

**PENGEMBANGAN SISTEM INFORMASI ASESMEN LEMBAGA
SERTIFIKASI PROFESI BERBASIS WEB
(STUDI KASUS: SMK NEGERI 4 MALANG)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Pendidikan

Disusun oleh:
Arif Rahmadani Vinanda
NIM: 156150601111012



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

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI ASESMEN LEMBAGA SERTIFIKASI PROFESI
BERBASIS WEB (STUDI KASUS: SMK NEGERI 4 MALANG)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Pendidikan

Disusun Oleh :
Arif Rahmadani Vinanda
NIM: 156150601111012

Skripsi ini telah diuji dan dinyatakan lulus pada
12 Juli 2019

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II

Satrio Agung Wicaksono, S.Kom., M.Kom.
NIP: 19860521 201212 1 001

Faizatul Amalia, S.Pd., M.Pd.
NIK: 201309 860821 2 001

Mengetahui
Ketua Jurusan Sistem Informasi

Dr. Eng. Herman Tolle, S.T., M.T.
NIP: 19740823 200012 1 001

PERNYATAAN ORISINALITAS

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

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 Juli 2019

Arif Rahmadani Vinanda

NIM: 156150601111012



PRAKATA

Puji syukur yang dapat penulis panjatkan atas kehadiran Allah Subhanahu Wa Ta'ala, karena berkat Rahmat dan Karunia-Nya, penulis dapat menyelesaikan laporan skripsi yang berjudul "PENGEMBANGAN SISTEM INFORMASI ASESMEN LEMBAGA SERTIFIKASI PROFESI BERBASIS WEB (STUDI KASUS: SMK NEGERI 4 MALANG)". Tidak lupa juga untuk menghaturkan Shalawat serta salam yang penulis panjatkan pada junjungan besar Nabi Muhammad Salallahu Alaihi Wa salam beserta keluarga, sahabat dan pengikutnya.

Penyusunan laporan ini tidak akan berhasil jika tidak ada bantuan dari berbagai pihak. Pada bagian ini, penulis ingin menghaturkan terima kasih dan rasa hormat kepada berbagai pihak yang telah membantu dalam pengerjaan skripsi ini kepada:

1. Bapak Satrio Agung Wicaksono, S.Kom., M.Kom. selaku pembimbing I yang selalu dengan senang hati dan sabar dalam memberikan arahan, masukan, saran dukungan, nasihat, dan motivasi dalam pengerjaan skripsi ini. Serta penulis banyak belajar hal-hal baru selama proses pengerjaan skripsi kepada beliau.
2. Ibu Faizatul Amalia, S.Pd., M.Pd. selaku pembimbing II yang selalu memberikan arahan, pengetahuan baru dan candaan yang membawa suasana gembira dan santai meski tetap serius dan sabar selama proses bimbingan dan pengerjaan skripsi ini, serta penulis dapat belajar banyak hal baru selama menimba ilmu kepada beliau.
3. Bapak M. Mahmudi selaku salah satu asesor Lembaga Sertifikasi Profesi SMK Negeri 4 Malang yang telah memberikan waktu dan kesempatan dalam proses wawancara terkait pengembangan sistem informasi.
4. Bapak M. Asrofi, M.Pd. selaku Wakil Kepala Sekolah SMK Negeri 4 Malang Bagian Humas yang telah memberikan bantuan baik dalam bentuk motivasi hingga izin dalam proses pengambilan data pada SMK Negeri 4 Malang.
5. Bapak, Ibu dan Keluarga yang dengan tulus selalu memberi doa, dukungan baik berupa materil maupun non materil, nasihat untuk bagaimana sabar dan tetap berusaha serta motivasi yang tiada henti untuk memberikan semangat kepada penulis dalam pengerjaan skripsi ini hingga selesai.
6. Teman-teman Pendidikan Teknologi Informasi angkatan 2015 dan angkatan setelahnya yang penulis tidak bisa sebutkan satu-persatu yang telah membantu penulis baik secara langsung ataupun tidak langsung.
7. Karyawan dan staf Fakultas Ilmu Komputer Universitas Brawijaya Malang yang membantu penulis dalam proses skripsi ini.
8. Serta semua pihak yang telah terlibat dalam pengerjaan skripsi ini yang tidak bisa disebutkan penulis satu persatu.

Semoga Allah Subhanahu Wa Ta'ala memberikan balasan berlipat ganda kepada pihak-pihak yang telah membantu penulis yang berpartisipasi dalam menyelesaikan skripsi ini. Penulis menyadari bahwa skripsi ini masih jauh dari kata "sempurna". Yang benar datangnya dari Allah Subhanahu Wa Ta'ala. Yang salah datangnya dari penulis sendiri sebagaimana manusia yang hakikatnya tidak luput dari kesalahan dan lupa. Akhir kata, semoga laporan skripsi ini dapat bermanfaat bagi lingkungan sekitar yang menggunakannya. Amin.

Malang, 12 Juli 2019

Penulis

arifrahmadanivinanda@gmail.com



ABSTRAK

Arif Rahmadani Vinanda, Pengembangan Sistem Informasi Asesmen Lembaga Sertifikasi Profesi Berbasis Web (Studi Kasus: SMK Negeri 4 Malang)

Pembimbing: Satrio Agung Wicaksono, S.Kom., M.Kom. dan Faizatul Amalia, S.Pd., M.Pd.

SMK Negeri 4 Malang salah satu Lembaga yang mendapatkan lisensi sebagai penyelenggara sertifikasi profesi untuk siswa SMK dari Badan Nasional Sertifikasi Profesi (BNSP). Pada proses sertifikasi profesi terdapat dokumen yang harus diisi oleh asesor dan siswa yang berjumlah lebih dari 50 (lima puluh) lembar halaman untuk satu siswa. Hal tersebut menimbulkan masalah yakni waktu pengisian, penyimpanan, dan penggunaan kertas yang berlebih. Penelitian menggunakan *System Development Life Cycle (SDLC) Waterfall* tanpa melakukan proses *deployment*. Untuk metode pengambilan data menggunakan metode wawancara dan observasi pada dokumen fisik asesmen pada jurusan Rekayasa Perangkat Lunak. Dari hasil wawancara dan observasi didapatkan kebutuhan perangkat lunak bagi dua aktor atau pengguna yakni siswa dan asesor yang selanjutnya digambarkan pada beberapa diagram *unified modeling language* sebelum dilakukan proses implementasi menggunakan bahasa pemrograman PHP dan menggunakan *framework CodeIgniter*. Untuk proses pengujian pertama menggunakan metode *black-box* dengan teknik *error guessing* yang menguji kemungkinan error yang akan terjadi pada sistem. Hasil yang didapatkan ialah seluruh kemungkinan error telah dapat tertangani dengan baik ditunjukkan dengan seluruh pengujian mendapatkan hasil valid. Untuk pengujian kedua menggunakan metode *user acceptance test* yakni melihat penerimaan sistem oleh pengguna menggunakan kuisisioner yang diberikan. Hasil yang didapatkan ialah seluruh responden memberikan jawaban “ya” pada setiap pertanyaan yang diberikan. Dari hasil tersebut dilakukan proses *acceptance decision* atau pengambilan keputusan terkait penerimaan menunjukkan bahwa seluruh fungsional atau dari segi *usefulness* sistem dapat diterima secara penuh oleh pengguna baik siswa maupun asesor.

Kata kunci: Lembaga Sertifikasi Profesi, *SDLC Waterfall*, Asesmen Mandiri, *CodeIgniter*, *Error Guessing*, *User Acceptance Test*

ABSTRACT

Arif Rahmadani Vinanda, Pengembangan Sistem Informasi Asesmen Lembaga Sertifikasi Profesi Berbasis Web (Studi Kasus: SMK Negeri 4 Malang)

Supervisors: Satrio Agung Wicaksono, S.Kom., M.Kom. dan Faizatul Amalia, S.Pd., M.Pd.

SMK Negeri 4 Malang is one of the Institutions that gets a license as an organizer of professional certification for vocational students from National Professional Certification Agency (BNSP). In the professional certification processes there are documents that must be filled in by the assessor and students totaling more than 50 (fifty) pages for one student. This causes problems, filling process time, documents storage, and the use of excessive paper. In this research uses the Waterfall System Development Life Cycle (SDLC) without deployment process. For data collection methods using interview and observation methods on physical assessment documents in Rekayasa Perangkat Lunak study. From the results of interviews and observations, the software requirements are obtained for both actors (student and assessor), and then illustrated on several diagram of a unified modeling language before the implementation process using the PHP programming language and using the CodeIgniter framework. For the first testing process using the black-box method with an error guessing technique to get error possibility that will occur in the system. The results obtained are all possible errors can be handled properly indicated by all tests get valid results. For the second test, the user acceptance test method is used to see the acceptance of the system by the user using the questionnaire. The results obtained were all respondents gave "yes" answers to each question given. From these results an acceptance decision process for decision making related to acceptance shows that all functionalities or in terms of the usefulness of the system can be fully accepted by users, both students and assessors.

Keywords: Professional Certification Institution, SDLC Waterfall, Self Assessment, CodeIgniter, Error Guessing, User Acceptance Test

DAFTAR ISI

PENGESAHAN	i
PERNYATAAN ORISINALITAS	ii
PRAKATA	iii
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	4
1.4 Manfaat	4
1.5 Batasan Masalah	4
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Penelitian Terkait	6
2.2 Sistem Informasi	7
2.3 SMK Negeri 4 Malang	7
2.4 Asesmen	8
2.5 Lembaga Sertifikasi Profesi (LSP)	8
2.6 Badan Nasional Sertifikasi Profesi (BNSP)	8
2.7 Rekayasa Perangkat Lunak	9
2.8 <i>System Development Life Cycle (SDLC)</i>	9
2.9 <i>Waterfall Model</i>	9
2.9.1 <i>Communication</i>	10
2.9.2 <i>Planning</i>	10
2.9.3 <i>Modeling</i>	10
2.9.4 <i>Construction</i>	21

BAB 3 METODOLOGI	25
3.1 Studi Literatur	26
3.2 <i>Communication</i>	26
3.3 <i>Planning</i>	27
3.4 <i>Modeling</i>	27
3.5 <i>Construction</i>	27
3.6 Pengambilan Kesimpulan dan Saran	28
BAB 4 HASIL DAN PEMBAHASAN	29
4.1 Analisis Kebutuhan	29
4.1.1 Analisis Permasalahan	29
4.1.2 Identifikasi Pengguna	29
4.1.3 Daftar Spesifikasi Kebutuhan	30
4.1.4 Pemodelan <i>Use Case</i>	31
4.1.5 <i>Activity Diagram</i>	42
4.2 Perancangan Sistem	47
4.2.1 <i>Collaboration Diagram</i>	47
4.2.2 <i>Sequence Diagram</i>	50
4.2.3 <i>Class Diagram</i>	56
4.2.4 <i>Physical Data Model</i>	61
4.2.5 Perancangan Antarmuka	64
4.3 Implementasi	67
4.3.1 Implementasi Antarmuka	67
4.3.2 Implementasi Fungsi	73
4.3.3 Implementasi <i>Database</i>	89
4.4 Pengujian dan Analisis	96
4.4.1 Pengujian <i>Black-box</i>	96
4.4.2 Analisis Hasil Pengujian <i>Black-box</i>	102
4.4.3 Pengujian <i>User Acceptance Testing</i>	102
4.4.4 Analisis Hasil Pengujian <i>User Acceptance Testing</i>	105
BAB 5 PENUTUP	106
5.1 Kesimpulan	106
5.2 Saran	107

