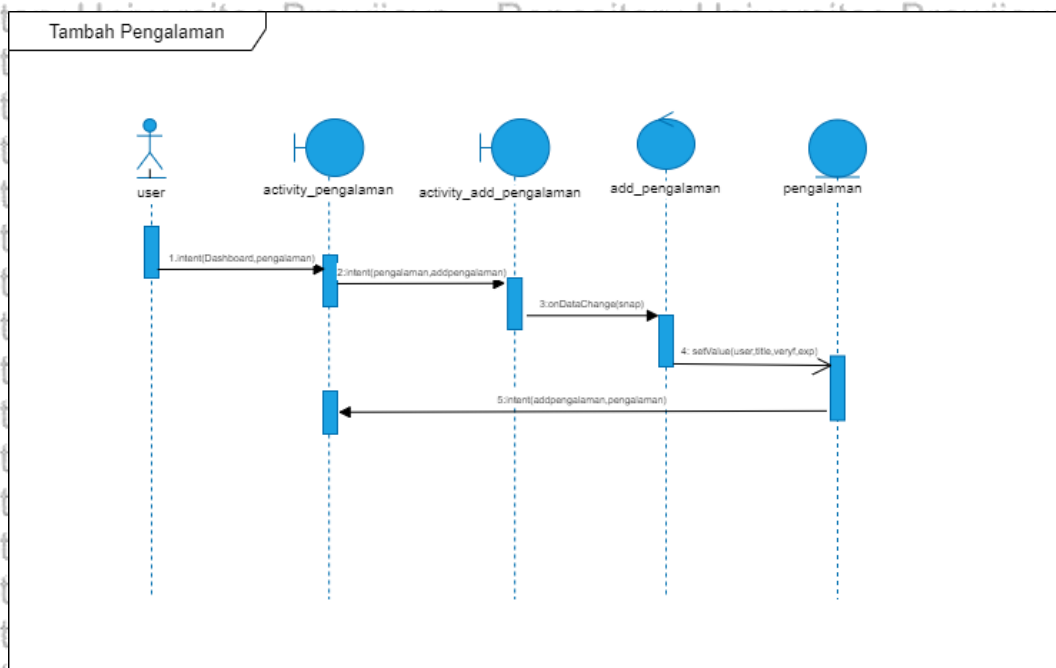
Pada bab perancangan serta implementasi ini akan dilakukan sebuah perancangan yang berkaitan dengan aplikasi pelatihan pendamping PSKD Universitas Brawijaya. Perancangan ini bertujuan untuk memberikan sebuah gambaran mengenai fitur-fitur apa saja yang dimiliki sistem nantinya dan kemudian menjadikannya sebuah acuan untuk membangun sistem secara matang dan meminimalisir adanya kesalahan atau *error* pada sistem. Pada tahapan sebelumnya atau pada bab sebelumnya, telah dilakukan analisis kebutuhan pada sistem terkait yang bisa dijadikan dasar sebagai perancangan sebuah sistem, yang melalui berbagai tahapan diantaranya terdiri dari perancangan pemodelan *sequence* diagram, perancangan komponen, perancangan suatu basis data, beserta perancangan tampilan/ antarmuka.

5.1.1 Perancangan *Sequence* Diagram

Pada diagram ini menjelaskan sebuah arah antara objek yang dilakukan antara aktor satu dengan yang lainnya yang saling berinteraksi dengan tujuan tertentu. Pada diagram ini, didalamnya memiliki objek yang telah didapatkan dari proses identifikasi yang sebelumnya terdapat pada *use case* dan proses spesifikasi kebutuhan. selanjutnya pada *sequence* kali ini akan menjelaskan tiga macam yang akan dijadikan sebagai sample.

5.1.1.1 *Sequence* Diagram Tambah Pengalaman

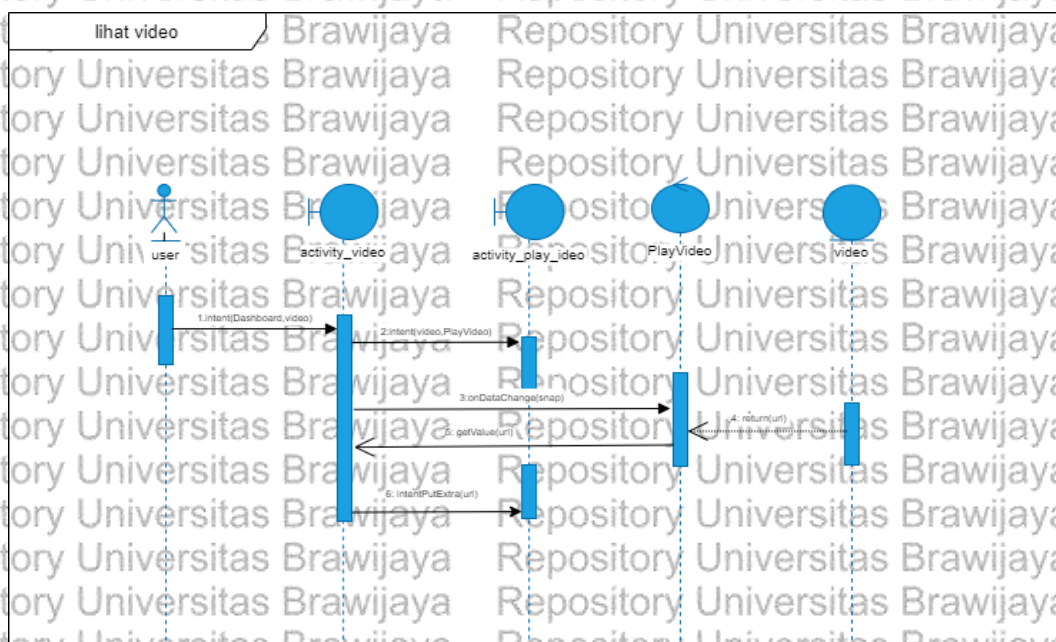
Sequence ini dijelaskan pada sebuah gambar dibawah yaitu Gambar 5.1, tambah pengalaman memiliki fungsi yaitu menyediakan media bagi para pengguna untuk menambahkan pengalaman yang dia miliki agar bisa berbagi ilmu seputar mendampingi mahasiswa difabel dalam kegiatan belajar mengajar sehingga pengguna baru memiliki gambaran tentang bagaimana mendampingi mahasiswa tersebut. *sequence* ini dimulai dari *user* yang memanggil *method* intent yang mengarahkan *user* dari halaman dashboard ke halaman *activity* pengalaman. Kemudian dari pengalaman *user* akan berpindah halaman dengan *method* intent ke halaman *add_pengalaman*. Pada halaman ini kemudian berinteraksi dengan *database* firebase dengan *method* *addListenerSingle* kemudian dari *method* ini kemudian mengirimkan data pada *database* dengan *setValue*, dengan data yang dikirimkan berupa judul memakai tipe data string, isi menggunakan tipe data string, penulis memakai tipe data string dan kemudian *verify* dengan tipe data string, data ini sebagai indikator verifikasi antara *user* dan *admin*.



Gambar 5.1 sequence diagram Tambah pengalaman

5.1.1.2 Sequence Diagram Lihat Video

Pada sequence Gambar 5.2 ini dijelaskan bagaimana alur komunikasi objek yang secara dinamis mengenai proses melihat video, dimulai ketika user berpindah dari halaman dashboard menuju halaman video dengan menggunakan intent, pada halaman tersebut akan ada daftar video yang user ingin putar, ketika video di klik maka user akan diarahkan pada halaman PlayVideo yang kemudian video diputar dengan link dari database yang diambil dari method getValue.

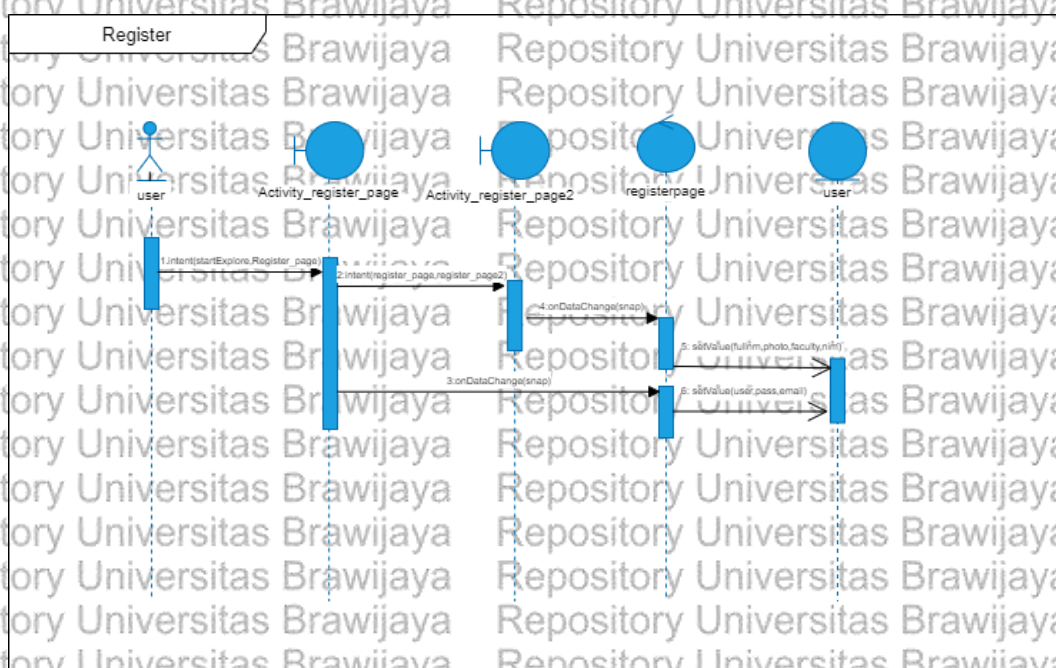


Gambar 5.2 sequence diagram lihat video



5.1.1.3 Sequence Diagram Register

pada *sequence diagram* Gambar 5.3 selanjutnya akan dijelaskan alur dari *register*, *register* memiliki fungsi untuk melakukan pendaftaran pagi pengguna aplikasi. Alur dimulai ketika *user* berada pada halaman *startExplore* kemudian *user* diarahkan pada halaman *register_page* dengan menggunakan *intent*, pada halaman *register_page* ini kemudian *user* diminta untuk mengisi data-data berupa *username, email, password* kemudian data tersebut akan disimpan pada tabel *user* dengan menggunakan *method setValue*. Ketika data tersebut sudah disimpan, *user* kemudian diarahkan lagi pada halaman *register_page2* dimana pada halaman ini *user* juga mengisi data-data tambahan yang berupa nama lengkap, *nim*, foto profil dan fakultas kemudian data tersebut akan disimpan kembali pada tabel *user* dengan menggunakan *method setValue*.



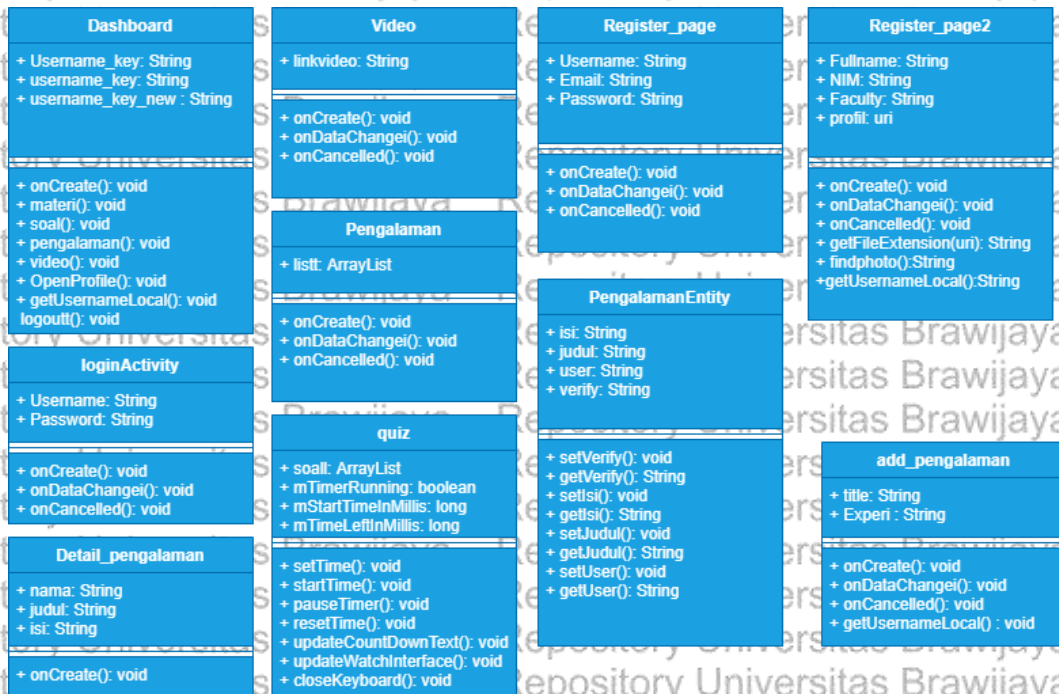
Gambar 5.3 *sequence diagram register*

5.1.2 Perancangan Class Diagram

Perancangan ini merupakan sebuah perancang yang berfungsi untuk melakukan sebuah identifikasi mengenai komponen-komponen *class diagram* diantaranya *class, object*, dan interaksi-interaksi pada sistem terkait. *Class* tersebut diperoleh dari proses perancangan sebelumnya yaitu *sequence diagram* melalui objek-objek yang selanjutnya dijadikan sebuah *class* terkait. Terdapat sebuah gambar yang menjelaskan tentang perancangan umum dan melihat relasi-relasi atau hubungan antar *class*. Dan kemudian pada perancangan secara umum akan dilakukan generalisasi pada setiap *class diagram*. Kemudian perancangan kepada *class diagram* yang bersifat secara umum akan dijelaskan pada sebuah Gambar 5.4, dan kemudian perancangan *class diagram* secara detail akan dijelaskan pada Gambar 5.5.