DAFTAR REFERENSI.....	108
LAMPIRAN.....	111



DAFTAR TABEL

Tabel 2.1 Simbol <i>Use Case Diagram</i>	11
Tabel 2.2 Simbol <i>Activity Diagram</i>	12
Tabel 2.3 Simbol <i>Collaboration Diagram</i>	15
Tabel 2.4 Simbol <i>Sequence Diagram</i>	16
Tabel 2.5 Simbol <i>Class Diagram</i>	20
Tabel 2.6 Contoh Kuesioner <i>User Acceptance Testing</i> (Evantio, 2017)	24
Tabel 4.1 Identifikasi Kebutuhan Fungsional	30
Tabel 4.2 <i>Use Case Scenario</i> Mendaftar Sertifikasi Profesi	32
Tabel 4.3 <i>Use Case Scenario</i> Mengunggah Berkas Sertifikasi	33
Tabel 4.4 <i>Use Case Scenario</i> Melakukan Proses Asesmen Mandiri	34
Tabel 4.5 <i>Use Case Scenario</i> Melihat Daftar Pemohon Sertifikasi	36
Tabel 4.6 <i>Use Case Scenario</i> Menilai Asesmen Mandiri Siswa	36
Tabel 4.7 <i>Use Case Scenario</i> Mencetak Hasil Asesmen Mandiri	38
Tabel 4.8 <i>Use Case Scenario</i> Mencari Hasil Asesmen Mandiri	39
Tabel 4.9 <i>Use Case Scenario</i> Login	40
Tabel 4.10 <i>Use Case Scenario</i> Logout	41
Tabel 4.11 <i>Use Case Scenario</i> Mencari Hasil Asesmen Mandiri	41
Tabel 4.12 Daftar Kegunaan Tabel	61
Tabel 4.13 Implementasi <i>Controller</i> Siswa.php	73
Tabel 4.14 Implementasi <i>Model</i> siswa_model.php	76
Tabel 4.15 Implementasi <i>Model</i> unit_model.php	76
Tabel 4.16 Implementasi <i>Model</i> elemen_model.php	77
Tabel 4.17 Implementasi <i>Model</i> pertanyaan_model.php	77
Tabel 4.18 Implementasi <i>Model</i> asesmen_mandiri_model.php	77
Tabel 4.19 Implementasi <i>Controller</i> Asesor.php	78
Tabel 4.20 Implementasi <i>Model</i> siswa_model.php	80
Tabel 4.21 Implementasi <i>Controller</i> Asesor.php	82
Tabel 4.22 Implementasi <i>Model</i> asesmen_mandiri_model.php	85
Tabel 4.23 Implementasi <i>Controller</i> Asesor.php	85
Tabel 4.24 Implementasi <i>Model</i> siswa_model.php	87

Tabel 4.25 Implementasi <i>Model</i> dokumen_model.php	88
Tabel 4.26 Implementasi <i>Model</i> dokumen_model.php	88
Tabel 4.27 Implementasi <i>Model</i> asesmen_mandiri_model.php	88
Tabel 4.28 Implementasi Pembuatan <i>Database</i>	89
Tabel 4.29 Implementasi Pembuatan Tabel <i>Provinces</i>	89
Tabel 4.30 Implementasi Pembuatan Tabel Level	89
Tabel 4.31 Implementasi Pembuatan Tabel Status	90
Tabel 4.32 Implementasi Pembuatan Tabel <i>Form</i>	90
Tabel 4.33 Implementasi Pembuatan Tabel Program Keahlian	90
Tabel 4.34 Implementasi Pembuatan Tabel Kartu Identitas	91
Tabel 4.35 Implementasi Pembuatan Tabel <i>Regencies</i>	91
Tabel 4.36 Implementasi Pembuatan Tabel Skema	92
Tabel 4.37 Implementasi Pembuatan Tabel Unit	92
Tabel 4.38 Implementasi Pembuatan Tabel Elemen	93
Tabel 4.39 Implementasi Pembuatan Tabel Pertanyaan	93
Tabel 4.40 Implementasi Pembuatan Tabel <i>User</i>	94
Tabel 4.41 Implementasi Pembuatan Tabel Asesor	94
Tabel 4.42 Implementasi Pembuatan Tabel Siswa	95
Tabel 4.43 Implementasi Pembuatan Tabel Dokumen	95
Tabel 4.44 Implementasi Pembuatan Tabel Asesmen Mandiri	96
Tabel 4.45 <i>Error Guessing</i> Melakukan Proses Asesmen Mandiri	98
Tabel 4.46 <i>Error Guessing</i> Melihat Daftar Pemohon Sertifikasi	99
Tabel 4.47 <i>Error Guessing</i> Menilai Asesmen Mandiri Siswa	100
Tabel 4.48 <i>Error Guessing</i> Mencari Hasil Asesmen Mandiri	102
Tabel 4.49 Tabel Hasil <i>User Acceptance Testing</i>	103

DAFTAR GAMBAR

Gambar 2.1 <i>Waterfall Model</i> (Pressman, 2015)	9
Gambar 2.2 Contoh <i>Use Case Diagram E-Complaint Management</i> (Evantion, 2017)	12
Gambar 2.3 Contoh <i>Activity Diagram</i> Menyelesaikan Pekerjaan Komplain (Evantio, 2017)	14
Gambar 2.4 Contoh <i>Collaboration Diagram</i> Menerima Pekerjaan Komplain (Evantio, 2017)	15
Gambar 2.5 Unifikasi Kelas Analisis <i>e-complaint management</i> (Evantio, 2017) ..	16
Gambar 2.6 Contoh <i>Sequence Diagram</i> Mengubah Data Kebutuhan Pekerjaan Komplain (Evantio, 2017)	18
Gambar 2.7 Contoh <i>Class Diagram</i> Universitas (Tong, 2017)	19
Gambar 2.8 Contoh <i>Physical Data Model</i> (Fowler & Stanwick, 2004)	21
Gambar 3.1 Langkah-langkah Pemecahan Masalah	25
Gambar 4.1 Aturan Penomoran Kebutuhan Fungsional.....	30
Gambar 4.2 <i>Use Case</i> Sistem Informasi Asesmen LSP	32
Gambar 4.3 <i>Activity Diagram</i> Melakukan Proses Asesmen Mandiri	43
Gambar 4.4 <i>Activity Diagram</i> Melihat Daftar Pemohon Sertifikasi	44
Gambar 4.5 <i>Activity Diagram</i> Menilai Asesmen Mandiri Siswa	45
Gambar 4.6 <i>Activity Diagram</i> Mencari Hasil Asesmen	46
Gambar 4.7 <i>Collaboration Diagram</i> Melakukan Proses Asesmen Mandiri	47
Gambar 4.8 <i>Collaboration Diagram</i> Melihat Daftar Pemohon Sertifikasi	48
Gambar 4.9 <i>Collaboration Diagram</i> Menilai Asesmen Mandiri Siswa	48
Gambar 4.10 <i>Collaboration Diagram</i> Mencari Hasil Asesmen Mandiri	49
Gambar 4.11 Diagram Unifikasi Kelas Analisis	50
Gambar 4.12 <i>Sequence Diagram</i> Melakukan Proses Asesmen Mandiri.....	51
Gambar 4.13 <i>Sequence Diagram</i> Melihat Daftar Pemohon Sertifikasi	52
Gambar 4.14 <i>Sequence Diagram</i> Menilai Asesmen Mandiri Siswa	53
Gambar 4.15 <i>Sequence Diagram</i> Mencari Hasil Asesmen Mandiri	55
Gambar 4.16 <i>Class Diagram Controller</i>	56
Gambar 4.17 <i>Class Diagram Model Awal</i>	57
Gambar 4.18 <i>Class Diagram Model</i> Setelah Normalisasi	58

Gambar 4.19 Unifikasi <i>Class Diagram</i>	60
Gambar 4.20 <i>Physical Data Model</i>	63
Gambar 4.21 Rancangan Antarmuka Beranda Siswa	64
Gambar 4.22 Rancangan Antarmuka Asesmen Mandiri.....	64
Gambar 4.23 Rancangan Antarmuka Beranda Asesor.....	65
Gambar 4.24 Rancangan Antarmuka Daftar Siswa	66
Gambar 4.25 Rancangan Antarmuka Detail <i>User</i>	66
Gambar 4.26 Rancangan Antarmuka Asesmen Mandiri (Asesor)	67
Gambar 4.27 Implementasi Antarmuka Beranda Siswa	68
Gambar 4.28 Implementasi Antarmuka Asesmen Mandiri	68
Gambar 4.29 Implementasi Antarmuka Beranda Asesor	69
Gambar 4.30 Implementasi Antarmuka Daftar Siswa	69
Gambar 4.31 Implementasi Antarmuka Detail <i>User</i>	70
Gambar 4.32 Implementasi Antarmuka Penilaian Asesmen Mandiri Siswa	71
Gambar 4.33 Implementasi Antarmuka Mencari Hasil Asesmen Mandiri	72
Gambar 4.34 Implementasi Antarmuka Mencetak Dokumen.....	72
Gambar 4.35 Contoh <i>Form</i> Melakukan Proses Asesmen Mandiri.....	97
Gambar 4.36 Contoh <i>Form</i> Melihat Daftar Pemohon Sertifikasi.....	98
Gambar 4.37 Contoh <i>Form</i> Menilai Asesmen Mandiri Siswa	100
Gambar 4.38 Contoh <i>Form</i> Mencari Hasil Asesmen Mandiri	101

DAFTAR LAMPIRAN

LAMPIRAN A CONTOH DOKUMEN ASESMEN LSP	111
LAMPIRAN B KERANGKA WAWANCARA	127
LAMPIRAN C HASIL WAWANCARA	129
LAMPIRAN D <i>USER ACCEPTANCE TESTING</i> SISWA	131
LAMPIRAN E <i>USER ACCEPTANCE TESTING</i> ASESOR	133



BAB 1 PENDAHULUAN

1.1 Latar belakang

Pada saat ini, SMK Negeri 4 Malang merupakan salah satu sekolah menengah kejuruan yang telah mempunyai lisensi sebagai Lembaga Sertifikasi Profesi Pihak Pertama (LSP-P1). Menurut Kementerian Pendidikan dan Kebudayaan (2017), Lembaga Sertifikasi Profesi (LSP) merupakan lembaga pelaksana kegiatan kompetensi kerja yang mendapatkan lisensi dari Badan Nasional Sertifikasi Profesi (BNSP). Berdasarkan jenisnya, LSP dibagi menjadi tiga yakni LSP pihak pertama, LSP pihak kedua dan LSP pihak ketiga. Dengan adanya sertifikasi profesi tersebut, dimaksudkan agar lulusan SMK mempunyai sertifikat untuk modal dalam melamar pekerjaan atau pertimbangan dalam perekrutan kerja oleh perusahaan.

Untuk mendapatkan sertifikat Lembaga Sertifikasi Profesi dari Badan Nasional Sertifikasi profesi, instansi yang telah mendapatkan lisensi Lembaga Sertifikasi Profesi perlu melakukan proses asesmen pada pemohon sertifikat (dalam hal ini siswa SMK Negeri 4 Malang). Pada proses penyelenggaraan asesmen, asesor adalah guru-guru terpilih yang telah melalui pelatihan dan telah mendapatkan lisensi sebagai asesor dari Badan Nasional Sertifikasi Profesi yang akan menyelenggarakan asesmen pada siswa dengan metode pengisian manual pada daftar kriteria atau kompetensi yang disyaratkan pada masing-masing siswa. Selain memuat hal tersebut, pada dokumen asesmen memuat asesmen mandiri, biodata siswa dan kesimpulan dari asesor terhadap hasil asesmen yang dilakukan.