Gambar 5.4 Perancangan Umum Class Diagram

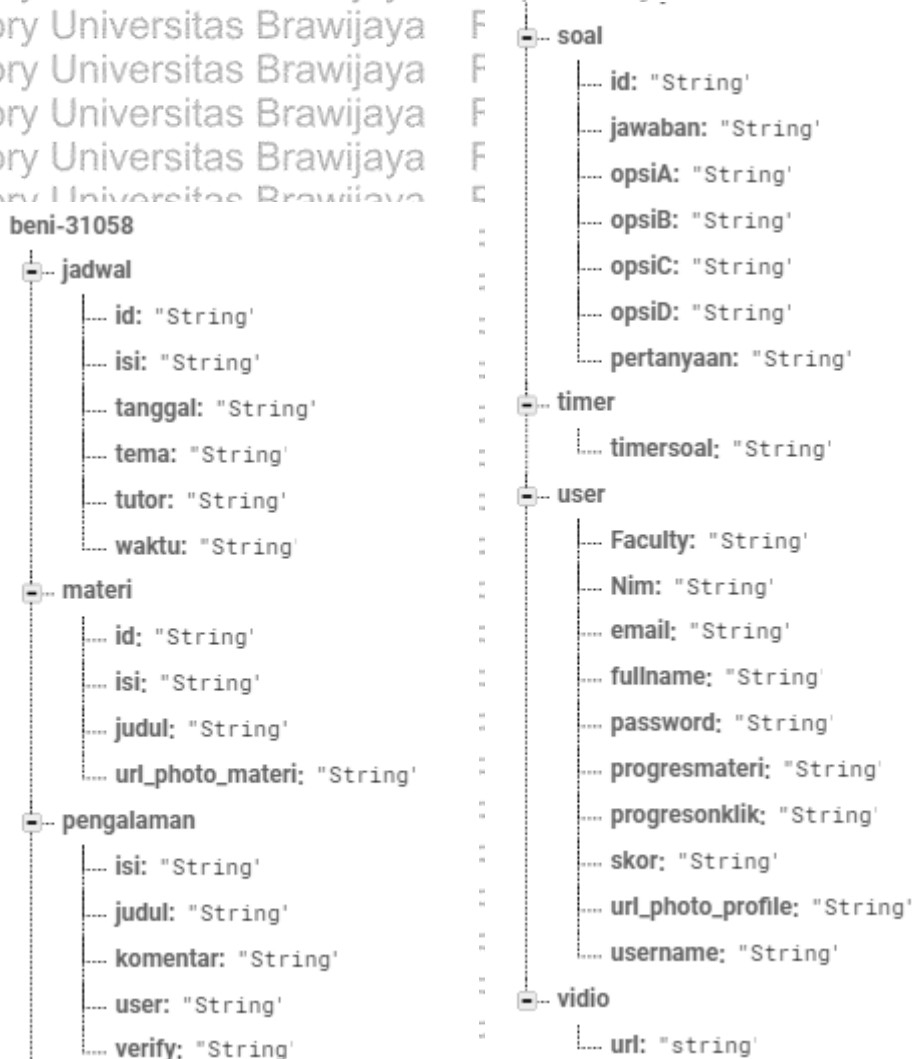


Gambar 5.5 Perancangan Spesifik Class Diagram



5.1.3 Perancangan Basis Data

Berikut merupakan merupakan rancangan Database Aplikasi Pelatihan pendamping dengan menggunakan *Firebase* pada *database* ini setiap kebutuhan diidentifikasi menggunakan paradigma *child* bertingkat. Pada aplikasi ini terdapat beberapa *child* yaitu soal yang dibawahnya terdapat beberapa *child* lagi antara lain jawaban,soal,opsiA, opsiB, opsiC, opsiD. Kemudian *user* diantaranya ada *Faculty*,*Nim*,*email*,*fullname*,*password*,*url_photo_profil*,*username*,kemudian *jadwal* terdapat isi dan waktu, kemudian *materi* terdapat judul dan materi,kemudian *pengalaman* terdapat isi,judul,komentar,penulis dan *verify*.selengkapnya dalam Gambar 5.6.





Gambar 5.6 Perancangan Database APP

5.1.4 Perancangan Komponen

Pada perancangan kali ini merupakan sebuah perancangan komponen yang didalamnya terdapat sistem yang terdiri dari rician subsistem yang akan dibuat gambarnya. Selanjutnya pada langkah rancangan ini akan dijelaskan tiga class yang terdiri dari baris code yang menyusun fitur diantaranya lihat video, tambah pengalaman dan register.

5.1.4.1 Algoritme class register

Pada *class register* memiliki fungsi sebagai suatu fitur untuk melakukan pendaftaran bagi pengunjung untuk mendapatkan akses kedalam sistem untuk mendapatkan fitur yang lebih dari seorang pengunjung. Data yang diperlukan antara lain seperti *username, email, password, NIM* serta fakultas.

Tabel 5.1 algoritma class register.

No	register
1	Inisialisasi username
2	Inisialisasi email
3	Inisialisasi password
4	Inisialisasi fullname
5	Inisialisasi nim
6	Inisialisasi faculty
7	Deklarasi button countinue
8	Pengecekan kondisi entitas tidak boleh null
9	Memanggil reference database
10	Memanggil method onchange sebagai pembacaan
11	Menyimpan username pada sharereference dengan key
12	Melakukan seleksi apakah username sudah ada atau tidak
13	Jika ada tampilkan toast nama sudah ada
14	Jika tidak makan save pada database firebase
15	Melakukan pindah dengan intent ke halaman selanjutnya
16	Memanggil method onCancelled
17	Untuk membaca error apalagi tidak mengizinkan pembacaan



5.1.4.2 *Algoritme class* lihat video

Pada *class* lihat video merupakan suatu fitur pada aplikasi yang berfungsi sebagai media untuk melihat kumpulan-kumpulan video yang terkait dengan pendampingan antara lain terdapat beberapa kategori diantaranya tuna rungu, tuna daksa, autis dan tuna netra.

Tabel 5.2 *algoritme class* lihat video

No	Lihat video
1	Inisialisasi Button play
2	Inisialisasi reference dengan tipe data database
3	Inisialisasi video url dengan tipe string
4	Deklarasi <i>method</i> onCreate
5	Memuat layout untuk mengakses video dengan R.layout
6	Mengambil id button play video
7	Memanggil listener pada button playvideo
8	Memanggil <i>method</i> onclick
9	Memanggil reference database yang mengarah pada tabel video
10	Deklarasi <i>method</i> onDataChanne yang mengizinkan pembacaan
11	data
12	Memuat url video sebagai reference dari database
13	Melakukan pindah activity dengan menggunakan intent
14	Membawa data url untuk dikirim melalui intent
15	Memanggil <i>method</i> startActivity dengan parameter intent
16	Memanggil <i>method</i> onCancelled yang berfungsi menampilkan
17	pesan error ketika database tidak mengizinkan pembacaan



5.1.4.3 *Algorithm class* tambah pengalaman

Pada *class* tambah pengalaman merupakan suatu fitur pada aplikasi yang berfungsi untuk menambah pengalaman dari setiap pendamping dalam melakukan pendampingan terhadap mahasiswa difabel di universitas brawijaya yang bertujuan agar berbagi ilmu dan pengalaman untuk pendamping-pendamping baru sehingga menjadi bekal untuk melakukan pendampingan, data yang ditambahkan diantaranya judul dan isi dari pengalaman.

Tabel 5.3 *algoritma class* tambah pengalaman.

No	Tambah pengalaman
1	Inisialisasi button save
2	Inisialisasi title dengan tipe string
3	Inisialisasi experience dengan tipe data string
4	Inisialisasi username key
5	Deklarasi <i>method</i> onCreate
6	Memuat layout untuk mengakses video dengan R.layout
7	Memanggil <i>method</i> getUsernameLocal()
8	Mengambil id button save dari layout
9	Mengambil id title dari layout
10	Mengambil id experience dari layout
11	Memanggil listener dari button save
12	Deklarasi <i>method</i> onClick dengan parameter view
13	Seleksi kondisi apabila title dalam keadaan kosong
14	Apabila kosong akan menampilkan toast peringatan
15	Jika tidak maka melakukan penyimpanan data pada firebase
16	Memanggil reference database
17	Memanggil <i>method</i> onDataChange ketika di izinkan membaca DB
18	Menyimpan pengalaman dalam tabel-tabel database
19	Memanggil <i>method</i> finish()
20	Melakukan pindah activity dengan intent ke halaman utama
21	Memanggil <i>method</i> startActivity dengan parameter intent
22	Menampilkan toast data berhasil ditambah
23	Memanggil <i>method</i> onCancelled ketika tidak diizinkan membaca
24	data pada firebase
25	



5.1.5 Perancangan Antarmuka

Perancangan ini berfungsi untuk memperjelas gambaran tentang rancangan tampilan sistem yang kemudian akan di implementasikan pada Aplikasi Pelatihan Pendamping PSLD Universitas Brawijaya. Terdapat beberapa halaman atau activity yang akan dijadikan sebuah sample perancangan. Dan akan dijelaskan lebih lanjut pada gambar-gambar berikut.

5.1.5.1 Perancangan Antarmuka Register

Halaman awal *register* berfungsi sebagai halaman pendaftaran bagi para pengguna baru agar bisa mengakses fitur-fitur pada sistem komponen-komponen yang berada didalamnya diantara lain logo dengan menggunakan *widget ImageView* kemudian ada jenis inputan *EditText* yang berupa *username, email* serta *password*. Dan mengandung tombol yang berfungsi untuk melanjutkan pengisian data selanjutnya atau disebut dengan *widget Button* yang tampak pada Gambar 5.7.

Gambar 5.7 Perancangan Antarmuka register



5.1.5.2 Perancangan Antarmuka *Register2*

Halaman *register2* merupakan lanjutan dari *register* sebelumnya fungsinya sama yakni berfungsi sebagai halaman pendaftaran bagi para pengguna baru agar bisa mengakses fitur-fitur pada sistem komponen-komponen yang berada didalamnya diantara lain logo dengan menggunakan *widget ImageView* kemudian ada jenis inputan *EditText* yang berupa *fullname*, *nim* dan *faculty*. kemudian terdapat *imageView* untuk menambah foto profil. Dan juga mengandung satu tombol yang berfungsi untuk melanjutkan melakukan pendaftaran secara keseluruhan atau disebut dengan *widget Button* yang tampak pada Gambar 5.8.

The image shows a user interface for a registration form. At the top left is a small square box labeled 'logo'. To its right is a circular area labeled 'foto'. Below these are three stacked rectangular input fields with labels 'Full Name', 'NIM', and 'Faculty'. At the bottom center is a wide rectangular button labeled 'Register'.

Gambar 5.8 Perancangan Antarmuka *register2*

5.1.5.3 Perancangan Antarmuka *Login*

Pada *activity* ini berfungsi sebagai halaman pembuka antara pengunjung untuk mengakses fitur-fitur utama pada sistem. Terdapat beberapa komponen *widget* didalamnya diantaranya terdapat sebuah logo yang diwakilkan oleh *imageView* kemudian ada *EditText* yang kemudian terdapat *username* dan *password* yang berfungsi melakukan validasi terhadap user yang telah terdaftar. kemudian terdapat *Button* atau tombol yang berfungsi untuk meneruskan data yang telah dituliskan pada *EditText* sebelumnya lebih lanjut dapat dilihat pada Gambar 5.9.



The login interface features a blue square icon with a white wheelchair symbol at the top. Below it are two white input fields with gray borders, labeled 'Username' and 'Password'. At the bottom is a white button with a gray border and the text 'Login' in bold.

Gambar 5.9 Perancangan Antarmuka *Login*

5.1.5.4 Perancangan Antarmuka *Video*

Pada halaman atau *activity* ini berfungsi sebagai halaman untuk melihat video-video pembelajaran yang berkaitan dengan cara pendampingan dan tata-cara untuk belajar Bahasa isyarat dengan beberapa *widget* diantaranya *ImageView*, *Button*, serta *LinierLayout*. Dalam Gambar 5.10.

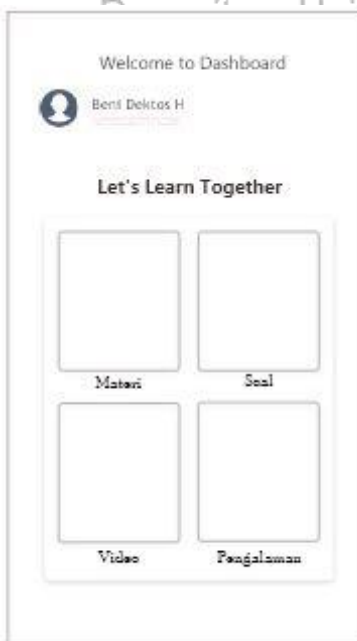
The video interface design shows a white rectangular area. At the top is a thin title bar. Below it is a large video thumbnail. Underneath the thumbnail is a grid of six smaller video thumbnails arranged in two rows and three columns. Each of these smaller thumbnails has a 'PLAY' button centered at the bottom.

Gambar 5.10 Perancangan Antarmuka *Video*



5.1.5.5 Perancangan Antarmuka Dashboard

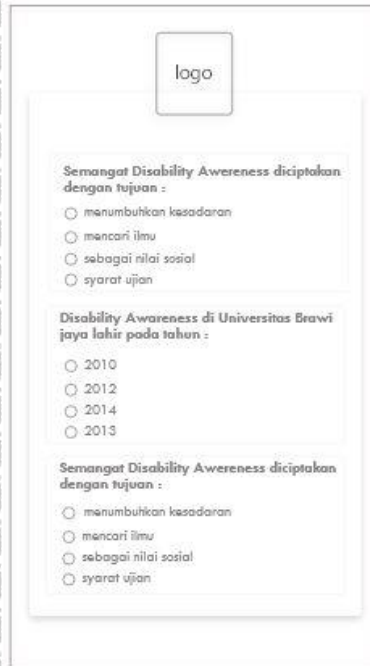
Halaman ini merupakan halaman awal dimana dari *Activity* tersebut kita bisa mengakses fitur-fitur utama yang berada pada aplikasi sehingga memberikan kemudahan dalam pengaksesannya. Halaman ini terdiri dari beberapa *widget* diantaranya *ImageView* untuk foto profil, *TextView* untuk nama *user* kemudian terdapat *linier layout* yang mengarahkan ke menu lainnya. Selengkapnya dalam Gambar 5.11.



Gambar 5.11 Perancangan Antarmuka Dashboard

5.1.5.6 Perancangan Antarmuka Soal

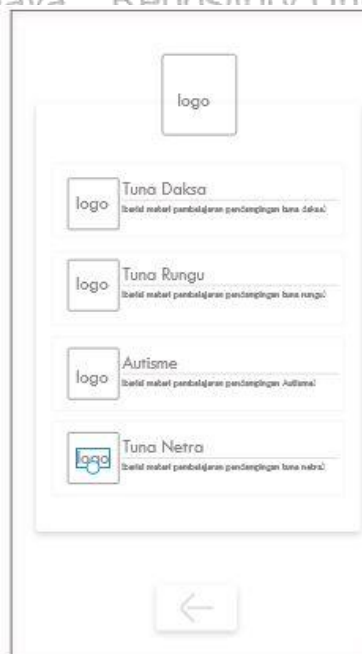
Halaman antarmuka soal berfungsi sebagai halaman bagi *user* untuk dapat mengerjakan soal-soal yang berkaitan dengan pelatihan pendampingan yang bertujuan untuk melakukan test terhadap pengetahuan terhadap *user*. Halaman ini terdiri dari beberapa *widget* diantaranya *ImageView* untuk pencantuman logo, *TextView* yang berfungsi untuk meletakkan soal, kemudian terdapat 4 *widget option* untuk meletakkan masing-masing pilihan jawaban dan berfungsi untuk memberikan pilihan *user* untuk memilih jawaban yang benar. Lebih lanjut dapat dilihat dalam Gambar 5.12.



Gambar 5.12 Perancangan Antarmuka Soal.

5.1.5.7 Perancangan Antarmuka Materi

Pada halaman ini berfungsi sebagai halaman pembuka untuk mengakses materi dari berbagai kategori pendampingan. Terdapat beberapa widget diantaranya ImageView sebagai logo, TextView sebagai penjelasan dari judul materi yang termuat. Selengkapnya dalam Gambar 5.13



Gambar 5.13 Perancangan Antarmuka Materi.



BAB 6 IMPLEMENTASI

6.1 Implementasi Sistem

Ketika berbagai tahap-tahap dalam perancangan telah dilakukan, tahap yang dilakukan selanjutnya ialah implementasi. Pada bab implementasi kemudian dibahas tentang sebuah hasil dari penerapan rancangan aplikasi pelatihan pendamping PSDL Universitas Brawijaya. Bagian ini akan membahas diantaranya spesifikasi sistem, proses implementasi yang diterapkan pada basis data, kode program dan penerapan hasil antarmuka.

6.1.1 Spesifikasi Perangkat Keras

Pada pengembangan aplikasi pelatihan pendamping PSDL Universitas Brawijaya dibutuhkan spesifikasi-spesifikasi khusus terhadap perangkat keras pendukung pembuatan aplikasi, spesifikasi yang mumpuni akan mendukung dengan baik pengembangan sistem karena dalam pembangunan sistem akan berjalan lancar dan tanpa adanya hambatan. Spesifikasi perangkat keras yang dibutuhkan antara lain akan dijelaskan pada Tabel 6.1.

Tabel 6.1 Spesifikasi Perangkat Keras

Komponen	Spesifikasi
<i>Processor</i>	Intel(R) Core(TM) i7-7200U CPU @2.50Ghz
<i>Memory (RAM)</i>	8,00 GB DDR3 L
<i>Display</i>	NVIDIA GeForce 930mx
<i>Hardisk</i>	1000 GB HDD

6.1.2 Spesifikasi Perangkat Lunak

Pada pengembangan aplikasi pelatihan pendamping PSDL Universitas Brawijaya juga dibutuhkan spesifikasi perangkat lunak sebagai pendukung yang berguna dalam pengembangan sistem, spesifikasi tersebut diantara lain akan dijelaskan pada Tabel 6.2.

Tabel 6.2 Spesifikasi Perangkat Lunak

Komponen	Spesifikasi
<i>Operation System (OS)</i>	Windows 10 Home 64-Bit
<i>Programming Language</i>	Java.xml
<i>IDE</i>	Android Studio



6.2 Implementasi Basis Data

Setelah sebelumnya dilakukan tahap perancangan basis data, basis data kemudian diimplementasikan. Pada tahapan implementasi terdapat tujuh buah tabel, yang selanjutnya bisa diamati lebih lanjut pada gambar-gambar berikut.

6.2.1 Tabel User

User	
+	admin
-	beni
	Faculty: "Filkom" NIM: "165150201111056" email: "beniryuuujzetsu@gmail.co" fullname: "beni dektos herc" password: "123" + progresmateri + progresonklik + skor url_photo_profile: "https://firebasestorage.googleapis.com/v0/b/dis" username: "beni"

Gambar 6.1 Implementasi Tabel User

Tabel *user* berguna menyimpan data-data yang berkaitan dengan berjalannya sistem terhadap *user*. tabel ini terdiri dari beberapa entitas yang diantaranya *faculty*, *nim*, *email*, *fullname*, *password*, *url photo profile*, *skor* dan *username*. Entitas-entitas tersebut saling berintegrasi dalam pembangunan sistem. Selengkapnya dalam Gambar 6.1.

6.2.2 Tabel Jadwal

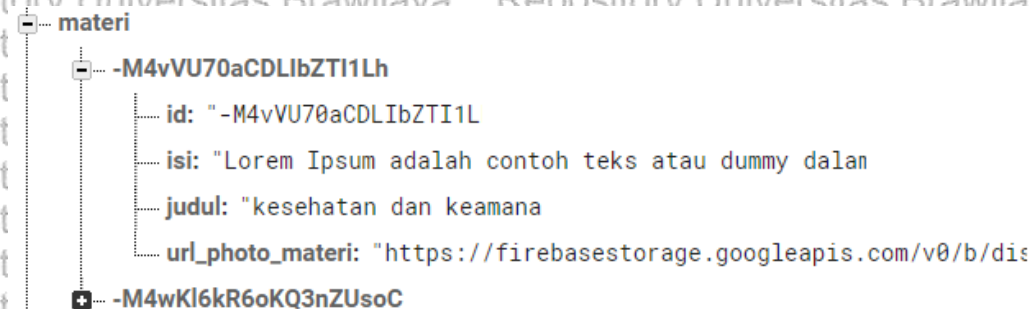
jadwal	
-	-M4hKDg9L2-VnzWNKRJE
	id: "-M4hKDg9L2-VnzWNKRJ" tanggal: "10-4-2020" tema: "Pelatihan skrips:" tempat: "PSLD 1.2" tutor: "sulistyoy" waktu: "8:0"

Gambar 6.2 Implemetasi Tabel jadwal



Tabel jadwal merupakan tabel yang berfungsi untuk menyimpan data terkait jadwal untuk pelatihan-pelatihan yang diberikan PSLD UB, tabel tersebut mempunyai beberapa entitas yang antara lain id, tema, pelatihan, tanggal pelatihan, tempat pelatihan, nama tutor yang melatih, dan waktu pelatihan. Selengkapnya dalam Gambar 6.2.

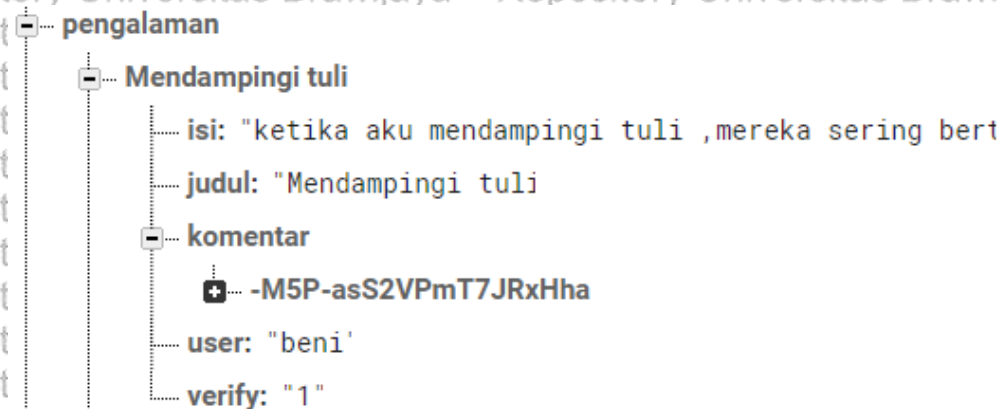
6.2.3 Tabel Materi



Gambar 6.3 Implementasi Tabel Materi

Tabel materi merupakan tabel yang berfungsi menyimpan suatu materi tambahan atau berbagai macam artikel-artikel yang menunjang pelatihan-pelatihan pendamping. Dimana tabel ini memiliki entitas-entitas diantara lain id, isi materi, judul materi, dan url foto dari materi. dalam Gambar 6.3.

6.2.4 Tabel Materi



Gambar 6.4 Implementasi Tabel Pengalaman

Pada tabel pengalaman terdapat beberapa entitas diantara lain seperti isi dari pengalaman, judul dari pengalaman, user yang berguna untuk mengetahui siapa yang menulis, verify untuk keperluan verifikasi dimana tabel ini memiliki fungsi berbagi pengalaman antar pendamping PSLD mengenai pengalaman mereka dalam mendampingi mahasiswa di tabel. dalam Gambar 6.4.



6.2.5 Tabel Soal

☐	soal
☐	Autis
☐	-M50wREDZOFAWBx1HwEJ
	id: "-M50wREDZOFAWBx1HwEJ"
	jawaban: "d"
	opsiA: "a"
	opsiB: "b"
	opsiC: "c"
	opsiD: "d"
	pertanyaan: "soal autis"
☐	Daksa
☐	Netra
☐	Quiz

Gambar 6.5 Implementasi Tabel Soal

Pada tabel soal terdapat lima macam kategori soal yang diantaranya Autis,Daksa,Netra,Tuli dan Quiz dimana pada setiap kategori terdapat beberapa entitas yaitu id soal,jawaban opsiA, opsiB, opsiC, opsiD, dan pertanyaan. Dimana tabel ini berisi memuat soal yang ditambahkan oleh admin untuk mendukung pelatihan pendampingan. Selengkapnya dalam Gambar 6.5.

6.2.6 Tabel Video

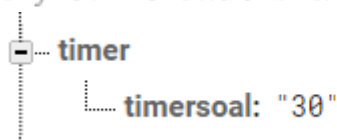
☐	vidio
	videoautissatu: "B7v8QefVFy4"
	videodua: "pjaBEouX_8g"
	videosatu: "mukNGgweHSI"

Gambar 6.6 Implementasi Tabel Video

Tabel ini berfungsi sebagai menyimpan key video yang digunakan untuk memutar video yang berisi konten-konten pelatihan pendampingan PSKD UB. Pada tabel ini terdapat entitas nama video yang berisi kunci dari pemutaran video youtube. Selengkapnya dalam Gambar 6.6.



6.2.7 Tabel Timer



Gambar 6.7 Implementasi Tabel Timer

Tabel ini berfungsi untuk menyimpan waktu yang digunakan untuk memberikan batas waktu terhadap pengerjaan kuis yang berfungsi sebagai media untuk menguji pengetahuan pendamping tentang mahasiswa difabel. Selengkapnya dalam Gambar 6.7.

6.3 Implementasi Kode Program

6.3.1 Implementasi Kode Program Register

Pada kode program Register memiliki fungsi sebagai kode untuk mendaftarkan seorang pengunjung menjadi seorang user dimana pada program ini akan menangkap beberapa data dari *User Interface Register* diantara lain foto *user*, *username*, *email*, *password*, *faculty* dan nim dari pengunjung. Dan selanjutnya dapat dilihat pada Tabel 6.3.