Proses asesmen dilakukan dalam beberapa tahap, diantaranya pengisian biodata oleh siswa pemohon sertifikat dan melengkapi dokumen transkrip rapor siswa. Setelah proses tersebut, siswa mengisi lembar asesmen mandiri yakni siswa akan menilai dirinya sendiri dalam hal kecukupan dokumen maupun kompetensi dasar yang dimiliki. Setelah itu, asesor akan melakukan pengecekan pada lembar asesmen mandiri siswa apakah data yang dimasukkan siswa telah valid dan memenuhi syarat. Kemudian siswa akan mengerjakan soal praktik yang telah disediakan oleh asesor berdasarkan kompetensi yang disyaratkan. Setelah siswa selesai mengerjakan soal praktik yang disediakan, asesor akan mulai melakukan proses asesmen pada hasil pekerjaan siswa untuk menilai kecukupan kompetensi siswa. Kecukupan yang disyaratkan terdapat pada masing-masing dokumen unit kompetensi yang didalamnya terdapat elemen kompetensi, kriteria unjuk kerja, batasan variabel, panduan penilaian, dan kompetensi kunci.

Dari hasil observasi yang dilakukan pada jurusan Rekayasa Perangkat Lunak (RPL) SMK Negeri 4 Malang menemukan bahwa dokumen asesmen satu siswa kurang lebih berjumlah 50 lembar kertas. Berdasarkan hasil wawancara dengan salah satu guru SMK Negeri 4 Malang pada jurusan Rekayasa Perangkat Lunak (RPL) yakni Bapak M. Mahmudi (2018), beliau menjelaskan bahwa berdasarkan jumlah dokumen yang mencapai lebih dari 50 lembar kertas untuk satu orang siswa akan menimbulkan masalah yakni proses pengisian dokumen yang

memakan waktu serta penyimpanan dokumen fisik yang sangat banyak pada ruangan yang terbatas (lampiran C). Untuk masalah waktu yang digunakan untuk proses pengisian lembar asesmen pada masing-masing siswa bersumber pada proses pengisian asesmen yang memerlukan panduan dalam mengisi asesmen tersebut, sedangkan panduan untuk mengisi asesmen yang dimaksud terletak pada lembar yang berbeda dengan lembar isian asesmen. Hal tersebut menyebabkan bertambahnya waktu dalam proses asesmen karena asesor akan membuka kembali halaman sebelumnya hanya untuk melihat panduan asesmen. Sedangkan untuk masalah penyimpanan dokumen, guru akan menyimpan dokumen hasil asesmen meskipun siswa tersebut telah mendapatkan sertifikat profesi. Jika dalam satu kelas terdapat 40 (empat puluh) siswa dan terdapat 3 (tiga) kelas dalam jurusan Rekayasa Perangkat Lunak, maka dalam satu kali proses asesmen akan menghasilkan kurang lebih 120 (seratus dua puluh) dokumen asesmen dan jika diambil titik paling minimal dalam jumlah lembar 1 (satu) dokumen asesmen yakni 50 (lima puluh) lembar kertas, maka dalam satu kali proses asesmen akan menghasilkan 6000 (enam ribu) lembar kertas yang digunakan atau 12 (dua belas) rim. Hal tersebut mengakibatkan penyimpanan dokumen yang dapat memakan *space* atau tempat dalam ruangan yang terbatas.

Menurut Ariawan (2009), dalam Buku Ajar Sistem Informasi Manajemen, sistem informasi adalah sebagai kumpulan/grup dari subsistem/bagian/komponen apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berarti dan berguna. Berdasarkan pengertian tersebut, penulis memahami bahwa dalam proses memecahkan masalah dalam penjabaran masalah di atas dapat menggunakan suatu sistem informasi sehingga dari pengembangan sistem informasi ini dapat mengambil data dari proses asesmen dan dapat menjadi informasi yang berguna untuk digunakan dalam pertimbangan siswa dapat memperoleh sertifikat profesi.

Berdasarkan wawancara dengan bapak M. Mahmudi (2018) bahwa pada SMK Negeri 4 Malang telah terdapat akses internet yang mudah dengan adanya teknologi *WiFi* (lampiran C). Dengan mudahnya akses internet pada sekolah tersebut dapat berpengaruh dalam hal manajemen pendidikan utamanya pada proses asesmen LSP yang dapat diselesaikan dengan adanya sistem informasi asesmen Lembaga Sertifikasi Profesi ini. Selain alasan tersebut, digunakannya sistem informasi berbasis web dimaksudkan karena sifat dari pengembangan sistem informasi asesmen Lembaga Sertifikasi Profesi masih dalam tahap pengembangan. Dalam tahapan ini berfungsi menghasilkan masukan-masukan seperti *platform* yang cocok untuk digunakan dalam proses asesmen. Mengingat *web based application* dapat diakses dari beberapa *platform* yakni *desktop*, *mobile*, dan sebagainya. Tidak menutup kemungkinan bahwa sistem informasi asesmen Lembaga Sertifikasi Profesi yang dibuat akan merambah pada aplikasi *native* seperti aplikasi *desktop*, *android*, *iOS*, dan lain sebagainya. Pada proses pengembangan sistem informasi asesmen Lembaga Sertifikasi Profesi ini

menggunakan *SDLC (System Development Life Cycle) Waterfall*. Penggunaan *SDLC Waterfall* digunakan dalam proses pengembangan sistem informasi asesmen Lembaga Sertifikasi Profesi karena kebutuhan fungsional sistem telah dapat didefinisikan pada awal perancangan sistem informasi yang akan dibuat. Oleh karena itu, pada proses pengembangan sistem informasi ini akan dilakukan secara berurutan mulai dari awal hingga akhir dan tidak ada proses mengulang apabila terdapat perubahan kebutuhan sistem pada proses pengembangan sistem informasi karena kebutuhan sistem telah disepakati dari awal perencanaan. Secara garis besar, pada *SDLC* memuat 3 (tiga) hal penting yang akan dilakukan pada proses pengembangan perangkat lunak, yakni penganalisisan kebutuhan perangkat lunak, perancangan dan implementasi perangkat lunak, serta pengujian hasil pengembangan perangkat lunak. Analisis kebutuhan penting dilakukan sebagai awal dari perancangan perangkat lunak yang akan digunakan. Dari analisis kebutuhan tersebut, dapat diketahui garis besar proses kerja perangkat lunak yang akan dikembangkan. Selain itu, dengan proses analisis kebutuhan dapat meningkatkan kualitas perangkat lunak yang akan dirancang sehingga perangkat lunak yang dikembangkan akan sesuai dengan kebutuhan pengguna. Setelah analisis kebutuhan telah didapatkan, masuk ke dalam tahap implementasi. Dalam tahap implementasi tersebut, terdapat proses perancangan perangkat lunak untuk mempermudah proses perancangan perangkat lunak serta implementasi yakni proses pembangunan perangkat lunak dari hasil perancangan yang akan dibuat. Terakhir, terdapat pengujian untuk melihat sejauh mana manfaat atau hasil dari pengembangan perangkat lunak yang telah dilakukan, sehingga dapat diketahui sejauh mana perangkat lunak yang dikembangkan dapat diterima oleh pengguna.

Berdasarkan penjabaran dari beberapa alasan tersebut, penulis melakukan penelitian skripsi dengan judul **“Pengembangan Sistem Informasi Asesmen Lembaga Sertifikasi Profesi Berbasis Web (Studi Kasus: SMK Negeri 4 Malang)”**. Diharapkan dari hasil penelitian ini akan memberikan dampak positif dari proses asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang dalam hal efisiensi waktu proses asesmen serta efisiensi penggunaan kertas dalam proses asesmen maupun penyimpanan dokumen dan data hasil asesmen.

1.2 Rumusan Masalah

1. Bagaimana hasil analisis kebutuhan sistem informasi asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang?
2. Bagaimana hasil perancangan dan implementasi sistem informasi asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang?
3. Bagaimana hasil pengujian sistem informasi asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang?

1.3 Tujuan

1. Untuk mengetahui hasil analisis kebutuhan sistem informasi asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang.
2. Untuk mengetahui hasil perancangan dan implementasi sistem informasi asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang.
3. Untuk mengetahui hasil dari pengujian sistem informasi asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang.

1.4 Manfaat

Manfaat utama dari sistem informasi ini adalah untuk menghemat penggunaan kertas dengan cara digitalisasi data asesmen. Dengan sistem informasi tersebut jumlah kertas yang dikeluarkan akan jauh lebih sedikit. Manfaat lain dari sistem informasi asesmen LSP ini adalah asesor akan lebih mudah dan cepat (menurut hasil wawancara) dalam melakukan proses asesmen pada siswa tanpa perlu direpotkan dengan banyaknya jumlah dokumen yang harus diisi serta proses melihat panduan pengisian asesmen. Selain itu, asesor akan dapat melihat atau mencari data hasil asesmen dengan mudah tanpa perlu mencari data tersebut pada dokumen fisik masing-masing siswa yang tersimpan dengan jumlah yang tidak sedikit.

1.5 Batasan Masalah

1. Objek penelitian dan pengambilan data terbatas pada jurusan Rekayasa Perangkat Lunak SMK Negeri 4 Malang.
2. Metode pengembangan perangkat lunak yang digunakan ialah menggunakan model *waterfall* tanpa proses *deployment* atau pengambilan *feedback* dan pemeliharaan.
3. Perangkat lunak asesmen yang dikembangkan terbatas pada asesmen mandiri tanpa asesmen-asesmen lain.

1.6 Sistematika Pembahasan

Sistematika pembahasan yang digunakan dalam penelitian ini adalah sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini menguraikan latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

2. BAB II LANDASAN KEPUSTAKAAN

Bab landasan kepastakaan ini meliputi kajian penelitian yang berisi tentang hasil-hasil penelitian terdahulu yang berkaitan dengan penelitian yang dilakukan serta landasan teori yang berisi tentang pembahasan pengertian Sistem Informasi, Asesmen, dan pembahasan *SDLC (System Development Life*

Cycle) beserta pembahasan tentang metode yang digunakan dalam masing-masing proses dalam *System Development Life Cycle (SDLC)*.

3. BAB III METODOLOGI PENELITIAN IMPLEMENTATIF

Dalam bab ini penulis mengemukakan tentang metode penelitian yang dilakukan oleh penulis dalam membuat sistem informasi. Agar sistematis, bab metode penelitian meliputi: perumusan masalah, analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis, dan kesimpulan serta saran.

4. BAB IV HASIL DAN PEMBAHASAN

Dalam bab hasil dan pembahasan penulis mengemukakan tentang pelaksanaan metodologi yang telah dituliskan sebelumnya. Dalam bab ini pula terdapat beberapa proses diantaranya analisis kebutuhan, perancangan sistem, implementasi, serta pengujian dan analisis dari perangkat lunak.

5. BAB V PENUTUP

Pada bab penutup menjelaskan kesimpulan yang didapatkan dari proses penelitian yang dilakukan serta apa saja yang telah didapatkan dari proses penelitian ini. Selain itu, terdapat saran yakni masukan dari peneliti untuk peneliti lain yang menggunakan skripsi ini sebagai acuan penelitian berikutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini menjelaskan tentang teori yang digunakan dalam proses penyelesaian masalah dalam skripsi ini yakni pengembangan sistem informasi asesmen lembaga sertifikasi profesi di SMKN 4 Malang.