Tabel 6.3 Implementasi kode program register

No	Register_page.class
1	public class Register_page extends AppCompatActivity {
2	
3	EditText Username, Email, Password;
4	DatabaseReference reference;
5	
6	String USERNAME_KEY = "usernamekey";
7	String username_key = "";
8	
9	Button countinue;
10	@Override
11	protected void onCreate(Bundle savedInstanceState) {
12	super.onCreate (savedInstanceState);
13	setContentview(R.layout. activity_register_page);
14	Username = findViewById(R.id. username);
15	Email = findViewById(R.id. email);
16	Password = findViewById(R.id. password);
17	countinue = findViewById(R.id. sign_countinue);
18	
19	countinue.setOnClickListener (new
20	View.OnClickListener() {
21	@Override
22	public void onClick(View view) {
23	if (Username.getText().toString().isEmpty()) {
24	Toast.makeText (getApplicationContext(),
25	" Username Tidak Boleh Kosong ", Toast.LENGTH_LONG).show();
26	}
27	}



```

28         else if
29         (Email.getText().toString().isEmpty()) {
30             Toast.makeText(getApplicationContext(),
31             "Email Tidak Boleh Kosong", Toast.LENGTH_LONG).show();
32         }
33         else if
34         (Password.getText().toString().isEmpty()) {
35             Toast.makeText(getApplicationContext(),
36             "Password Tidak Boleh Kosong", Toast.LENGTH_LONG).show();
37         } else {
38             //menyimpan lokal
39             SharedPreferences sharedPreferences =
40             getSharedPreferences(USERNAME_KEY, MODE_PRIVATE);
41             SharedPreferences.Editor editor =
42             sharedPreferences.edit();
43             editor.putString(username_key,
44             Username.getText().toString());
45             editor.apply();
46             ;
47             // simpan pada database
48             reference =
49             FirebaseDatabase.getInstance().getReference().child("User").ch
50             ild(Username.getText().toString());
51             reference.addListenerForSingleValueEven
52             t(new ValueEventListener() {
53                 @Override
54                 public void onDataChange(@NonNull
55                 DataSnapshot dataSnapshot) {
56
57                 dataSnapshot.getRef().child("username").setValue(Username.getT
58                 ext().toString());
59
60                 dataSnapshot.getRef().child("email").setValue(Email.getText().
61                 toString());
62
63                 dataSnapshot.getRef().child("password").setValue(Password.getT
64                 ext().toString());
65
66                 }
67                 @Override
68                 public void onCancelled(@NonNull
69                 DatabaseError
70                 databaseError) {
71
72                 }
73             });
74
75             //intent pindah
76             Intent intent = new
77             Intent(Register_page.this, Register_page2.class);
78             startActivity(intent);
79

```



6.3.2 Implementasi Kode Program Tambah Pengalaman

Pada kode program tambah pengalaman berfungsi sebagai media bagi *user* yaitu pendamping untuk membagikan pengalamannya terkait pendampingan mahasiswa difabel di Universitas Brawijaya, tujuannya agar pendamping lain memiliki informasi-informasi terkait pendampingan. Data yang dimasukan antara lain judul dan isi pengalaman. Selengkapnya dapat dilihat pada tabel 6.4.

Tabel 6.4 Implementasi kode program tambah pengalaman

No	Add_pengalaman.class
1	
2	
3	public class add_pengalaman extends AppCompatActivity {
4	
5	Button saveexp ;
6	EditText Title, Experi ;
7	
8	
9	String USERNAME_KEY = "usernamekey";
10	String username_key = "";
11	String username_key_new = "";
12	DatabaseReference reference ;
13	String value ;
14	int jumlah2 ;
15	int jumlah3 ;
16	
17	@Override
18	protected void onCreate(Bundle savedInstanceState) {
19	super .onCreate(savedInstanceState);
20	setContentView(R.layout. activity_add_pengalaman);
21	
22	getUsernameLocal();
23	
24	saveexp = findViewById(R.id. save_experience);
25	Title = findViewById(R.id. Titleexp);
26	Experi = findViewById(R.id. experience);
27	
28	
29	saveexp .setOnClickListener(new View.OnClickListener()
30	{
31	@Override
32	public void onClick(View v) {
33	reference =
34	FirebaseDatabase.getInstance().getReference().child("pengalama
35	n ").child(Title .getText().toString());
36	
37	
38	reference .addListenerForSingleValueEvent(new
39	ValueEventListener() {
40	@Override
41	public void onDataChange(@NonNull
42	DataSnapshot dataSnapshot) {
43	
44	
45	dataSnapshot.getRef().child("judul").setValue(Title .getText()).



```

46 toString();
47
48 dataSnapshot.getRef().child("isi").setValue(Experi.getText().t
49 oString());
50
51 dataSnapshot.getRef().child("user").setValue(username_key_new)
52 ;
53
54 dataSnapshot.getRef().child("verify").setValue("0");
55 finish();
56 Intent intent = new
57 Intent(add_pengalaman.this, Pengalaman.class);
58 startActivity(intent);
59 finish();
60
61 Toast.makeText(getApplicationContext(), "pengalaman berhasil
62 ditambah", Toast.LENGTH_SHORT).show();
63 }
64 @Override
65 public void onCancelled(@NonNull
66 DatabaseError databaseError) {
67
68 }
69 });
70 }
71
72 });
73 }
74
75 public void getUsernameLocal() {
76 SharedPreferences sharedPreferences =
77 getSharedPreferences(USERNAME_KEY, MODE_PRIVATE);
78 username_key_new =
79 sharedPreferences.getString(username_key, "");
80
81 }
82 }
83

```

6.3.3 Implementasi Kode Program Lihat Video

Pada kode program lihat video berfungsi untuk pendamping memutar video-video yang mengedukasi untuk kebutuhan pendampingan mahasiswa difabel. Dan pada program ini pendamping dapat memutar video berdasarkan kategori. Lebih lanjut dapat dilihat pada tabel 6.5.



Tabel 6.5 Implementasi kode program lihat video

No	Lihat_Video.class
1	
2	public class PlayVideo extends Activity {
3	String videourl;
4	@SuppressWarnings("SetJavaScriptEnabled")
5	@Override
6	protected void onCreate(Bundle savedInstanceState) {
7	super.onCreate(savedInstanceState);
8	
9	requestWindowFeature(Window.FEATURE_NO_TITLE);
10	
11	getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
12	WindowManager.LayoutParams.FLAG_FULLSCREEN);
13	getWindow().requestFeature(Window.FEATURE_PROGRESS);
14	setContentView(R.layout.activity_play_video);
15	final ProgressBar Pbar;
16	Pbar = (ProgressBar) findViewById(R.id.pb4);
17	WebView wv = (WebView) findViewById(R.id.webViel1);
18	//wv.setWebViewClient(new Callback());
19	WebSettings webSettings = wv.getSettings();
20	webSettings.setBuiltInZoomControls(true);
21	webSettings.setJavaScriptEnabled(true);
22	//wv.setBackgroundColor(0x919191);
23	final String mimeType = "text/html";
24	final String encoding = "UTF-8";
25	String html = getHTML();
26	wv.loadDataWithBaseURL("", html, mimeType, encoding,
27	"");
28	final Activity activity = this;
29	
30	wv.setWebChromeClient(new WebChromeClient() {
31	public void onProgressChanged(WebView view, int
32	progress) {
33	// Activities and WebViews measure progress
34	// with different scales.
35	// The progress meter will automatically
36	// disappear when we reach 100%
37	activity.setProgress(progress * 100);
38	{
39	if (progress < 100 && Pbar.getVisibility()
40	== ProgressBar.GONE) {
41	Pbar.setVisibility(ProgressBar.VISIBLE);
42	Pbar.setProgress(progress);
43	if (progress == 100) {
44	Pbar.setVisibility(ProgressBar.GONE);
45	}
46	}
47	}
48	});
49	wv.setWebViewClient(new WebViewClient() {
50	
51	
52	
53	
54	
55	



```

56         public void onReceivedError(WebView view, int
57         errorCode, String description, String failingUrl) {
58             Toast.makeText(activity, "Oh no! " +
59             description, Toast.LENGTH_LONG).show();
60         }
61     };
62     ww.setDownloadListener(new DownloadListener() {
63
64         public void onDownloadStart(String url, String
65         userAgent, String disposition, String mimetype, long
66         contentLength) {
67             Intent intent = new Intent(Intent.ACTION_VIEW,
68             Uri.parse(url));
69             startActivity(intent);
70             finish();
71         }
72     });
73
74     }
75
76     private String getHTML() {
77
78
79         videourl =
80         getIntent().getStringExtra("EXTRA_SESSION_ID");
81         // TODO Auto-generated method stub
82         String html1 = "<iframe class=\"youtube-player\"
83         style=\"border: 0; width: 100%; height: 95%; padding: 0px;
84         margin: 0px\" id=\"ytplayer\" type=\"text/html\"
85         src=\"http://www.youtube.com/embed/\"
86             + videourl
87             + \"?fs=0\" frameborder=\"0\">\n\"
88             + \"</iframe>\n\";
89         return html1;
90     }
91
92     @Override
93     public void onBackPressed() {
94         super.onBackPressed();
95
96         onPause();
97     }
98
99     public void onPause()
100     {
101         super.onPause();
102         System.exit(0);
103     }
104
105 }
106
107

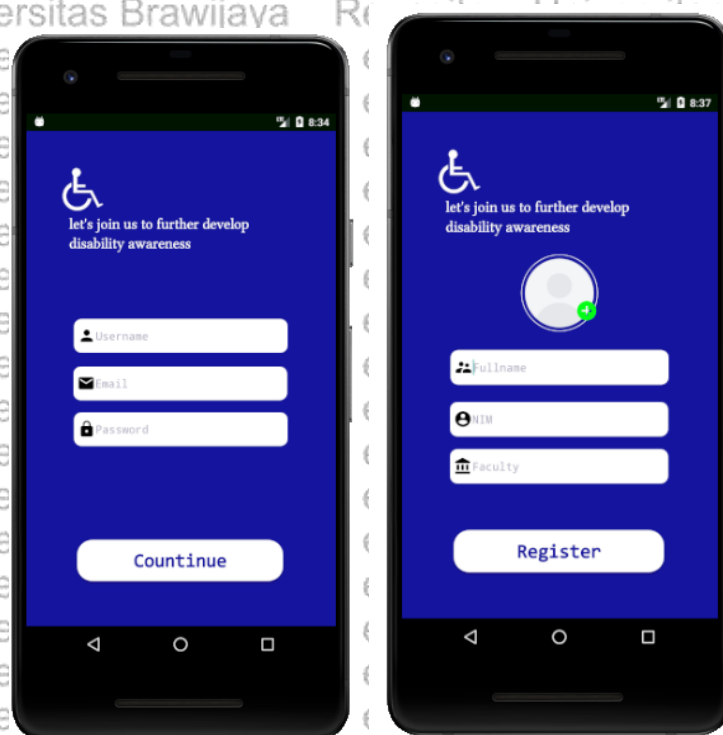
```



6.4 Implementasi Antarmuka

6.4.1 Implementasi Antarmuka *Register*

Yang pertama merupakan implementasi antar muka pada fungsi *register* yang berguna sebagai media awal melakukan pendaftaran bagi pengunjung untuk menjadi *user*. pada halaman ini pengunjung diminta mengisi data-data umum seperti nama, email, *password*, nama lengkap dan nim serta gambar, selengkapnya dapat dilihat dalam gambar 6.8.



Gambar 6.8 Implementasi antar muka *register*

6.4.2 Implementasi Antarmuka *Login*

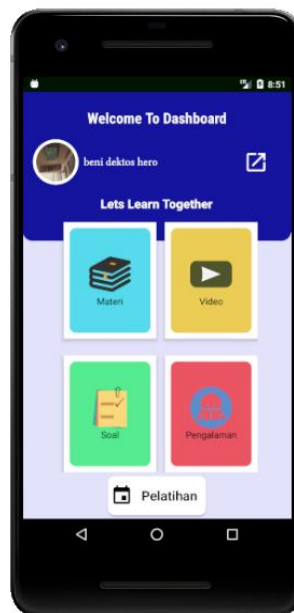
Kemudian implementasi antar muka *login*, dimana antarmuka ini memiliki sebuah fungsi sebagai antarmuka yang menghubungkan pengunjung pada menu-menu utama yang berada pada dashboard. Pada antarmuka ini pengunjung diminta untuk mengisi data berupa *username* dan *password* yang kemudian diveryifikasi oleh sistem agar bisa masuk ke dalam menu-menu pada aplikasi pelatihan pendampingan. Selanjutnya dapat dilihat dalam gambar 6.9.



Gambar 6.9 implementasi antarmuka *login*

6.4.3 Implementasi Antarmuka *Dashboard*

Selanjutnya pada implementasi antarmuka dashboard, antarmuka ini berfungsi sebagai sebuah portal pembuka utama yang berisi kumpulan-kumpulan menu utama yang bisa diakses oleh *user*. Selengkapnya dapat dilihat dalam gambar 6.10.



Gambar 6.10 Implementasi antarmuka *dashboard*



6.4.4 Implementasi Antarmuka *Pengalaman*

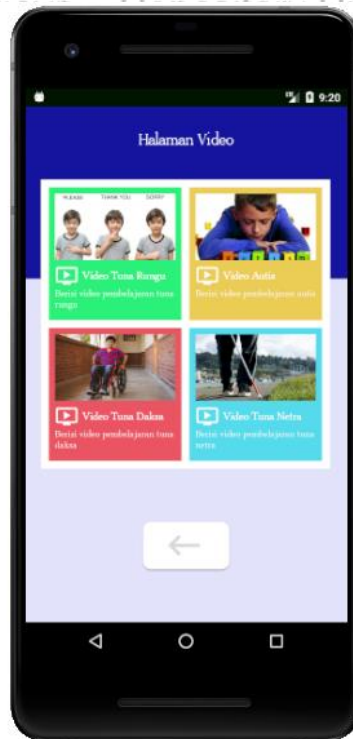
Antarmuka pengalaman memiliki fungsi sebagai tempat berbagi pengalaman antar *user* mengenai proses pendampingan mahasiswa difabel di universitas brawijaya. Pada antarmuka ini *user* cukup mengisi data berupa judul pengalaman dan isi dari pengalaman, pengalaman yang telah diisi tidak akan langsung tampil, tetapi menunggu di verifikasi oleh admin. Selengkapnya dapat dilihat dalam Gambar 6.11.



Gambar 6.11 implementasi antarmuka pengalaman.

6.4.5 Implementasi Antarmuka *Video*

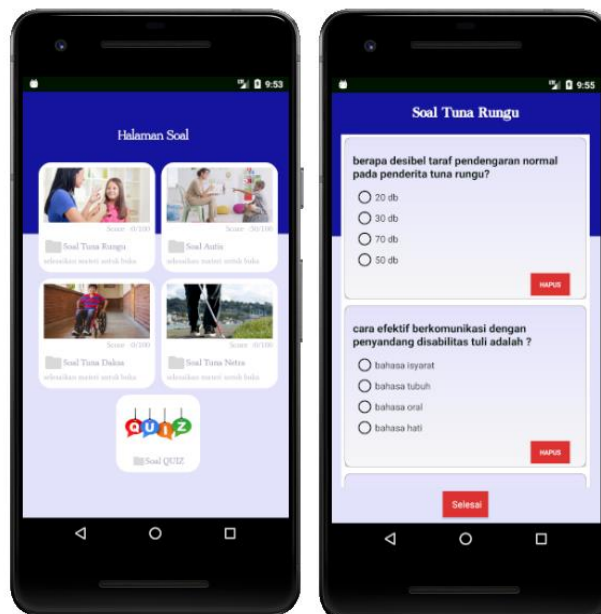
Pada antarmuka lihat video, antarmuka ini memiliki fungsi sebagai media yang digunakan para pendamping untuk menonton berbagai video yang berkaitan dengan pendampingan. Dimana video-video tersebut diklasifikasikan menjadi empat kategori yaitu video yang berkaitan dengan pelatihan tuna daksa, tuna netra, tuna rungu dan autis. Selengkapnya dapat dilihat dalam Gambar 6.12.



Gambar 6.12 Implementasi antarmuka lihat video

6.4.6 Implementasi Antarmuka soal

Pada antarmuka soal memiliki fungsi sebagai aktivitas untuk memilih jenis soal yang akan kita kerjakan yang dimana soal-soal dibagi menjadi empat kategori seperti daksa, tuli, tuna netra serta autis. Lihat dalam Gambar 6.13.

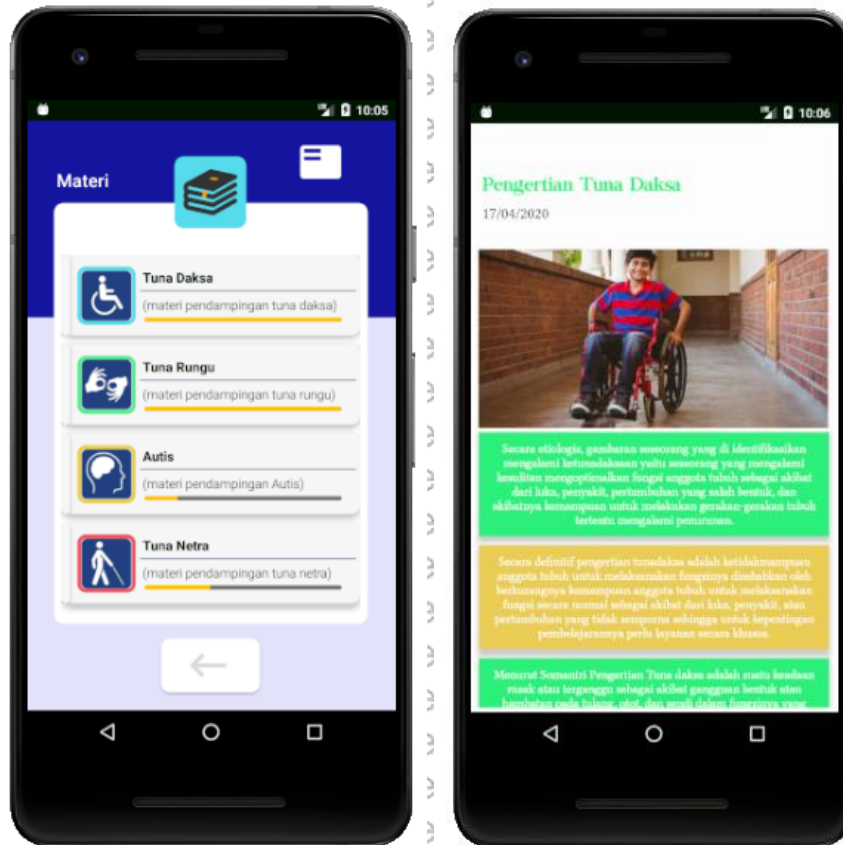


Gambar 6.13 Implementasi antarmuka soal



6.4.7 Implementasi Antarmuka materi

Pada antarmuka materi memiliki fungsi utama yaitu sebagai antarmuka yang berisi materi-materi pelatihan dalam mendampingi mahasiswa difabel dengan membedakannya menjadi empat kategori dimana setiap kategori terdapat panduan panduan dalam melakukan pendampingan tersebut. Kategori tersebut diantaranya seperti daksa,tuli, tuna netra serta autis. Selengkapnya dapat ditelitidalam Gambar 6.14.



Gambar 6.14 Implementasi antarmuka Materi

BAB 7 PENGUJIAN

7.1 Pengujian Unit

Kemudian pada tahapan yang terakhir dilakukan pengujian unit, dimana pengujian ini bertujuan melakukan pengujian terhadap objek, komponen, kelas serta hasil dari perancangan. Pada bab ini akan dilakukan sebuah pengujian yang digunakan untuk menyesuaikan algoritma yang sebelumnya sudah di implementasikan dengan analisis kebutuhan yang sebelumnya telah dibuat. Pada pengujian ini dilakukan dengan metode basis *path testing*.

7.1.1 Pengujian Unit tambah soal

1. Pseudocode

Pengujian terhadap unit method `setOnClickListener` yang akan dijelaskan pada Tabel 7.1.

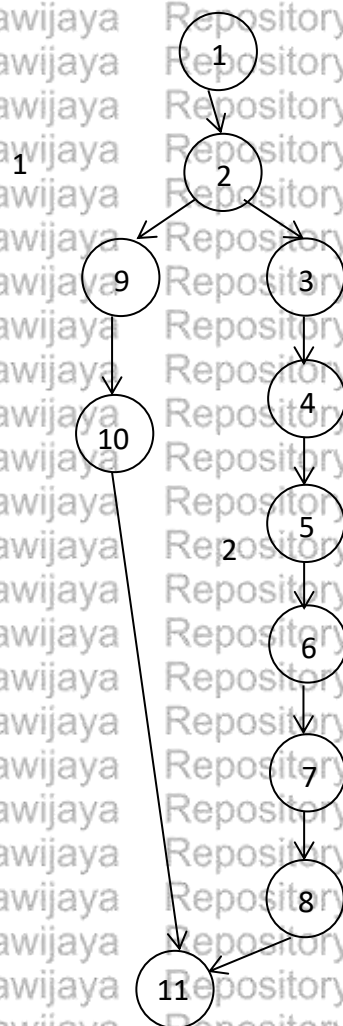
Tabel 7.1 Pseudocode Algoritma tambah soal.

No	Add_soal <code>addValueSingleEventListener()</code>
1	Inisialisasi objek soal dengan tipe data string
2	Inisialisasi objek opsia dengan tipe data string
3	Inisialisasi objek opsib dengan tipe data string
4	Inisialisasi objek opsic dengan tipe data string
5	Inisialisasi objek opsid dengan tipe data string
6	Inisialisasi objek jawaban dengan tipe data string
7	Melakukan cek kondisi terhadap objek-objek soal
8	yaitu soal, opsia, opsib, opsic, opsid
9	Jika tidak null memanggil database reference
10	Mengisi database dengan objek soal, opsia, opsib
11	Opsic, Opsid, dan jawaban.
12	Menginisialisasi objek dengan nilai null
13	Menampilkan toast dengan data telah ditambah
14	Menginisiasi intent pindah ke halaman utamasoal
15	Memanggil <code>method finish</code>
16	Jika null maka menampilkan toast
17	Menampilkan toast objek tidak boleh null
18	Selesai

2. Basis Path Testing

2.1 Flow Graph

Berdasar pada algoritma sebelumnya akan menjadi dasar untuk membuat *flowgraph* untuk fitur tambah soal sebagai berikut dalam Gambar 7.1.



Gambar 7.1 Flowgraph tambah soal

2.2 Cyclomatic Complexity

1. $V(G) = \text{jumlah region} = 2$

2. $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 11 - 11 + 2 = 2$

3. $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

2.3 Independent Path

1. Jalur 1 = 1-9-10-11

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



Tabel 7.2 Hasil Pengujian Unit Tambah soal.

jalur	Prosedur uji	Hasil pengujian yang diharapkan	Hasil pengujian	status
1	<i>method</i> <i>singleEventValue</i> listener menambahkan soal dari <i>user</i> yang kemudian ditambahkan pada <i>firebase database</i> pada tabel soal ketika tidak ada data	“Menjalankan code program berupa <i>Listener</i> dan menampilkan pesan “opsi tidak boleh kosong “	“Menjalankan code program berupa <i>Listener</i> dan menampilkan pesan “soal berhasil ditambah “	valid
2	<i>method</i> <i>singleEventValue</i> listener menambahkan soal dari <i>user</i> yang kemudian ditambahkan pada <i>firebase database</i> pada tabel jika terdapat data	code program berupa <i>Listener</i> dijalankan dan menampilkan pesan “soal berhasil ditambah	Menjalankan code program berupa <i>Listener</i> dan menampilkan pesan “soal berhasil ditambah”	valid

7.1.2 Pengujian Unit Edit Profil

1. Pseudocode

Pengujian selanjutnya terhadap *method* *addValueEventSingleListener* dari fungsi edit profil yang merupakan sekumpulan code program dan *method* yang berasal dari *class* *edit_profile*. Selengkapnya dapat dilihat pada Tabel 7.3.



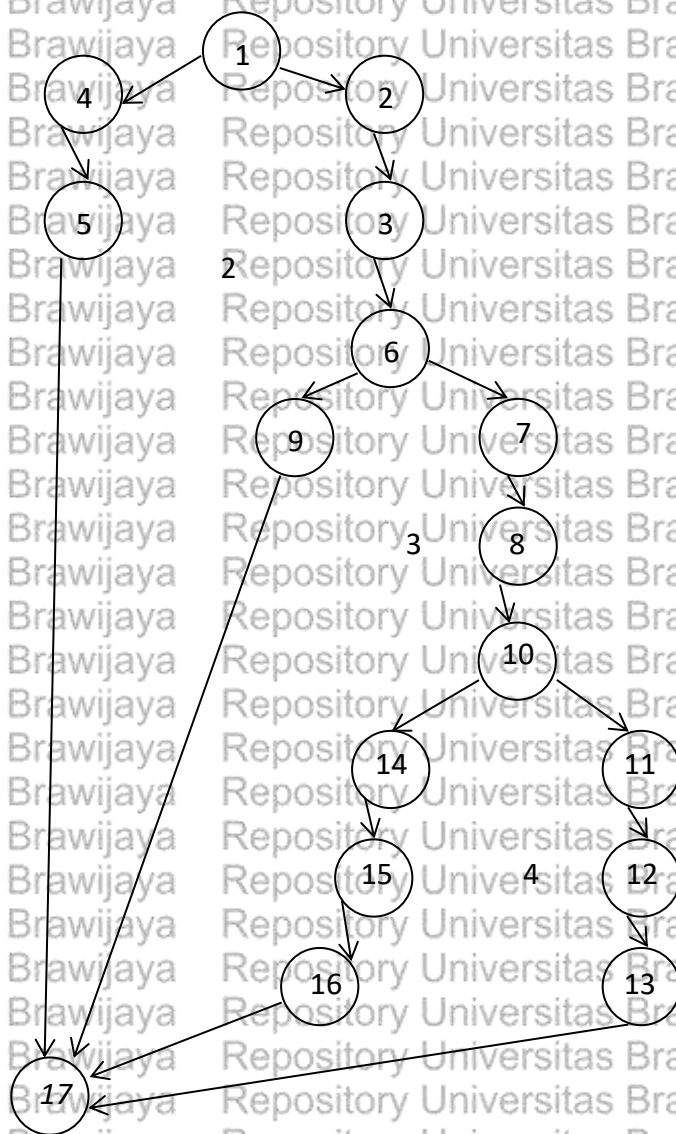
Tabel 7.3 Pseudocode Algoritma Edit Profile

No	edit_profile addValueSingleEventListener()
1	
2	Memanggil reference addValueEventListener (1)
3	Jika data berhasil membaca atau onDataChange maka (2)
4	Fullname memuat data fullname dari database
5	nim memuat data nim dari database
6	Faculty memuat data faculty dari database
7	Photo memuat data photo dari database
8	Jika data tidak di izinkan dibaca muncul (4)
9	Memunculkan pesan error dari database (5)
10	user melakukan klik pada button save (6)
11	memanggil reference database user "User" (7)
12	membaca entitas tabel dari database
13	memuat entitas dan mengisi nilai dari editText (8)
14	jika tidak di izin membaca menampilkan pesan error (9)
15	melakukan pengecekan apakah photo profile tidak null (10)
16	jika tidak memanggil database storage dan get ekstensi (11)
17	menempatkan file photo baru sebagai pengganti (12)
18	jika sukses simpan data dan kembali ke profil (13)
19	jika data berupa null (14)
20	melakukan pemindahan halaman langsung ke profile (15)
21	memanggil method finish (16)
22	selesai (17)

2. Basis Path Testing

2.1 Flow Graph

Berdasar pada algoritma sebelumnya akan menjadi dasar untuk membuat *flowgraph* untuk fitur *edit profile* sebagai berikut dalam Gambar 7.2.