2.1 Penelitian Terkait

Dalam proses penelitian pengembangan Sistem Informasi Asesmen Lembaga Sertifikasi Profesi ini, penulis mengumpulkan penelitian terkait untuk menunjang proses penelitian dan pembuatan laporan. Penelitian terkait pertama ialah penelitian skripsi yang dilakukan oleh Nurmala (2017) yang mengangkat judul Aplikasi E-School pada SMP Negeri 1 Cikarang Barat. Dalam penelitian tersebut, penulis mengambil informasi bagaimana metode yang digunakan peneliti untuk mendapatkan data dari sekolah. Metode yang digunakan ialah metode wawancara dengan *stakeholder* yang berkaitan dengan sistem informasi yang akan dibuat. Setelah melakukan wawancara, peneliti melakukan analisis kebutuhan sistem dari hasil wawancara yang dilakukan. Dari hasil analisis kebutuhan tersebut, peneliti dapat membuat perancangan perangkat lunak serta implementasi perangkat lunak menjadi aplikasi E-School. Untuk mengambil kesimpulan dari hasil penelitian, peneliti melakukan wawancara kembali disertai dengan kuesioner kepada *stakeholder* atau guru pengguna aplikasi tersebut.

Untuk penelitian terkait kedua, penulis mengambil informasi dari penelitian yang dilakukan oleh Pratama (2017) yang mengangkat judul Sistem Informasi Manajemen Pariwisata Terpadu Sebagai Media Promosi Kabupaten Banyuwangi. Dalam penelitian tersebut, pemecahan masalah promosi Kabupaten Banyuwangi menggunakan suatu sistem informasi manajemen sebagai media promosi. Dari kesimpulan penelitian tersebut, bahwa pengembangan sistem informasi manajemen pariwisata sebagai media promosi menggunakan *framework CodeIgniter* dapat dilaksanakan dengan baik dapat dibuktikan dengan pengujian perangkat lunak yang dijalankan menunjukkan bahwa seluruh fungsional dapat berjalan dengan baik. Dari penelitian terkait tersebut, penulis mengambil informasi bagaimana pengembangan sistem informasi menggunakan *framework CodeIgniter* dapat diimplementasikan atau dikembangkan untuk mengatasi suatu permasalahan yang ada.

Untuk penelitian terakhir, penulis mengambil informasi dari penelitian yang dilakukan oleh Evantio (2017) yang mengangkat judul tentang Pengembangan Sistem Informasi *E-Complaint Management* (Studi Kasus: Batching Plant Produksi Beton P.T. Holcim Indonesia Regional Jawa Timur). Penulis mengambil informasi dari penelitian terkait tersebut mengenai bagaimana peneliti sebelumnya melakukan pengujian terhadap sistem informasi yang telah dikembangkan menggunakan *User Acceptance Test (UAT)*. Kesimpulan dari penelitian tersebut bahwa terdapat 4 (empat) permasalahan utama dalam proses laporan keluhan PT.

Holcim Indonesia diantaranya laporan keluhan tidak dapat diakses bersamaan karena data tersimpan dalam *file excel*. Selain itu, pelanggan tidak mengetahui tindak lanjut dari keluhan yang mereka ajukan. Dari hasil pemecahan masalah menggunakan perangkat lunak yang digunakan dan dilihat dari hasil pengujian *User Acceptance Test (UAT)* menyatakan bahwa sistem bisa diterima oleh pengguna dari *leader* dengan mendapatkan nilai 5 dari total 5 poin pertanyaan kuisisioner UAT. Tetapi tidak diterima secara keseluruhan oleh sales dengan mendapatkan nilai 2 dari total 5 poin pertanyaan kuisisioner UAT.

Dari beberapa penelitian terkait tersebut, dapat ditarik kesimpulan bahwa beberapa penelitian sebelumnya menyelesaikan permasalahannya dengan menggunakan teknologi informasi khususnya dengan membangun suatu sistem informasi.

2.2 Sistem Informasi

Menurut Davis (1991), "Sistem informasi adalah suatu sistem yang menerima *input* atau masukan data dan instruksi, mengolah data sesuai dengan instruksi dan mengeluarkan hasilnya". Sedangkan menurut Jogiyanto (2005) dalam buku "Analisis dan Desain Sistem Informasi" menyebutkan bahwa suatu sistem yang berjalan pada sebuah organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung semua operasi, bersifat manajerial dan menyediakan pihak luar tertentu dengan kumpulan laporan yang dibutuhkan disebut juga dengan sistem informasi. Selain itu, pada latar belakang juga disebutkan bahwa sistem informasi menurut Ariawan (2009), dalam Buku Ajar Sistem Informasi Manajemen, sistem informasi adalah sebagai kumpulan/group dari subsistem/bagian/ komponen apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berarti dan berguna. Dari beberapa pengertian tersebut dapat disimpulkan bahwa sistem informasi merupakan suatu sistem atau kumpulan perintah yang menerima *input* berupa data maupun perintah pengguna dan mengolah masukan dan data tersebut sehingga didapatkan keluaran informasi yang berguna bagi pengguna. Oleh karena itu, dengan menggunakan sistem informasi dapat pula untuk menyelesaikan masalah yakni proses asesmen Lembaga Sertifikasi Profesi (LSP) pada SMK Negeri 4 Malang.

2.3 SMK Negeri 4 Malang

SMK Negeri 4 Malang adalah sekolah yang beralamat di Jalan Tanimbar Nomor 22 Kota Malang serta dikepalai oleh Dr. Wadib Su'udi M.M. SMK Negeri 4 Malang telah memiliki lisensi LSP P1 pada jurusan Rekayasa Perangkat Lunak dan Multimedia berdasarkan Surat Keputusan Kementerian Pendidikan Dan Kebudayaan Republik Indonesia nomor 467/D5.6/KP tahun 2018 tentang Perubahan Atas Keputusan Nomor 003/D5.6/KP/2018 tentang Perubahan Atas Keputusan Nomor 1066/D.5.6/KP/2017 tentang Penetapan SMK Sebagai Jejaring

Kerja Lembaga Sertifikasi Profesi Pihak Pertama (LSP P1) Di Lingkungan Sekolah Menengah Kejurusan (Tahap 1). Berdasarkan laman *web* Badan Nasional Sertifikasi Profesi, terdapat 5 (lima) guru yang telah menerima lisensi sebagai asesor. 5 (lima) guru tersebut bertugas untuk melakukan asesmen pada siswa pemohon sertifikasi profesi.

2.4 Asesmen

Asesmen adalah proses pengungkapan dan pemahaman permasalahan, kebutuhan, dan potensi klien, serta sumber yang dapat digunakan untuk mengatasi permasalahan klien (Glosarium Kemensos, 2018). Dari pengertian tersebut, asesmen digunakan untuk menentukan kualifikasi pemohon sertifikasi dan untuk mengetahui potensi dari pemohon sertifikasi yang didapatkan dari proses asesmen yang dilakukan oleh Lembaga Sertifikasi Profesi (LSP). Pada proses asesmen yang dilakukan di SMK Negeri 4 Malang, dilakukan oleh guru yang telah mendapat lisensi sebagai asesor dari Badan Nasional Sertifikasi Profesi (BNSP) dengan cara mengisi lembar asesmen yang berisi syarat kompetensi yang harus dikuasai oleh pemohon sertifikasi profesi yakni siswa. Asesor tersebut bekerja dibawah naungan LSP SMK Negeri 4 Malang.

2.5 Lembaga Sertifikasi Profesi (LSP)

Lembaga Sertifikasi Profesi (LSP) adalah lembaga pelaksana kegiatan kompetensi kerja yang mendapatkan lisensi dari Badan Nasional Sertifikasi Profesi (BNSP) (Kemdikbud, 2018). LSP bertugas menyelenggarakan proses asesmen untuk mendapatkan sertifikasi profesi bagi pemohon. Hasil asesmen akan dijadikan pertimbangan dalam penentuan apakah pemohon sertifikasi profesi telah layak untuk menerima sertifikasi profesi yang dikeluarkan oleh Badan Nasional Sertifikasi Profesi (BNSP).

2.6 Badan Nasional Sertifikasi Profesi (BNSP)

Badan Nasional Sertifikasi Profesi (BNSP) merupakan badan independen yang bertanggung jawab kepada Presiden yang memiliki kewenangan sebagai otoritas sertifikasi personil dan bertugas melaksanakan sertifikasi kompetensi profesi bagi tenaga kerja. Pembentukan BNSP merupakan bagian integral dari pengembangan paradigma baru dalam sistem penyiapan tenaga kerja yang berkualitas. Berbeda dengan paradigma lama yang berjalan selama ini, sistem penyiapan tenaga kerja dalam format paradigma baru terdapat dua prinsip yang menjadi dasarnya, yaitu: pertama, penyiapan tenaga kerja didasarkan atas kebutuhan pengguna (*demand driven*); dan kedua, proses diklat sebagai wahana penyiapan tenaga kerja dilakukan dengan menggunakan pendekatan pelatihan berbasis kompetensi (*Competency Based Training / CBT*). Pengembangan sistem penyiapan tenaga kerja dengan paradigma baru ini dimulai pada awal tahun 2000 yang ditandai dengan ditandatanganinya Surat Kesepakatan Bersama (SKB) antara Menteri

Tenaga Kerja, Menteri Pendidikan Nasional, Ketua Umum Kadin Indonesia (BNSP, 2018).

2.7 Rekayasa Perangkat Lunak

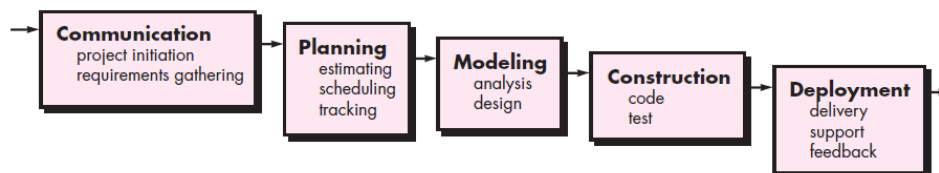
Rekayasa Perangkat Lunak (*Software Engineering*) menurut Sommerville (2011) dalam bukunya yang berjudul *Software Engineering 9th* menyatakan bahwa Rekayasa Perangkat Lunak (*Software Engineering*) adalah disiplin ilmu yang berfokus pada segala aspek tentang produksi perangkat lunak mulai dari tahap awal perencanaan sistem hingga perawatan sistem setelah perangkat lunak digunakan. Dari pengertian tersebut dapat diketahui bahwa Rekayasa Perangkat Lunak adalah suatu keilmuan tentang perancangan, pembuatan, hingga perawatan perangkat lunak yang digunakan untuk tujuan tertentu. Dalam proses pengembangan perangkat lunak terdapat teori terkait tahapan-tahapan yang dilakukan untuk membangun suatu perangkat lunak yang disebut dengan *SDLC* (*System Development Life Cycle*).

2.8 System Development Life Cycle (SDLC)

System Development Life Cycle (SDLC) dalam Bahasa Indonesia disebut dengan Siklus Hidup Pengembangan Sistem menurut Kendall & Kendall (2014) menyatakan bahwa *SDLC* adalah sebuah model pendekatan yang memiliki tahapan-tahapan dalam melakukan sebuah analisis dan membangun rancangan sistem dengan menggunakan siklus yang terperinci terhadap kegiatan pengguna. Terdapat berbagai macam *SDLC* yang ada, namun pada dasarnya terdapat lima fase yakni perencanaan, analisis, desain, implementasi dan perawatan. Salah satu *SDLC* yang menerapkan beberapa fase mulai dari perencanaan hingga perawatan adalah *waterfall*.

2.9 Waterfall Model

Menurut Pressman (2015) dari bukunya yang berjudul “Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku I” menyatakan bahwa model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Contoh alur pengembangan perangkat lunak menggunakan model *waterfall* dapat dilihat pada gambar 2.1.



Gambar 2.1 *Waterfall Model* (Pressman, 2015)

Dalam proses tersebut terdapat 5 (lima) tahapan dalam pengembangan aplikasi atau perangkat lunak, yakni *communication*, *planning*, *modeling*,

repository.ub.ac.id

construction, dan *deployment*. Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan di dalam Model *Waterfall* menurut Pressman (2015):

2.9.1 Communication

Pada proses *communication* merupakan tahapan awal dalam pengembangan perangkat lunak yakni dengan melakukan diskusi bersama *customer* atau pengguna jasa. Dari hasil diskusi tersebut, didapatkan sejumlah informasi diantaranya kebutuhan sistem, permasalahan yang dihadapi, bagaimana sistem bekerja dan lain sebagainya.

Dalam diskusi tersebut digunakan teknik wawancara secara langsung kepada narasumber atau *stakeholder* atau pemegang kepentingan dari sistem yang akan dibuat. Untuk butir-butir pertanyaan yang diajukan kepada narasumber dapat dilihat pada lampiran B. Dalam proses *communication* yang telah dilakukan yakni diskusi dan wawancara akan menghasilkan kesepakatan tentang fitur-fitur dan fungsi aplikasi yang akan dibuat. Setelah tahap *communication* selesai dilakukan, akan dilanjutkan menuju tahap *planning*.

2.9.2 Planning

Tahap *planning* merupakan tahap perencanaan dan dalam tahap ini mendefinisikan tentang estimasi pembuatan aplikasi, penjadwalan pembuatan aplikasi, dan *tracking* atau pemantauan proses pembuatan aplikasi. Selain itu, *planning* merupakan tahap untuk menghasilkan kebutuhan-kebutuhan aplikasi yang akan dirancang. Tahapan selanjutnya yang akan dilakukan ialah *modeling*.

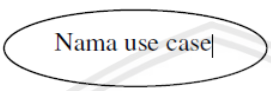
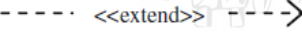
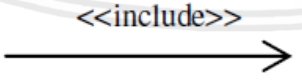
2.9.3 Modeling

Modeling merupakan proses perancangan dasar aplikasi. Pengembang akan membuat rancangan aplikasi yang berfokus pada arsitektur *software* dan proses kerja program. Tujuan dari proses *modeling* adalah memahami alur kerja aplikasi sebelum aplikasi tersebut diimplementasikan pada kode program. Pada proses *modeling* dapat digambarkan pada beberapa bentuk diantaranya ialah *Unified Modeling language (UML)* dan *Data Flow Diagram (DFD)*. Untuk penggambaran perancangan perangkat lunak menggunakan *object-oriented programming* menggunakan bentuk *UML*. Menurut Rosa & Shalahuddin (2015) *UML* adalah salah satu standar bahasa visual yang banyak digunakan di dunia industri untuk mengidentifikasi *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. *UML* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. Dalam *UML* terdapat 9 (sembilan) standar diagram yang dapat digunakan untuk menggambarkan sebuah sistem, antara lain: *class diagram*, *object diagram*, *use case diagram*, *sequence diagram*, *collaboration diagram*, *state chart diagram*, *activity diagram*, *component diagram*, dan *deployment diagram*.

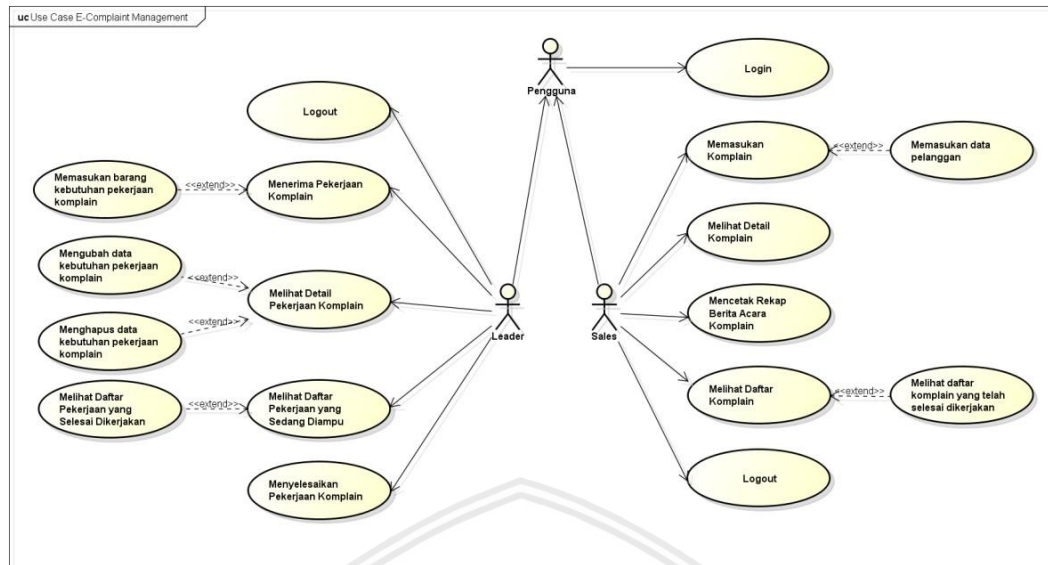
2.9.3.1 Use Case Diagram

Use case diagram menurut Rosa & Shalahuddin (2015) dalam bukunya yang berjudul “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek” menjelaskan bahwa *use case diagram* adalah pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Dalam *use case diagram* menggambarkan fungsi-fungsi atau hal yang dapat dilakukan oleh satu atau lebih aktor dalam sebuah sistem operasi. Contoh penggambaran *use case diagram* dapat dilihat pada tabel 2.1.

Tabel 2.1 Simbol Use Case Diagram

No.	Nama	Simbol	Fungsi
1	<i>Use Case</i>		Digunakan untuk menggambarkan fungsionalitas satu atau lebih aktor. Penulisan menggunakan kata kerja.
2	<i>Actor</i>		Digunakan untuk menggambarkan aktor atau jenis pengguna sistem informasi. Untuk penulisan menggunakan nama benda.
3	<i>Association</i>		Digunakan untuk menggambarkan komunikasi aktor dengan <i>use case</i> . Sehingga dapat digunakan untuk menunjukkan bahwa aktor dapat melakukan suatu fungsionalitas tertentu.
4	<i>Extend</i>		Merupakan hubungan antara <i>use case</i> untuk menunjukkan suatu alur dari <i>use case</i> satu menuju <i>use case</i> lainnya.
5	<i>Include</i>		Merupakan hubungan antara <i>use case</i> untuk menunjukkan bahwa seorang aktor harus menjalankan suatu <i>use case</i> untuk dapat menjalankan <i>use case</i> lainnya.

Untuk contoh dari *use case diagram* dapat dilihat pada gambar 2.2. Dari gambar 2.2 tersebut terdapat *use case diagram e-complaint management* (Evantio, 2017). Dalam gambar 2.2 menjelaskan bahwa terdapat 2 aktor yakni *leader* dan *sales*. 2 aktor tersebut merupakan turunan dari aktor pengguna, dengan kata lain *leader* dan *sales* merupakan bagian dari pengguna. Kedua aktor tersebut dapat melakukan kegiatan *login*, namun untuk fungsional yang dapat dijalankan selanjutnya kedua aktor tersebut mempunyai fungsional yang berbeda.

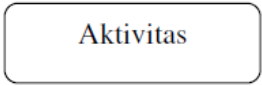


Gambar 2.2 Contoh Use Case Diagram E-Complaint Management (Evantion, 2017)

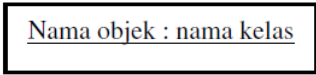
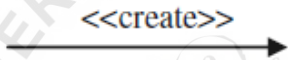
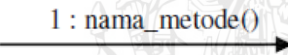
2.9.3.2 Activity Diagram

Menurut Rosa & Shalahuddin (2015) *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. *Activity diagram* menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor. Simbol yang digunakan dalam penggambaran *activity diagram* dapat dilihat pada tabel 2.2.

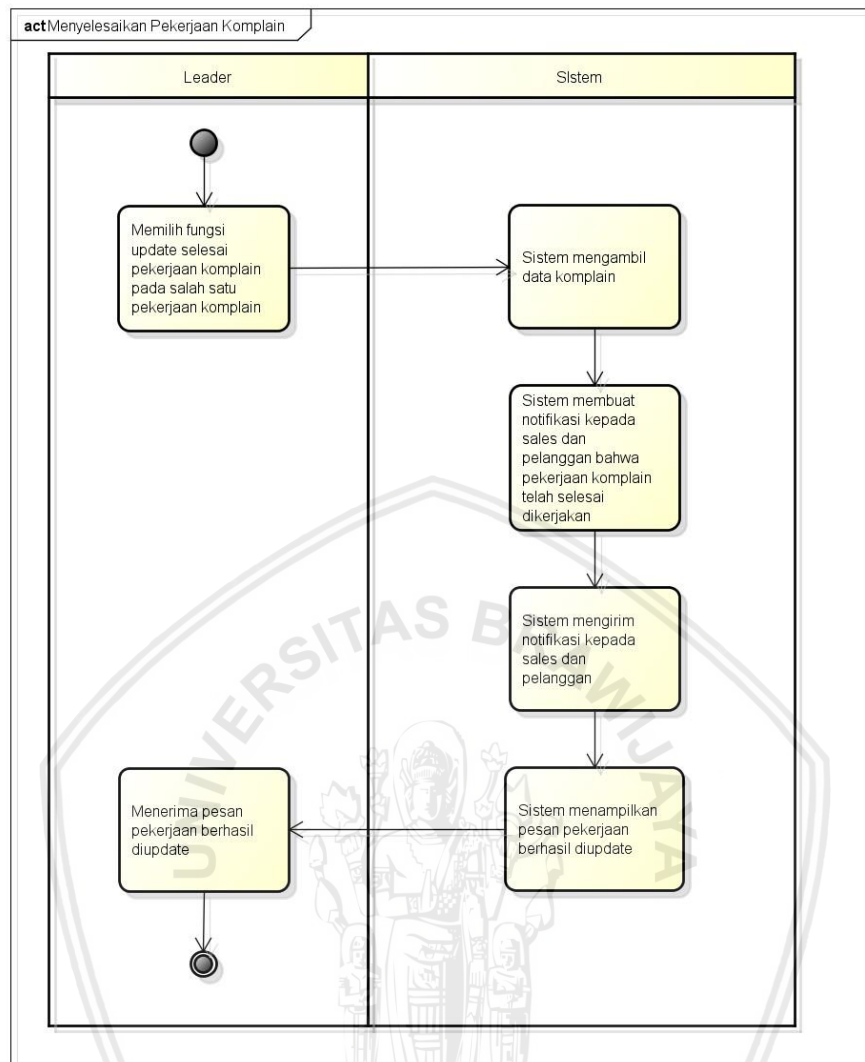
Tabel 2.2 Simbol Activity Diagram

No.	Nama	Simbol	Fungsi
1	Status awal		Digunakan untuk menggambarkan status awal dari aktivitas.
2	Aktivitas		Digunakan untuk menggambarkan aktivitas yang dilakukan sistem.
3	Percabangan		Digunakan untuk menggambarkan percabangan jika terdapat pilihan aktivitas lebih dari satu.
4	Penggabungan		Digunakan untuk menggambarkan penggabungan lebih dari satu aktivitas menjadi satu.

Tabel 2.2 Simbol *Activity Diagram* (Lanjutan)

No.	Nama	Simbol	Fungsi
5	Status akhir		Digunakan untuk menggambarkan status akhir dari aktivitas.
6	<i>Object</i>		Digunakan untuk menyatakan objek yang berinteraksi dengan pesan.
7	Waktu aktif		Digunakan untuk menyatakan objek dalam keadaan aktif dan berinteraksi.
8	Pesan tipe <i>create</i>		Digunakan untuk menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
9	Pesan tipe <i>call</i>		Digunakan untuk menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.