Gambar 7.2 Flowgraph edit profile



2.2 Cyclomatic Complexity

1. $V(G) = \text{jumlah region} = 4$
2. $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 19 - 17 + 2 = 4$
3. $V(G) = \text{jumlah predicate node} + 1 = 3 + 1 = 4$

2.3 Independent Path

1. Jalur 1 = 1 - 4 - 5 - 17
2. Jalur 2 = 1 - 2 - 3 - 6 - 9 - 17
3. Jalur 3 = 1 - 2 - 3 - 6 - 7 - 8 - 10 - 14 - 15 - 16 - 17
4. Jalur 4 = 1 - 2 - 3 - 6 - 7 - 8 - 10 - 11 - 12 - 12 - 17

Tabel 7.4 Hasil Pengujian Edit Profile.

jalur	Prosedur uji	Hasil pengujian yang diharapkan	Hasil pengujian	status
1	<i>method</i> Edit_ProfileSinggel eventvalueListener() melakukan fungsi edit data dari user terkait data diri dengan tidak adanya data dari firebase	Menjalankan code program berupa Add valueEventListener dan menampilkan pesan "null object reference"	Menjalankan code program berupa Add valueEventListener dan menampilkan pesan "null object reference"	valid
2	<i>method</i> Edit_ProfileSinggel eventvalueListener() melakukan fungsi edit data dari user terkait data diri dengan tidak adanya objek dari firebase	Menjalankan code program berupa Add valueEventListener dan menampilkan pesan "null object reference"	Menjalankan code program berupa Add valueEventListener dan menampilkan pesan "null object reference"	valid

Tabel 7.5 Hasil Pengujian Edit *Profile* (Lanjutan).

jalur	Prosedur uji	Hasil pengujian yang diharapkan	Hasil pengujian	status
3	<i>method</i> Edit_ProfileSinggel eventvalueListener() melakukan fungsi edit data dari user terkait data diri,dengan tidak adanya foto yang diganti	Menjalankan code program berupa kondisi photo_location != null dan memindahkan <i>activity</i> edit ke <i>activity profile</i> dengan foto yang sama	Menjalankan code program berupa kondisi photo_location != null dan memindahkan <i>activity</i> edit ke <i>activity profile</i> dengan foto yang sama	valid
4	<i>method</i> Edit_ProfileSinggel eventvalueListener() melakukan fungsi edit data dari user terkait data diri,dengan ada foto <i>profile</i> yang diganti	Menjalankan code program berupa kondisi photo_location != null dan memindahkan <i>activity</i> edit ke <i>activity profile</i> dengan foto yang berbeda	Menjalankan code program berupa kondisi photo_location != null dan memindahkan <i>activity</i> edit ke <i>activity profile</i> dengan foto yang berbeda	valid

7.1.3 Pengujian Unit tambah pengalaman

1. Pseudocode

Pengujian dilakukan terhadap method addValueSingleEventListener tambah pengalaman dari sekumpulan fungsi tambah pengalaman yang berasal dari class add_pengalaman akan dijelaskan pada Tabel 7.6.



Tabel 7.6 Pseudocode Algoritma tambah pengalaman.

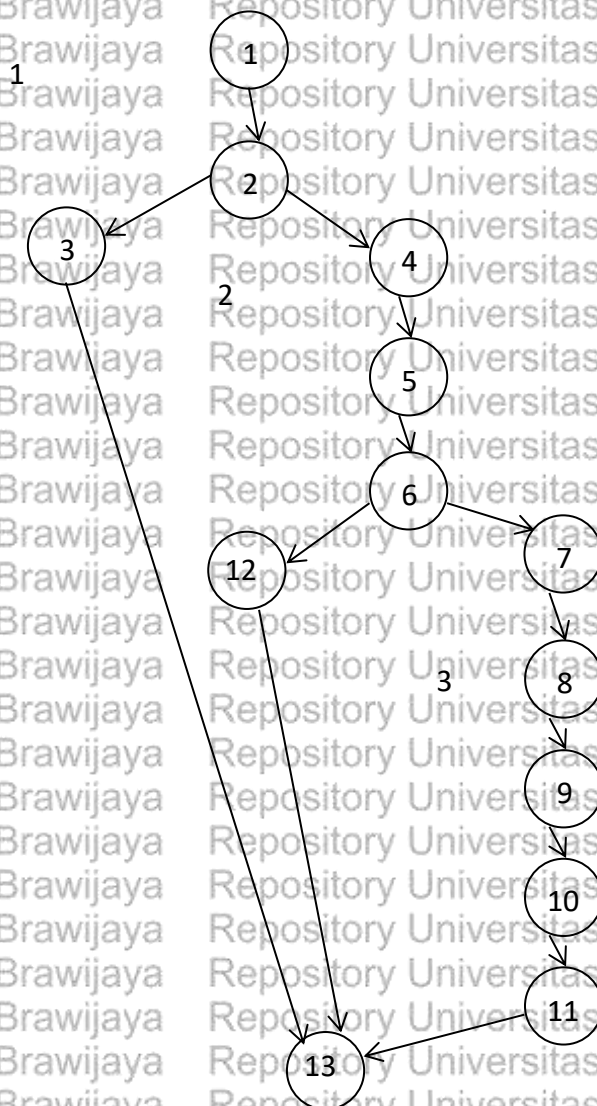
No	Add_pengalaman addValueSingleEventListener()
1	
2	Memanggil method addvalueSingleEventListener (1)
3	
4	Memeriksa apakah judul atau isi kosong atau tidak. (2)
5	Jika kosong maka akan menampilkan toast tidak
6	Boleh kosong (3)
7	
8	Jika tidak kosong maka data dilewatkan (4)
9	Memanggil reference database pengalaman (5)
10	Memanggil listener addlistenerforsinggleValueEvent (6)
11	
12	Melakukan pembacaan data dengan data onChange (7)
13	
14	Set isi judul, isi, user dan verify dengan data editText (8)
15	Memanggil method finish (9)
16	Pindah activity dengan intent ke halaman pengalaman (10)
17	Tampilkan toast "pengalaman telah ditambah" (11)
18	
19	Jika data tidak diizinkan dibaca maka tampil pesan error
20	Melalui method onCancelled DatabaseError (12)
21	Selesai (13)

2. Basis Path Testing

2.1 Flow Graph

Berdasar pada algoritma sebelumnya akan menjadi dasar untuk membuat *flowgraph* untuk fitur tambah pengalaman sebagai berikut dalam Gambar

7.3.



Gambar 7.3 Flowgraph tambah pengalaman

2.2 Cyclomatic Complexity

1. $V(G) = \text{jumlah region} = 3$
2. $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 14 - 13 + 2 = 3$
3. $V(G) = \text{jumlah predicate node} + 1 = 2 + 1 = 3$



2.3 Independent Path

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

Tabel 7.7 Hasil Pengujian Tambah Pengalaman.

jalur	Prosedur uji	Hasil pengujian yang diharapkan	Hasil pengujian	status
1	<i>Method</i> addValueSingle EventListener() melakukan penambahan data pengalaman dengan nilai null	Menjalankan code program berupa kondisi judul dan isi pengalaman berupa null menampilkan toast “judul tidak boleh kosong”	Menjalankan code program berupa kondisi judul dan isi pengalaman berupa null menampilkan toast “judul tidak boleh kosong”	valid
2	<i>Method</i> addValueSingle EventListener() melakukan penambahan data pengalaman dengan nilai tidak null dan tidak bisa membaca database	Menjalankan code program berupa kondisi judul dan isi pengalaman tidak null dan menampilkan pesan error tidak mengizinkan pembacaan firebase	Menjalankan code program berupa kondisi judul dan isi pengalaman tidak null dan menampilkan pesan error tidak mengizinkan pembacaan firebase	valid



Tabel 7.8 Hasil Pengujian Tambah Pengalaman. (lanjutan)

jalur	Prosedur uji	Hasil pengujian yang diharapkan	Hasil pengujian	status
3	<i>Method</i> <code>addValueSingle</code> <code>EventListener()</code> melakukan penambahan data pengalaman dengan nilai tidak null dan memberi izin pembacaan database	Menjalankan code program berupa kondisi judul dan isi pengalaman tidak dan menampilkan pesan “ pengalaman berhasil ditambah “	Menjalankan code program berupa kondisi judul dan isi pengalaman tidak dan menampilkan pesan “ pengalaman berhasil ditambah “	valid

7.2 Pengujian Integrasi

1. *Pseudocode*

Pada pengujian ini menggunakan pengujian integrasi dimana memiliki fungsi untuk menguji *method*, pada pengujian kali ini akan dijalankan kepada *method* `onClick` pada listener `mButtonStartPause` pada class *quiz*. Yang berfungsi menjalankan waktu dan melakukan pause terhadap waktu.

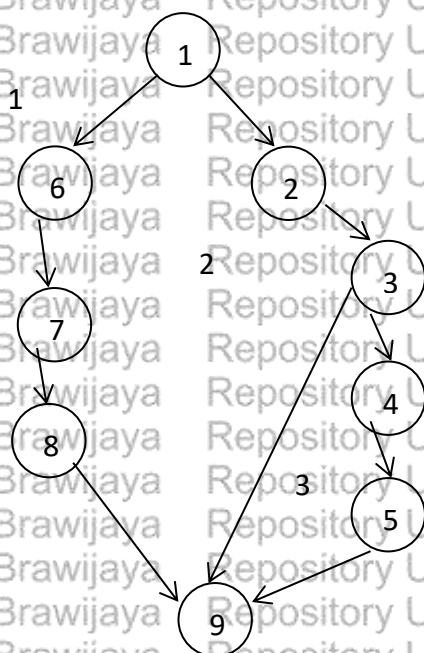
Tabel 7.9 *Pseudocode method onClick*

No	<i>quiz</i>
1	Memanggil <i>method</i> <code>getUsernameLocal()</code> } ①
2	Check kondisi apakah <code>mTimerRunning == true</code> ②
3	Jika ia apakah <code>username key_new equals ("admin")</code> ③
4	Memanggil <i>method</i> <code>pauseTimer()</code> ④
5	Set <code>visibility soal = invisible</code> ⑤
6	Else ⑥
7	Memanggil <i>method</i> <code>startTime()</code> ⑦
8	Set <code>visibility soal = visible</code> ⑧
9	Selesai ⑨



2. Basis Path Testing

2.1 Flowgraph



Gambar 7.4 Flowgraph method onClick

2.2 Cyclomatic Complexity

1. $V(G) = \text{jumlah region} = 3$
2. $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 10 - 9 + 2 = 3$
3. $V(G) = \text{jumlah predicate node} + 1 = 2 + 1 = 3$

2.3 Independent Path

1. Jalur 1 = 1 – 2 – 3 – 9
2. Jalur 2 = 1 – 6 – 7 – 8 – 9
3. Jalur 3 = 1 – 2 – 3 – 4 – 5 – 9

Tabel 7.10 Hasil Pengujian integrasi *method onclick*

jalur	Prosedur uji	Hasil pengujian yang diharapkan	Hasil pengujian	status
1	<i>Method</i> dengan nama <i>onClick</i> dari listener <i>starttime</i> yang berasal dari class <i>quiz</i> dengan kondisi waktu berjalan bernilai <i>true</i> dan <i>user login</i>	Menjalankan <i>code method</i> dengan waktu yang berjalan tanpa berhenti	Menjalankan <i>code method</i> dengan waktu yang berjalan tanpa berhenti	valid
2	<i>Method</i> dengan nama <i>onClick</i> dari listener <i>starttime</i> yang berasal dari class <i>quiz</i> dengan kondisi waktu berjalan bernilai <i>false</i>	Menjalankan <i>code method</i> dengan waktu yang telah terhenti	Menjalankan <i>code method</i> dengan waktu yang telah terhenti	valid
3	<i>Method</i> dengan nama <i>onClick</i> dari listener <i>starttime</i> yang berasal dari class <i>quiz</i> dengan kondisi waktu berjalan bernilai <i>true</i> dan <i>user login</i> adalah admin	Menjalankan <i>code method</i> dengan waktu yang berjalan dan dapat berhenti dengan <i>method pauseTimer()</i>	Menjalankan <i>code method</i> dengan waktu yang berjalan dan dapat berhenti dengan <i>method pauseTimer()</i>	valid



7.3 Pengujian Validasi

7.3.1 Pengujian Validasi Login

1. Kasus uji telah login

Tabel 7.11 Pengujian validasi berhasil login

Nama Kasus	Login
Prosedur	1. Aktor mengisi <i>username</i> dan <i>password</i> dengan lengkap 2. Aktor menekan tombol <i>login</i> pada Activity. 3. Sistem memeriksa <i>username</i> serta <i>password</i> . 4. Sistem menampilkan Activity utama.
Hasil yang diharapkan	Aktor berhasil melakukan <i>login</i> sehingga memiliki hak akses yang dikenali sistem
Hasil	Aktor berhasil melakukan <i>login</i> sehingga memiliki hak akses yang dikenali sistem
Status	valid

2. Kasus uji *login* : alternatif 1

Tabel 7.12 Pengujian validasi *login* alternatif *username* salah

Nama Kasus	Username salah
Prosedur	1. aktor memasukan <i>username</i> yang salah 2. aktor memasukan <i>password</i> yang benar 3. aktor menekan tombol <i>login</i>
Hasil yang diharapkan	Sistem menampilkan toast “ <i>username</i> salah”
Hasil	Sistem menampilkan toast “ <i>username</i> salah”
Status	valid



3. Kasus uji login : alternatif 2

Tabel 7.13 Pengujian validasi login alternatif *password* salah

Nama Kasus	<i>password</i> salah
Prosedur	1. aktor memasukkan <i>username</i> yang benar 2. aktor memasukkan <i>password</i> yang salah 3. aktor menekan tombol login
Hasil yang diharapkan	Sistem menampilkan toast “ <i>password</i> salah”
Hasil	Sistem menampilkan toast “ <i>password</i> salah”
Status	valid

7.3.2 Pengujian Validasi *Register*

1. Kasus uji telah *Register*

Tabel 7.14 Pengujian validasi berhasil *register*

Nama Kasus	<i>Register</i>
Prosedur	1. Aktor menekan “<i>Create New Account</i>”. 2. Sitem menampilkan <i>Activity Register</i>. 3. Aktor mengisikan <i>Username</i>, <i>Email</i> serta <i>Password</i>. 4. Aktor menekan tombol <i>Continue</i>. 5. Sistem menampilkan <i>Activity Register</i> kedua. 6. Aktor mengisikan kolom <i>fullname</i>, <i>nim</i> serta <i>faculty</i> 7. Aktor mengunggah foto profil 8. Aktor menekan tombol <i>Register</i>.
Hasil yang diharapkan	Aktor berhasil melakukan <i>register</i>
Hasil	Aktor berhasil melakukan <i>register</i>
Status	valid



2. Kasus uji *register* : alternatif 1

Tabel 7.15 Pengujian validasi *register* alternatif salah satu kolom tidak diisi

Nama Kasus	Salah satu kolom tidak diisi
Prosedur	1. Aktor tidak mengisi kolom 2. Aktor tidak mengisi salah satu kolom 3. Aktor menekan tombol <i>continue</i> 4. Sistem memeriksa kolom
Hasil yang diharapkan	Sistem menampilkan toast “kolom tidak boleh kosong”
Hasil	Sistem menampilkan toast “kolom tidak boleh kosong”
Status	valid

3. Kasus uji *register* : alternatif 2

Tabel 7.16 Pengujian validasi *register* alternatif *username* telah ada

Nama Kasus	<i>Username</i> telah ada
Prosedur	1. Aktor tidak mengisi kolom 2. Aktor menekan tombol <i>continue</i> 3. Sistem memeriksa kolom
Hasil yang diharapkan	Sistem menampilkan toast “ <i>userame</i> sudah ada”
Hasil	Sistem menampilkan toast “ <i>userame</i> sudah ada”
Status	valid

7.3.3 Pengujian Validasi lihat materi

Tabel 7.17 Pengujian validasi lihat materi

Nama Kasus	Lihat materi
Prosedur	1. Aktor menekan tombol yang berisi materi. 2. Sistem menampilkan beberapa kategori materi. 3. Aktor memilih materi yang ingin dipelajari.
Hasil yang diharapkan	Aktor berhasil membuka materi
Hasil	Aktor berhasil membuka materi
Status	valid



7.3.4 Pengujian Validasi lihat video

Tabel 7.18 Pengujian validasi lihat video

Nama Kasus	Lihat video
Prosedur	1. Aktor menekan tombol yang berisi video. 2. Sistem menampilkan beberapa video pelatihan. 3. Aktor memutar video yang ingin dipelajari.
Hasil yang diharapkan	Aktor berhasil memutar video
Hasil	Aktor berhasil memutar video
Status	valid

7.3.5 Pengujian Validasi lihat pengalaman

Tabel 7.19 Pengujian validasi lihat pengalaman

Nama Kasus	Lihat pengalaman
Prosedur	1. Aktor menekan tombol yang berisi pengalaman. 2. Sistem menampilkan beberapa list pengalaman. 3. Aktor memilih pengalaman yang ingin dipelajari.
Hasil yang diharapkan	Aktor berhasil membuka pengalaman
Hasil	Aktor berhasil membuka pengalaman
Status	valid

7.3.6 Pengujian Validasi komentar

Tabel 7.20 Pengujian validasi komentar

Nama Kasus	komentar
Prosedur	1. Aktor membuka salah satu pengalaman. 2. Aktor mengisi kolom komentar. 3. Aktor menekan tombol kirim. 4. Sistem menyimpan komentar.
Hasil yang diharapkan	Aktor berhasil menambahkan komentar
Hasil	Aktor berhasil menambahkan komentar
Status	valid



7.3.7 Pengujian Validasi kerjakan soal

Tabel 7.21 Pengujian validasi kerjakan soal

Nama Kasus	Kerjakan soal
Prosedur	1. Aktor membuka menu soal. 2. Aktor mengerjakan soal-soal pilihan ganda. 3. Aktor menekan tombol selesai.
Hasil yang diharapkan	Aktor berhasil mengerjakan soal
Hasil	Aktor berhasil mengerjakan soal
Status	valid

7.3.8 Pengujian Validasi lupa *password*

Tabel 7.22 Pengujian validasi lupa *password*

Nama Kasus	Lupa <i>password</i>
Prosedur	1. Aktor membuka menu lupa <i>password</i>. 2. Aktor mengisi email dan <i>username</i>. 3. Sistem mengirimkan <i>password</i> ke email.
Hasil yang diharapkan	Sistem berhasil mengirimkan <i>password</i>
Hasil	Sistem berhasil mengirimkan <i>password</i>
Status	valid

7.3.9 Pengujian Validasi lihat jadwal pelatihan

Tabel 7.23 Pengujian validasi lihat jadwal pelatihan

Nama Kasus	Lihat jadwal pelatihan
Prosedur	1. Aktor menekan tombol lihat jadwal. 2. Sistem menampilkan jadwal pelatihan. 3. Aktor melakukan klik pada jadwal. 4. Sistem menampilkan detail pada jadwal
Hasil yang diharapkan	Sistem menampilkan jadwal
Hasil	Sistem menampilkan jadwal
Status	valid

7.3.10 Pengujian Validasi tambah jadwal

Tabel 7.24 Pengujian validasi tambah jadwal

Nama Kasus	Tambah jadwal
Prosedur	1. Aktor memilih menu tambahkan jadwal. 2. Aktor mengisikan waktu,tempat dan tutor pelatihan 3. Aktor menekan tombol tambah. 4. Sistem menyimpan jadwal.
Hasil yang diharapkan	Sistem berhasil menyimpan jadwal
Hasil	Sistem berhasil menyimpan jadwal
Status	valid

7.3.11 Pengujian Validasi lihat user

Tabel 7.25 Pengujian validasi lihat user

Nama Kasus	Lihat user
Prosedur	1. Aktor memilih menu user. 2. Sistem menampilkan user.
Hasil yang diharapkan	Sistem berhasil menampilkan user
Hasil	Sistem berhasil menampilkan user
Status	valid

7.3.12 Pengujian Validasi hapus user

Tabel 7.26 Pengujian validasi hapus user

Nama Kasus	Hapus user
Prosedur	1. Aktor memilih menu user. 2. Sistem menampilkan user. 3. Aktor menekan tombol hapus pada user. 4. Sistem menghapus user.
Hasil yang diharapkan	Sistem berhasil menghapus user
Hasil	Sistem berhasil menghapus user
Status	valid



7.3.13 Pengujian Validasi hapus soal

Tabel 7.27 Pengujian validasi hapus soal

Nama Kasus	Hapus soal
Prosedur	1. Aktor memilih menu soal. 2. Sistem menampilkan kumpulan soal. 3. Aktor memilih tombol hapus pada soal
Hasil yang diharapkan	Aktor berhasil menghapus soal
Hasil	Aktor berhasil menghapus soal
Status	valid

7.3.14 Pengujian Validasi tambah materi

Tabel 7.28 Pengujian validasi tambah materi

Nama Kasus	Tambah materi
Prosedur	1. Aktor memilih menu materi. 2. Sistem menampilkan <i>activity</i> materi. 3. Aktor memilih tombol tambah materi 4. Sistem menampilkan kolom tambah materi. 5. Aktor mengisi kolom dengan materi. 6. Aktor menekan tombol simpan 7. Sistem menyimpan materi.
Hasil yang diharapkan	Aktor berhasil menambah materi
Hasil	Aktor berhasil menambah materi
Status	valid



7.3.15 Pengujian Validasi hapus materi

Tabel 7.29 Pengujian validasi hapus materi

Nama Kasus	Hapus materi
Prosedur	1. Aktor memilih menu materi. 2. Sistem menampilkan kumpulan materi 3. Aktor memilih tombol hapus pada materi 4. Sistem menghapus materi.
Hasil yang diharapkan	Aktor berhasil menghapus materi
Hasil	Aktor berhasil menghapus materi
Status	valid

7.3.16 Pengujian Validasi verifikasi pengalaman

Tabel 7.30 Pengujian validasi verifikasi pengalaman

Nama Kasus	Verifikasi pengalaman
Prosedur	1. Aktor memilih menu pengalaman. 2. Sistem menampilkan kumpulan pengalaman. 3. Aktor memilih tombol verifikasi pengalamani 4. Sistem menampilkan pengalaman di sisi user.
Hasil yang diharapkan	Aktor berhasil melakukan verifikasi
Hasil	Aktor berhasil melakukan verifikasi
Status	valid



7.3.17 Pengujian Validasi hapus jadwal

Tabel 7.31 Pengujian validasi hapus jadwal

Nama Kasus	Hapus jadwal
Prosedur	1. Aktor memilih tombol jadwal. 2. Sistem menampilkan jadwal. 3. Aktor memilih tombol hapus jadwal. 4. Sistem menghapus jadwal.
Hasil yang diharapkan	Aktor berhasil menghapus jadwal
Hasil	Aktor berhasil menghapus jadwal
Status	valid

7.3.18 Pengujian Validasi kerjakan quiz

Tabel 7.32 Pengujian validasi hapus jadwal

Nama Kasus	Kerjakan quiz
Prosedur	1. Aktor memilih menu <i>quiz</i>. 2. Sistem menampilkan <i>activity quiz</i>. 3. Aktor memilih tombol <i>start</i>. 4. Sistem menampilkan soal.
Hasil yang diharapkan	Aktor berhasil mengerjakan <i>quiz</i>
Hasil	Aktor berhasil mengerjakan <i>quiz</i>
Status	valid

7.4 Pengujian Usability

Pada pengujian yang terakhir merupakan pengujian *usability* dengan menggunakan suatu metode dengan nama SUS atau *System Usability Scale* dengan memberikan akses pada sistem yang sebelumnya telah dikembangkan untuk menjadi bahan responden. Setelah akses sistem diberikan dan telah dicoba, kemudian diberikan sebanyak sepuluh pertanyaan mengenai kemudahan dari sistem dengan pola pertanyaan positif dan kemudian negatif yang dibuat dan dilakukan pengukuran menggunakan skala *likert*. Pada Skala ini merupakan model pengukuran yang digunakan oleh SUS untuk memberikan tolak ukur terhadap tingkat kenyamanan dalam sistem terkait. Terdiri dari lima skala yang menjadi nilai dari skala ini, antara lain sangat setuju, setuju, netral, tidak setuju serta sangat tidak setuju. Selanjutnya dari kesepuluh pertanyaan yang akan diberikan untuk pertanyaan dengan nomor ganjil diberikan pertanyaan berupa



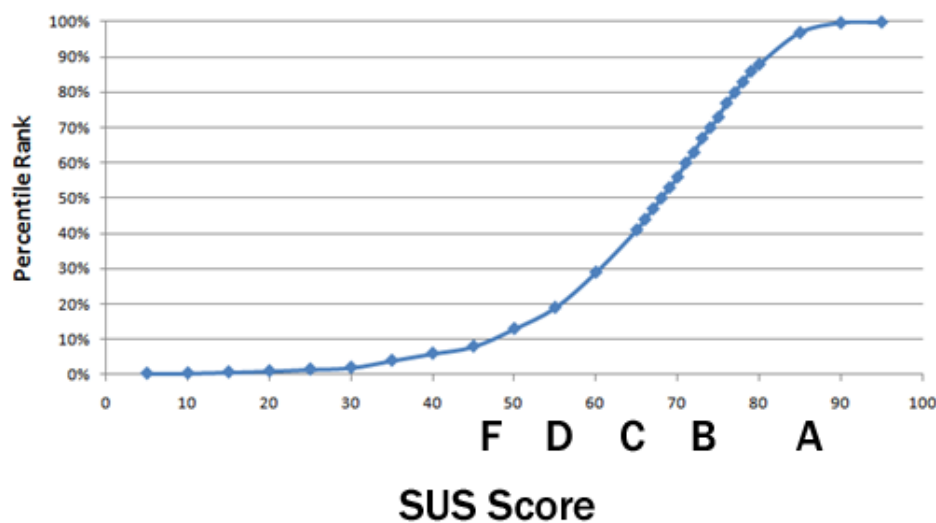
pertanyaan positif dan untuk nomor soal genap akan diberikan sebuah pertanyaan negative didalam penggunaan sistem terkait. Ketika menjawab pertanyaan dengan urutan ganjil maka nilainya akan dikurangi satu, kemudian ketika menjawab soal dengan urutan genap nilainya akan dikurangi lima. Selanjutnya dari setiap nilai pertanyaan akan dijumlahkan dan kemudian dikali dengan 2,5 dan hasil akhir diperoleh berupa ukuran sebuah tingkat *usability* dari sistem yang terkait. Kemudian akan didapatkan sebuah nilai akhir yang memiliki *range* nilai 0 – 50,9 untuk SUS, selanjutnya marginal bernilai 51 – 70,9 serta 71 – 100 untuk *acceptable*. Berikut merupakan sepuluh urutan pertanyaan yang kemudian akan diberikan pada beberapa responden yang mencoba sistem, dapat diamati pada Tabel 7.33.