Untuk contoh dari *activity diagram* dapat dilihat pada gambar 2.3. Dalam contoh tersebut terdapat aktivitas menyelesaikan pekerjaan complain dalam sistem *e-complaint management* (Evantio, 2017). Aktivitas tersebut dilakukan oleh *leader* yang dimulai dengan memilih fungsi *update* selesai pekerjaan complain pada salah satu pekerjaan. Setelah itu, sistem akan merespon dengan mengambil data complain, lalu sistem membuat notifikasi kepada *sales* dan pelanggan bahwa pekerjaan complain telah selesai dikerjakan. Langkah terakhir, sistem menampilkan pesan bahwa pekerjaan berhasil di update dan pesan tersebut akan diterima *leader* sebagai pelaksana aktivitas.



Gambar 2.3 Contoh Activity Diagram Menyelesaikan Pekerjaan Komplain (Evantio, 2017)

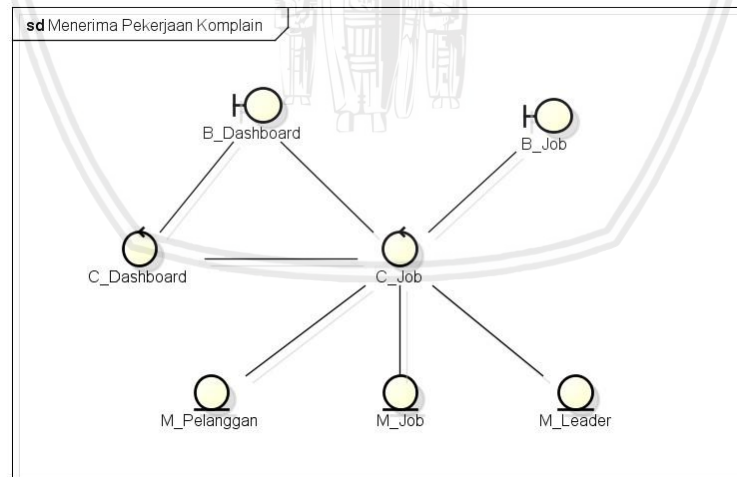
2.9.3.3 Collaboration Diagram

Collaboration diagram adalah penggambaran relasi antar kelas yang terjadi pada suatu fungsional. Kegunaan dari penggambaran *collaboration diagram* ialah untuk menentukan jumlah kelas yang akan dibuat di dalam sistem. Selain itu, kegunaan *collaboration diagram* ialah memperjelas komunikasi yang terjadi antar kelas pada fungsional yang dijalankan pengguna. Berikut pada tabel 2.3 adalah simbol-simbol dari *collaboration diagram*.

Tabel 2.3 Simbol *Collaboration Diagram*

No.	Nama	Simbol	Fungsi
1	<i>Model</i>		Digunakan untuk menggambarkan kelas yang berhubungan dengan data dan <i>database</i> .
2	<i>View</i>		Digunakan untuk menggambarkan kelas yang menampilkan halaman pada pengguna atau yang berinteraksi dengan pengguna
3	<i>Controller</i>		Digunakan untuk menggambarkan kelas yang mengelola data dari semua proses dalam sistem.
4	<i>Relation</i>		Digunakan untuk menggambarkan hubungan antara beberapa simbol seperti <i>model</i> , <i>view</i> , dan <i>controller</i> .

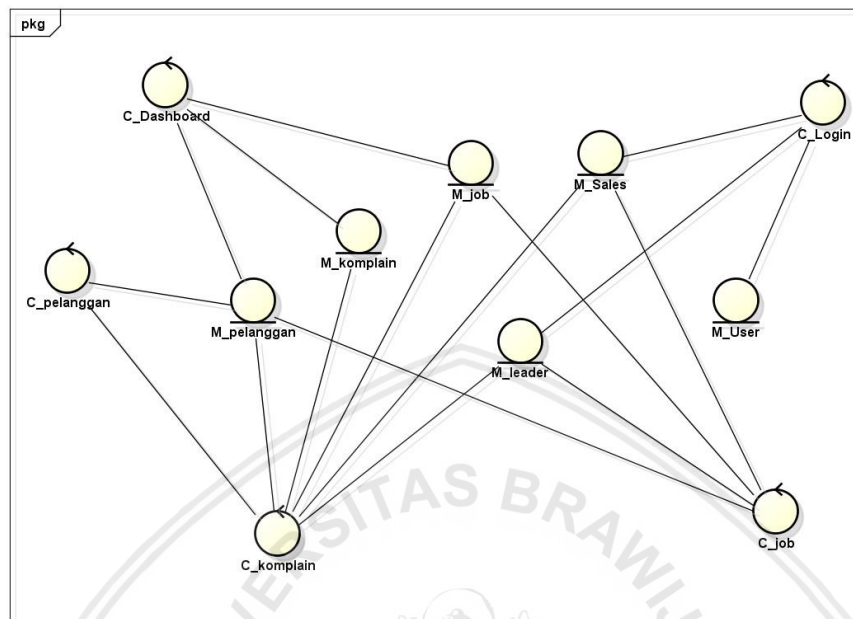
Untuk contoh dari *collaboration diagram* dapat dilihat pada gambar 2.4. Dari contoh tersebut yakni menerima pekerjaan komplain *e-complaint management* (Evantio, 2017), terdapat 2 (dua) *view*, 2 (dua) *controller*, dan 3 (tiga) *model* untuk menjalankan proses menerima pekerjaan *complaint*. Dapat dilihat bahwa semua *model* dikendalikan oleh *controller* C_Job dan *view* dari B_Dashboard dan B_Job. Sedangkan *Controller* C_Dashboard berfungsi sebagai pengendali dari *view* B_Dashboard dan berhubungan dengan C_Job.



Gambar 2.4 Contoh *Collaboration Diagram* Menerima Pekerjaan Komplain (Evantio, 2017)

Setelah *collaboration diagram* didapatkan dari seluruh fungsional yang ada, maka dilanjutkan dengan melakukan unifikasi kelas analisis. Unifikasi kelas analisis yakni pengumpulan atau penggabungan seluruh *collaboration diagram* menjadi satu kesatuan yang bertujuan untuk melihat bagaimana tingkat relasi antar satu

bojek dengan objek yang lain. Selain itu, penggabungan tersebut untuk mendapatkan kelas minimum yang akan diimplementasikan ke dalam sistem. Contoh unifikasi kelas analisis dapat dilihat pada gambar 2.5.



Gambar 2.5 Unifikasi Kelas Analisis *e-complaint management* (Evantio, 2017)

2.9.3.4 Sequence Diagram

Menurut Rosa & Shalahuddin (2015) *sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Selain itu, dalam *sequence diagram* juga menggambarkan proses interaksi pengguna dengan sistem dan proses yang terjadi setelah pengguna berinteraksi dengan sistem. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case*. Dalam penggambaran *sequence diagram* menggunakan simbol-simbol yang dapat dilihat pada tabel 2.4.

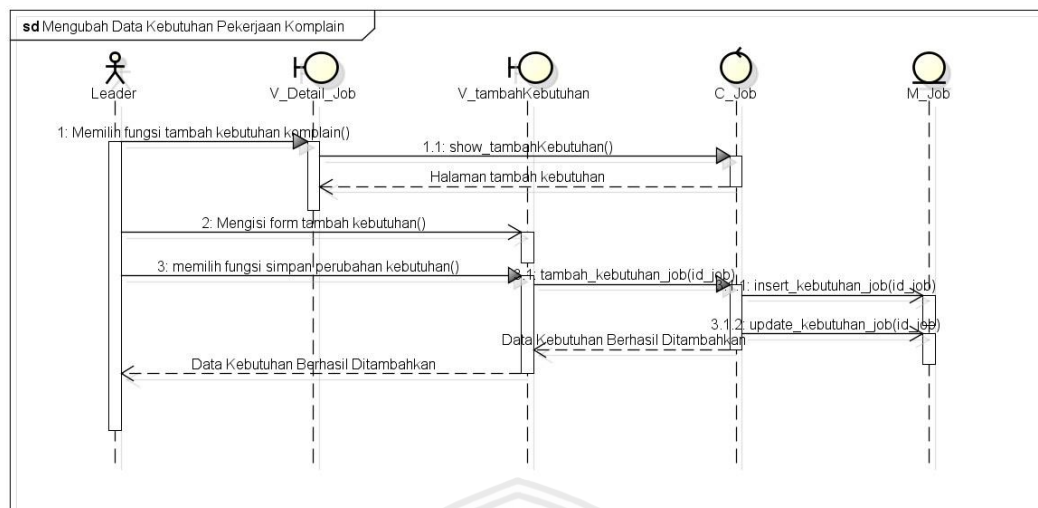
Tabel 2.4 Simbol *Sequence Diagram*

No.	Nama	Simbol	Fungsi
1	Actor		Digunakan untuk menggambarkan aktor atau orang yang berinteraksi dengan sistem.
2	Boundary		Digunakan untuk menggambarkan tampilan atau view.

Tabel 2.4 Simbol *Sequence Diagram* (Lanjutan)

No.	Nama	Simbol	Fungsi
3	<i>Control</i>		Digunakan untuk menggambarkan proses suatu tugas atau fungsi sistem.
4	<i>Entity</i>		Digunakan untuk menggambarkan informasi yang harus disimpan oleh sistem.
5	<i>Lifeline</i>		Digunakan untuk menggambarkan fungsi atau proses yang dilakukan pada satu waktu tertentu.
6	<i>Message</i>		Digunakan untuk menggambarkan interaksi atau pesan yang dikirimkan dari objek satu menuju objek lainnya.
7	<i>Reply Message</i>		Digunakan untuk menggambarkan pesan balasan dari objek yang berkomunikasi dengan objek lainnya.

Untuk contoh dari *sequence diagram* dapat dilihat pada gambar 2.6. Dari gambar 2.6 menunjukkan proses pengubahan data kebutuhan pekerjaan komplain yang dilakukan oleh *leader*. Dalam *sequence diagram* tersebut terdapat 2 (dua) *view*, 1 (satu) *controller*, dan 1 (satu) *model*. Proses tersebut dimulai dari *leader* memilih fungsi tambah kebutuhan komplain pada *view* V_Detail_Job. Setelah itu, sistem akan memanggil fungsi `show_tambahKebutuhan` untuk memanggil halaman tambah kebutuhan. Kemudian, *leader* memilih fungsi simpan perubahan kebutuhan pada *view* V_tambahKebutuhan. Langkah selanjutnya, sistem akan memanggil fungsi `tambah_kebutuhan_job(id_job)` pada *controller* C_Job. Fungsi tersebut akan berhubungan dengan *model* M_Job pada `insert_kebutuhan_job(id_job)` dan `update_kebutuhan_job(id_job)` untuk menyimpan data pada *database*. Setelah itu, sistem akan menampilkan pesan bahwa data kebutuhan berhasil ditambahkan kepada *leader*.

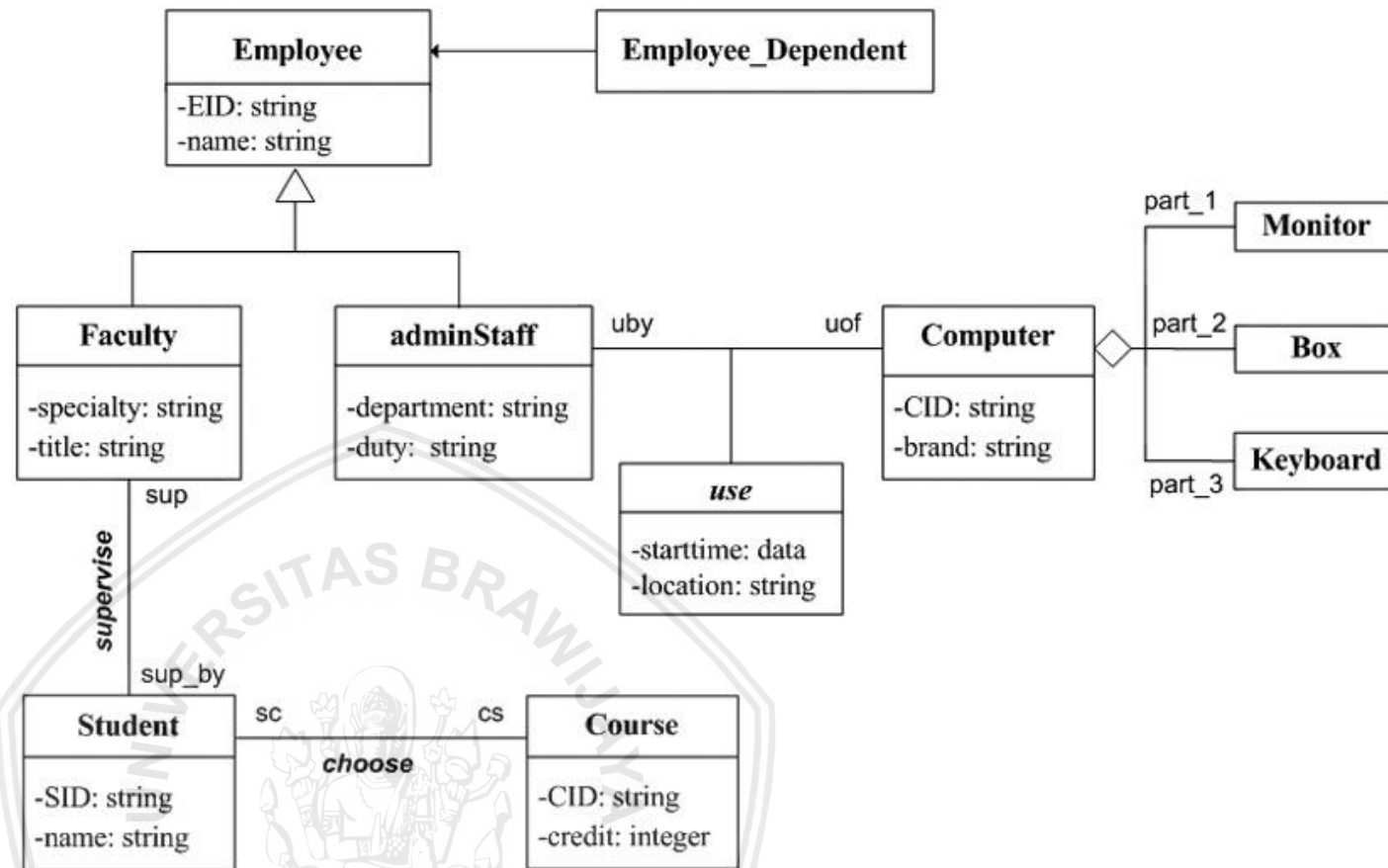


Gambar 2.6 Contoh Sequence Diagram Mengubah Data Kebutuhan Pekerjaan Komplain (Evantio, 2017)

2.9.3.5 Class Diagram

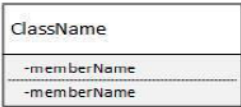
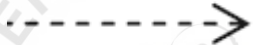
Menurut Rosa & Shalahuddin (2015) *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Dalam sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* adalah kondisi suatu objek dinyatakan dalam *attribute/properties*. Sedangkan perilaku (*behavior*) digunakan untuk mendefinisikan bagaimana sebuah objek dapat melakukan aksi dan juga memberikan reaksi. Contoh *class diagram* dapat dilihat pada gambar 2.7.

Dari gambar 2.7, dapat dilihat bahwa pada contoh *class diagram* universitas tersebut terdapat 7 (tujuh) kelas utama diantaranya *employee*, *faculty*, *adminStaff*, *student*, dan lain sebagainya. Pada masing-masing kelas tersebut terdapat atribut yakni karakteristik atau nilai yang dimiliki oleh kelas tersebut. Terdapat beberapa hubungan yang terjadi pada beberapa kelas tersebut diantaranya asosiasi, generalisasi, dan agregasi. Contoh Untuk penggambaran *class diagram* beserta hubungannya menggunakan beberapa simbol yang dapat dilihat pada tabel 2.5.



Gambar 2.7 Contoh Class Diagram Universitas (Tong, 2017)

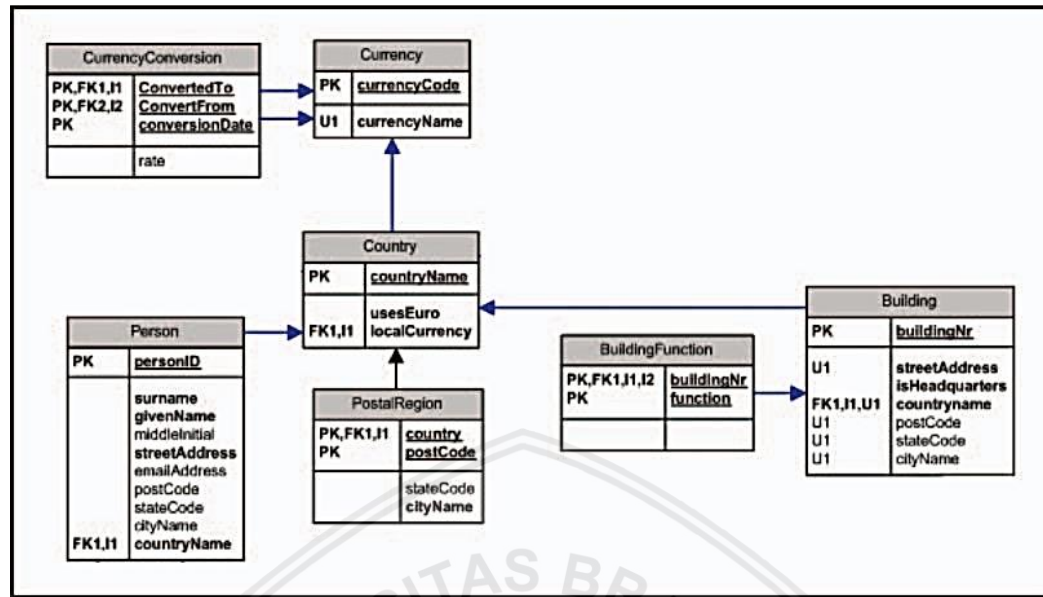
Tabel 2.5 Simbol *Class Diagram*

No.	Nama	Simbol	Fungsi
1	<i>Class</i>		Digunakan untuk menggambarkan satu kelas beserta <i>behavior</i> -nya
2	<i>Interface</i>		Digunakan untuk menggambarkan konsep <i>interface</i> dalam pemrograman berorientasi objek
3	<i>Association</i>		Merupakan hubungan antar kelas dengan makna umum
4	<i>Directed Association</i>		Merupakan hubungan antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain
5	<i>Generalization</i>		Merupakan hubungan antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
6	<i>Dependency</i>		Merupakan hubungan antar kelas dengan makna kebergantungan antar kelas
7	<i>Aggregation</i>		Merupakan hubungan antar kelas dengan makna semua bagian

2.9.3.6 *Physical Data Model*

Menurut West (2011) dalam bukunya *Developing High Quality Data Models* dikutip dari *ScienceDirect.com* berpendapat bahwa *physical data model* merupakan representasi dari struktur database baik tabel beserta kolomnya. Entitas biasanya merepresentasikan tabel dan garis relasi merepresentasikan *foreign key* antar tabel.

Dalam proses perancangan *physical data model*, diperlukan pengetahuan yang lebih dalam *database management system* (DBMS) karena diperlukan untuk: (1) Merepresentasikan *logical data model* dalam skema *database*; (2) Memakai entitas dan atribut sesuai dengan kebutuhan sistem; dan (3) Mengkonfigurasi *database* sesuai dengan performa yang disyaratkan. (Sherman, 2015). Contoh *physical data model* dapat dilihat pada gambar 2.8.



Gambar 2.8 Contoh *Physical Data Model* (Fowler & Stanwick, 2004)

Dari gambar 2.7 tersebut dapat dilihat bahwa pada *physical data model* merupakan gambaran asli dari *database* yang akan dibuat. Contohnya terdapat tanda untuk *foreign key*, *primary key*, serta atribut yang sesuai dengan kolom yang akan dibuat pada *database*. Selain itu, terdapat garis yang menunjukkan relasi antar tabel yang akan dibuat.

2.9.4 Construction

Tahap *construction* merupakan proses penerapan rancangan aplikasi yang telah dibuat dalam kondisi yang sesungguhnya. Dalam tahap *construction* terdapat dua kegiatan yang dilakukan yakni implementasi (*code*) dan pengujian (*test*).

2.9.4.1 Implementasi

Proses implementasi merupakan salah satu proses dalam tahapan *construction* yakni penerapan rancangan perangkat lunak dalam kode program. Dalam penerapannya, terdapat metode yang dinamakan *framework* yakni kerangka kerja dalam proses implementasi kode program. Berdasarkan pengertian umum secara harfiah *framework* adalah kerangka kerja. Secara umum, *framework* menggunakan struktur MVC (*Model, View, Controller*). Penggunaan struktur MVC tersebut dimaksudkan agar pembuat atau pengembang program terbantu dalam pembuatan perangkat lunak khususnya web karena dengan adanya *framework* dengan struktur MVC pengembang dapat mengorganisasikan jalannya perangkat lunak pada satu ketentuan tertentu sehingga pengembang selanjutnya akan mudah dalam membaca jalannya program dengan struktur MVC ini. Terdapat beberapa jenis *framework* yang ada diantaranya *CodeIgniter*.

Codeigniter merupakan salah satu *framework* atau kerangka kerja (yang telah dijelaskan di atas) PHP dengan struktur MVC (*Model, View, Controller*) yang digunakan untuk pembangunan *website* dinamis. Dikutip dari *website* codeigniter bahwa codeigniter merupakan *framework* pengembangan aplikasi bagi pengembang yang mengembangkan *website* menggunakan *php* (CodeIgniter, 2018). Tujuan utama dari codeigniter itu sendiri menurut *website* codeigniter ialah mempercepat proses pengembangan *website* serta memperkecil penggunaan *code* untuk menjalankan suatu proses.

A. Model

Model merupakan bagian dari *framework* yang bertugas sebagai penyedia fungsi-fungsi yang berhubungan dengan *database* seperti *insert, update, delete*, dan lain sebagainya. *Model* pada *framework* tidak dapat terhubung langsung dengan *view*, namun akan berhubungan dengan *controller*.

B. View

View merupakan bagian dari *framework* yang bertugas sebagai pembentuk tampilan perangkat lunak. Untuk mengakses *view* dibutuhkan konfigurasi pada bagian *controller* sehingga pada akses langsung pada perangkat lunak *controller* akan berjalan terlebih dahulu untuk menampilkan *view*.

C. Controller

Controller merupakan bagian dari *framework* yang bertugas sebagai jembatan antara *view* dan *model*. *Controller* merupakan wadah pengguna untuk melakukan interaksi dengan aplikasi sehingga pada *controller* terdapat fungsi-fungsi proses untuk menentukan apakah yang akan diproses dan yang ditampilkan pada pengguna.

2.9.4.2 Pengujian

Setelah proses implementasi selesai maka terdapat proses pengujian atau *test* pada perangkat lunak yang telah selesai diimplementasikan. Pengujian perangkat lunak ialah proses formal yang berisi pengujian khusus yang meliputi pengujian unit perangkat lunak, perangkat lunak yang terintegrasi atau keseluruhan perangkat lunak yang dijalankan pada komputer. Seluruh pengujian dijalankan berdasarkan prosedur pengujian untuk mengetahui penerimaan dari suatu kondisi yang dijalankan (Galin, 2004). Terdapat beberapa alasan dilakukannya pengujian suatu perangkat lunak diantaranya untuk memastikan bahwa perangkat lunak yang telah dikembangkan telah sesuai dengan perancangan yang telah dilakukan. Selain itu, pengujian dilakukan untuk memastikan perangkat lunak yang telah berjalan telah valid atau dapat digunakan dalam berbagai skenario pengguna yang berbeda. Untuk pengujian pada pengembangan sistem informasi Lembaga Sertifikasi Profesi SMK Negeri 4 Malang menggunakan pengujian *Black-box* dan *user acceptance testing* (UAT).