Tabel 7.33 Daftar Pertanyaan SUS

No	Pertanyaan
1	Sistem ini dapat mempermudah saya mengenali tentang disabilitas
2	Saya perlu mendapatkan tentang tatacara menggunakan sistem ini pada awal
3	Materi yang diberikan pada sistem ini sudah jelas
4	Bahasa dalam penggunaan sistem ini sulit dipelajari
5	Sistem ini dibutuhkan untuk mengenali dan mempelajari tentang disabilitas
6	Saya kesulitan dalam memahami sistem ini
7	Tampilan dan tombol-tombol pada sistem mudah dipahami
8	Saya merasa kebingungan pada alur sistem ini
9	Saya merasa fitur pada sistem ini sudah terintegrasi dengan baik
10	Saya merasa terdapat fitur dalam sistem yang sulit dipahami

7.4.1 Hasil Pengujian *Usability*

SUS merupakan suatu metode yang dipergunakan untuk melakukan pengujian *usability*, SUS memiliki pengertian yaitu sebuah teknologi independen yang sudah sejak lama telah diuji pada perangkat keras serta perangkat lunak seperti konsumen, situs web, ponsel serta IVR. SUS merupakan sebuah metode yang telah menjadi standar industri dengan berdasar pada referensi lebih dari 600 publikasi. Skor pada metode ini memiliki rata-rata dari 500 publikasi adalah 68. Ketika skor tersebut telah melewati 68 maka akan dianggap diatas rata-rata serta yang berada dibawahnya merupakan dibawah rata-rata. Untuk mendapatkan cara terbaik, skor SUS diinterpretasikan menjadi sebuah perangkat presentil melalui sebuah proses normalisasi yang lebih lanjut pada gambar 7.5.

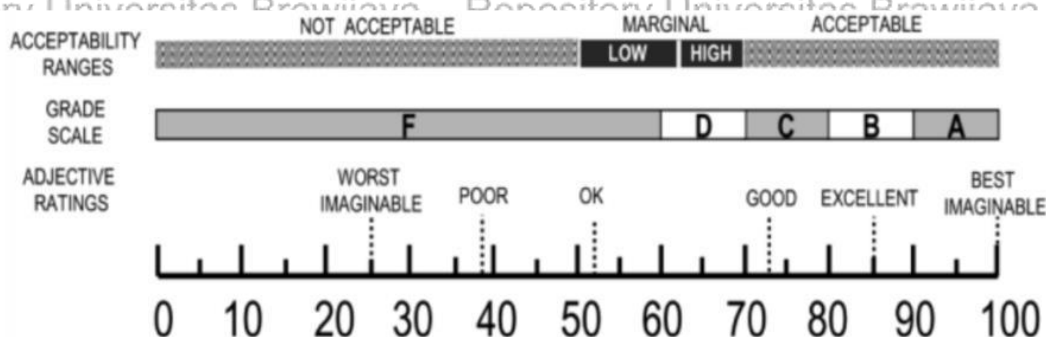


Gambar 7.5 presentil skor SUS

(Sumber : Brooke,2011)

SUS memiliki validitas yang mengacu pada seberapa bagik suatu object dapat mengukur apa yang ingin dimaksudkan untuk diukur, dalam hal ini apa yang dirasakan oleh pengguna. SUS merupakan metode yang telah terbukti efektif dalam membedakan sistem yang dapat digunakan serta sistem yang tidak dapat digunakan, SUS juga memiliki kolerasi yang tinggi dengan pengukuran yang berbasis kuisioner lainnya.

Pada tahap selanjutnya kemudian dilakukan sebuah analisis , analisis tersebut berdasarkan pada hasil pertanyaan-pertanyaan yang telah dijawab oleh responden. Dari jawaban-jawaban yang telah diberikan oleh responden dapat diketahui nilai akhir yang kemudian memberikan hasil skala *likert* dengan menggunakan metode SUS. Dari hasil tersebut kemudian dapat disamakan dengan tingkat skala konversi SUS yang kemudian bisa diamati pada Gambar 7.6.



Gambar 7.6 Rating dan Skala Konversi SUS

(Sumber : Brooke,2011)



Tabel 7.34 hasil kuisioner SUS

No	Nama		
	Alliya ilmi	Agung Cahya	Oktaviani dwi
1	4	4	5
2	2	2	3
3	4	4	3
4	1	2	2
5	4	5	5
6	2	2	2
7	3	4	5
8	2	2	2
9	3	4	4
10	2	2	2

Tabel 7.35 hasil perhitungan skor SUS

No	Nama		
	Alliya ilmi	Agung Cahya	Oktaviani dwi
1	3	3	4
2	3	3	2
3	3	3	2
4	4	3	3
5	3	4	4
6	3	3	3
7	2	3	4
8	3	3	3
9	2	3	3
10	3	3	3
Total	29	31	31
Total x 2.5	72,5	77,5	77,5
Rata-rata		75,8	



Pada hasil pengujian tersebut mendapatkan hasil rata-rata yaitu 75,8. Berdasar hitungan tersebut didapat hasil total *likert*. Yang apabila dikonversi menjadi skala SUS oleh karena itu sistem ini masuk ke dalam kategori *acceptable* atau sistem ini telah mencapai kelayakan pengguna.

BAB 8 PENUTUP

8.1 Kesimpulan

Dari berbagai tahapan yang kemudian telah dilewati seperti fase analisis kebutuhan, selanjutnya perancangan yang kemudian dilakukan implementasi dan pada akhirnya dilakukan pengujian, dapat ditarik berbagai kesimpulan sebagai berikut :

1. Pada analisis kebutuhan diperoleh yaitu terdapat 23 kebutuhan fungsional yang telah teridentifikasi dengan tambahan 1 kebutuhan kategori non fungsional. Kemudian untuk aktor yang telah teridentifikasi terdapat 3 aktor yaitu, admin, user serta pengunjung. Sementara sumber dari analisis kebutuhan yang didapat berdasarkan hasil dari wawancara sebelumnya.
2. Kemudian pada tahapan perancangan diantaranya menggunakan model perancangan yang diantaranya *sequence diagram* , *use case diagram*, perancangan komponen, perancangan antarmuka, perancangan database yang memodelkan firebase dan *usecase scenario*. Keseluruhan model bertujuan memberikan gambaran mengenai jalannya sebuah sistem.
3. Berdasar dari hasil perancangan yang kemudian telah dijalankan sebelumnya dilakukan sebuah implementasi yang diantaranya implementasi database, implementasi kode program, implementasi antarmuka serta implementasi spesifikasi sistem.
4. Tahapan akhir dari seluruh tahapan yaitu pengujian, dari sebuah pengujian diperoleh hasil berupa tingkat hasil valid 100% yang terdiri diantaranya pengujian unit, pengujian integrasi serta pengujian validasi. Berikutnya untuk pengujian *usability* hasil skor akhir yang diperoleh yaitu 75,8 yang masuk dalam kategori *acceptable* yang artinya sudah sesuai dengan apa yang pengguna butuhkan.

8.2 Saran

1. Lebih membuat aplikasi lebih dinamis agar dapat menambahkan soal berupa gambar – gambar yang lebih menghidupkan soal-soal agar lebih menarik dan lebih mengasah kemampuan.
2. Menambahkan fitur pendataan tentang jumlah disabilitas yang berada disuatu kampus dan berbagai kategorinya serta sebarannya sehingga pengguna dapat mengetahui apakah pada fakultas masing-masing terdapat mahasiswa dengan kebutuhan khusus atau tidak.



DAFTAR REFERENSI

- A.S, R., & Shalahuddin, M. (2015). Analisis Berorientasi Objek. Dalam R. A.S (Penyunt.), *Rekayasa Perangkat Lunak* (hal. 114). Bandung: Informatika Bandung.
- Andre. (2019). *utorial Belajar Java Part 1: Pengertian Bahasa Pemrograman Java*. Dipetik january 12, 2020, dari <https://www.duniailkom.com/tutorial-belajar-java-pengertian-bahasa-pemrograman-java/>
- Brooke, J. (2011). *MEASURING USABILITY WITH THE SYSTEM USABILITY SCALE (SUS)*. Dipetik May 4, 2020, dari <https://measuringu.com/sus/>
- Clark, D. (2013). *Begining C# Object Oriented Programming* (2nd ed.). New York: Paul Manning.
- Komputerdia. (2019). *Pengenalan, Pengertian dan Memahami Android Studio Lebih Dekat*. Dipetik january 1, 2020, dari <https://www.komputerdia.com/2017/08/pengenalan-dan-memahami-android-studio-lebih-dekat.html>
- Pratiwi, A., Lintangari, A. P. Rizky, U. F., & Rahajeng, U. W. (2018). *disabilitas dan pendidikan inklusif di perguruan tinggi* (Tim UB Press ed.). malang: UB Press.
- PSLD. (2013, july 3). *Sejarah PSLD*. Diambil kembali dari [psld.ub.ac.id:](http://psld.ub.ac.id/) <http://psld.ub.ac.id/in/profil/sejarah-psld/>
- Rahmayani, I. (2015). *Indonesia Raksasa Teknologi Digital Asia*. Dipetik January 13, 2020, dari https://kominfo.go.id/content/detail/6095/indonesia-raksasa-teknologi-digital-asia/0/sorotan_media
- Riadi, M. (2018). *Pengertian, Jenis dan Hak Penyandang Disabilitas*. Dipetik January 13, 2020, dari <https://www.kajianpustaka.com/2018/07/pengertian-jenis-dan-hak-penyandang-disabilitas.html>
- SAFAAT H, N. (2011). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: INFORMATIKA.
- Simarmata, J. (2010). Pengertian Rekayasa Perangkat Lunak. Dalam N. WK (Penyunt.), *Rekayasa Perangkat Lunak* (hal. 10). Yogyakarta: C.V ANDI OFFSET (Penerbit Andi).
- Smjeh. (2012, juny 29). *Anatomy Physiology of an Android*. Diambil kembali dari [androidteam.googlecode.com:](http://androidteam.googlecode.com/) <http://androidteam.googlecode.com/files/Anatomy-Physiology-of-an-Android.pdf>
- Sommerville, I. (2016). *Software Engineering* (10th edition ed.). USA: Pearson Education Limited.



Lampiran A Hasil wawancara

Narasumber : Alies Poetri Lintang Sari

Posisi : Kepala Bidang Pelayanan

1. Pengertian dari pendamping?

Pendamping merupakan seorang individu yang mendampingi mahasiswa-mahasiswa berkebutuhan khusus atau biasa yang disebut dengan mahasiswa difabel yang tugasnya adalah mendampingi proses pembelajaran atau proses akademik lainnya berjalan dengan seperti yang seharusnya. Pada jenjang anak-anak pendamping juga biasa disebut dengan istilah *shadow teacher* atau guru bayangan yang menemani dan mengarahkan mereka ketika belajar.

2. Alur pelatihan untuk menjadi pendamping di PS LD UB ?

Untuk menjadi pendamping di PS LD UB yang pertama dengan mendaftarkan terlebih dahulu melalui form pembukaan pendamping online yang telah disediakan, kemudian peserta yang terpilih akan diinformasikan lebih lanjut melalui sosial media untuk kegiatan test selanjutnya untuk menjadi pendamping.

3. Pelatihan-pelatihan apa saja yang diberikan kepada calon pendamping ?

Untuk pembukaan diawal berupa pengenalan apa itu PS LD UB kemudian dijelaskan struktur mengenai PS LD UB, kemudian calon pendamping dikenalkan mengenai pentingnya menumbuhkan kesadaran disabilitas atau Disability Awareness. Kemudian akan diberikan materi-materi pelatihan mahasiswa difabel dalam proses belajar dan proses akademik lainnya sesuai dengan kategori disabilitas, kemudian disertai dengan pelatihan langsung seperti pelatihan Bahasa isyarat dan pelatihan menggunakan kursi roda.

4. Bagaimana cara menentukan pendamping itu diterima sebagai pendamping ?

Untuk penentuan diterimanya sebagai pendamping atau tidak yaitu dengan cara evaluasi dari hasil praktek pelatihan-pelatihan mendampingi mahasiswa difabel berdasarkan kategori dan diakhiri dengan ujian tertulis yang berupa soal-soal yang berkaitan dengan materi yang telah diberikan sebelumnya dari hari pertama.

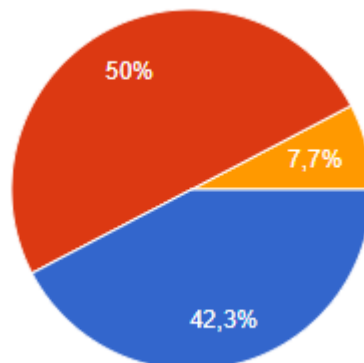


Lampiran B Hasil Survey Penguasaan Bahasa Isyarat

Survey ini bertujuan untuk mengetahui tingkat penguasaan para pendamping di PSKD UB mengenai praktik penggunaan Bahasa isyarat. Survey dilakukan melibatkan 26 responden dengan angkatan yang terdiri dari 2015,2016,2017,2018 dengan rata-rata sudah mendampingi selama 6 bulan. Dengan diklasifikasikan menjadi 3 kategori yaitu kurang,cukup dan baik dan berikut persebaran data dapat dilihat pada Tabel 0.1.

Tabel 0.1 hasil survey Bahasa isyarat

No	kategori	Presentase
1	kurang	42,3%
2	cukup	50 %
3	Baik	7,7%



Gambar 0.1 bagan presentase penguasaan Bahasa isyarat.

Dari hasil tersebut dapat ditarik kesimpulan masi banyaknya mahasiswa/I yang bertindak sebagai pendamping kurang baik dalam penggunaan Bahasa isyarat sebagai praktik pendampingan,sehingga diperlukan sebuah pelatihan lebih lanjut.