A. *Black-box Testing*

Pengertian pengujian *Black-box* menurut Rosa & Shalahuddin (2015) menyatakan bahwa pengujian *Black-box* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dilakukan untuk mengetahui apakah fungsi – fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Untuk teknik yang digunakan dalam pengujian *Black-box testing* adalah *error guessing*. Dalam teknik *error guessing*, pengujian didesain oleh penguji yang mencoba menemukan kesalahan yang mayoritas mungkin terdapat pada program. Informasi (dari pengujian) yang bagus adalah kesalahan-kesalahan yang ditemukan pada awal proyek dan tentu saja keahlian penguji (Bertolino, 2001). Dari pengertian tersebut, dapat diketahui bahwa teknik *error guessing* adalah pengujian yang bergantung pada penguji tersebut untuk menebak atau mencari kesalahan yang terdapat pada perangkat lunak yang diuji. Kesalahan tersebut diantaranya bilangan dibagi dengan 0 (nol), menginputkan karakter kosong pada *fields*, menekan tombol *submit* tanpa mengisi nilai, mengunggah data melebihi kapasitas maksimal yang ditentukan, serta parameter yang salah (Software Testing Help, 2019).

Setelah dilakukan pengujian didapatkan kesimpulan bahwa pengujian *error guessing* apakah mendapatkan hasil sesuai dengan hasil yang diharapkan. Apabila hasil pengujian sesuai dengan hasil yang diharapkan maka pengujian mendapatkan hasil valid pada fungsional yang diujikan. Apabila tidak, maka pengujian mendapatkan hasil tidak valid pada fungsional yang dilakukan pengujian.

B. *User Acceptance Testing (UAT)*

Selain pengujian fungsional, Untuk pengujian *Black-box* juga menggunakan metode *user acceptance testing* (UAT). UAT menurut jurnal internasional yang berjudul “*Acceptance Testing Technique: A Survey along with Its Operating Frameworks*” (Pathak, 2016) menyatakan bahwa *UAT* adalah pengujian yang dilakukan untuk menjamin bahwa kerangka kerja yang dibuat memenuhi semua persyaratan klien. Pengujian ini adalah pengujian terakhir yang dilakukan klien dalam hal persyaratan, akurasi dan konsistensi produk yang dibuat. Tujuan utama *testing* tersebut ialah untuk mendeteksi semua kemungkinan kesalahan program yang terjadi saat digunakan oleh klien pada masa yang akan datang.

Untuk proses pengujian *user acceptance testing* (UAT) menggunakan kuesioner yang akan diberikan kepada pengguna aplikasi. Pengguna aplikasi tersebut akan menentukan fitur yang dibuat pada sistem apakah telah memenuhi persyaratan yang diberikan. Apabila semua fitur dapat diterima dan dinyatakan oleh pengguna fitur tersebut berjalan baik, maka dari *user acceptance testing*

tersebut dinyatakan berhasil dan program telah layak untuk digunakan. Contoh kuesioner *user acceptance testing* dapat dilihat pada tabel 2.6 didapatkan dari penelitian pengembangan sistem informasi *E-Complaint Management* (Evantio, 2017).

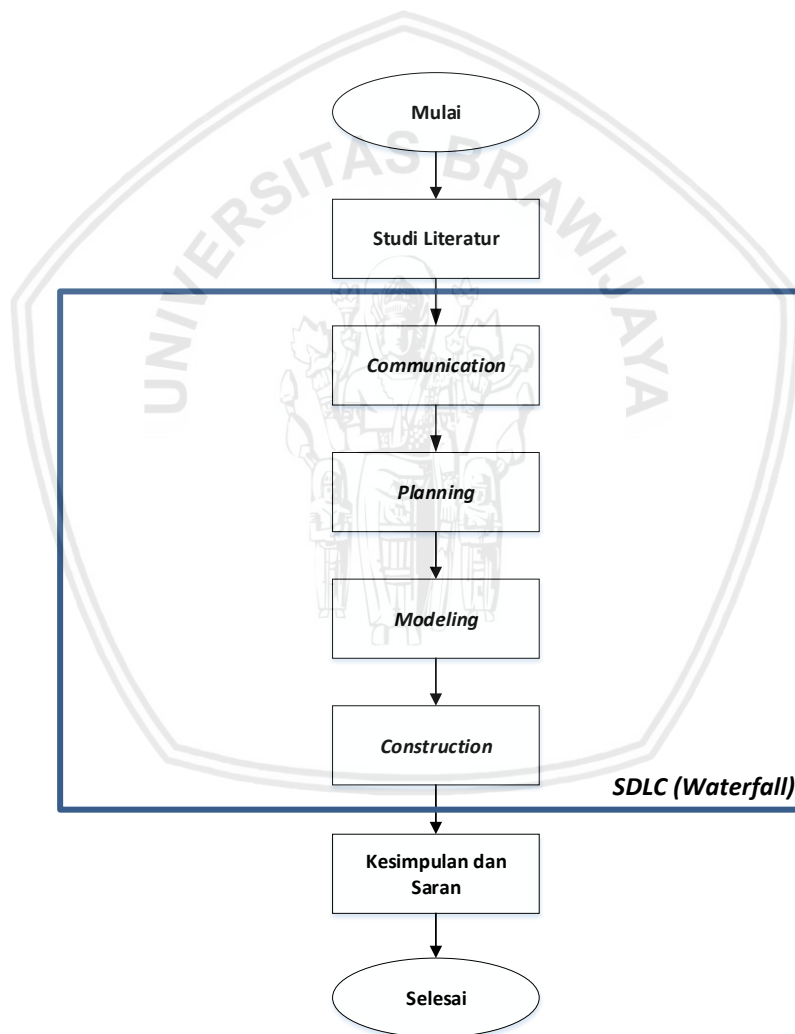
Tabel 2.6 Contoh Kuesioner User Acceptance Testing (Evantio, 2017)

No.	Pertanyaan	Hasil		Komentar
		Ya	Tidak	
1	Sistem Informasi E-Complaint Management berjalan sesuai dengan kebutuhan yang didefinisikan			
2	Proses pencatatan komplain oleh sistem berjalan dengan baik			
3	Pemberian informasi update pekerjaan komplain berjalan dengan baik			
4	Informasi yang diberikan oleh Sistem Informasi E-Complaint Management mudah dimengerti			
5	Informasi yang dimuat dalam dokumen rekap berita acara pekerjaan komplain cukup lengkap			

Setelah seluruh pengujian telah dijalankan dan jawaban telah didapatkan, maka dilakukan "*acceptance decision*" (pengambilan keputusan). Hal tersebut juga disebut "*Go/No-Go decision*" (Keputusan lanjut atau tidak lanjut). Jika pengguna telah puas maka diambil keputusan lanjut, jika tidak maka diambil keputusan tidak lanjut (Software Testing Help, 2019) Pada proses penilaian UAT dilakukan dengan melihat hasil kuesioner yang diisi oleh pengguna aplikasi. Apabila seluruh pertanyaan menghasilkan jawaban "ya", maka dapat ditarik kesimpulan bahwa seluruh fungsional telah dapat diterima pengguna dan dapat dijalankan dengan baik. Namun apabila terdapat beberapa jawaban "tidak", maka dapat ditarik kesimpulan bahwa aplikasi masih terdapat kekurangan yang mengganggu fungsional aplikasi. Sedangkan apabila seluruh pertanyaan mendapatkan jawaban "tidak" maka dapat disimpulkan bahwa aplikasi belum dapat diterima oleh responden atau *stakeholder* yang menggunakan aplikasi dan harus dilakukan perbaikan atau pengembangan ulang.

BAB 3 METODOLOGI

Pada bab metodologi menjelaskan tentang langkah-langkah dalam proses penyelesaian masalah dengan urutan langkah-langkah seperti pada gambar 3.1. Pada gambar tersebut menjelaskan bahwa langkah-langkah penelitian implementatif dimulai dengan studi literatur, *communication*, *planning*, *modeling*, *construction*, dan pengambilan kesimpulan serta saran. Tidak digunakannya proses *deployment* pada proses pengembangan perangkat lunak karena tidak diperlukannya pengembangan dan evaluasi perangkat lunak berdasarkan umpan balik bagi pengguna pada proses penelitian.



Gambar 3.1 Langkah-langkah Pemecahan Masalah

3.1 Studi Literatur

Studi literatur merupakan proses untuk mencari referensi teori relevan dengan penelitian yang akan dilakukan. Referensi relevan yang digunakan sebagai bahan pertimbangan dalam melakukan penelitian ini ialah penelitian skripsi yang telah dilakukan sebelumnya yang mencakup penelitian pada sekolah, penelitian menggunakan *SDLC Waterfall*, serta penelitian yang menggunakan *Object Oriented Programming* dalam penyelesaian masalahnya. Dari referensi penelitian tersebut, dapat diambil manfaat yakni untuk memperkuat dasar teori dalam melakukan penelitian serta dapat menjadi dasar dalam penyelesaian masalah yang dilakukan selama penelitian.

Untuk literatur yang digunakan diantaranya adalah buku “Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek” dari Rosa & Shalahuddin (2015) yang membahas tentang pemodelan perangkat lunak berorientasi objek. Dari buku tersebut bermanfaat dalam pemodelan perangkat lunak menggunakan *Unified Modeling language (UML)*, diantaranya *class diagram*, *use case diagram*, *sequence diagram*, *collaboration diagram*, dan *activity diagram*. Selain itu, terdapat jurnal berjudul “Acceptance Testing Technique: A Survey along with Its Operating Frameworks” (Pathak, 2016). Dalam jurnal tersebut menjelaskan apa itu *User Acceptance Testing (UAT)* dan bagaimana melakukan *UAT* yang baik.

3.2 Communication

Pada proses *communication* digunakan untuk mengetahui akar permasalahan yang ada di lapangan. Metode yang digunakan adalah wawancara dengan salah satu *stakeholder* atau guru yang telah mempunyai lisensi dari Lembaga Sertifikasi Profesi dibawah naungan Badan Nasional Sertifikasi Profesi (kerangka wawancara dapat dilihat pada lampiran B). Hasil wawancara pertama digunakan untuk memperjelas latar belakang permasalahan sehingga masalah yang didefinisikan jelas dan dapat diselesaikan dengan baik. Tidak hanya itu, wawancara digunakan untuk mendapatkan urgensi dikembangkannya sistem informasi asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang. Selain dengan teknik wawancara, dalam perumusan masalah penulis melakukan observasi langsung pada jurusan Rekayasa Perangkat Lunak di SMK Negeri 4 Malang guna untuk mengetahui secara langsung bahwa dengan kondisi permasalahan yang ada sekarang dapat diselesaikan menggunakan sistem informasi yang dikembangkan oleh peneliti.

Dari hasil wawancara dan observasi tersebut digunakan untuk proses penggalan kebutuhan perangkat lunak. Penggalan atau analisis kebutuhan penting dilakukan untuk mendefinisikan apa saja yang harus ada dalam perangkat lunak yang akan dikembangkan. Selain itu, penggalan kebutuhan berfungsi sebagai awal dari perancangan perangkat lunak yang akan dikembangkan. Hasil proses analisis kebutuhan tersebut kemudian digambarkan pada *use case diagram*.

3.3 Planning

Planning merupakan proses untuk merencanakan hal-hal apa saja yang akan dilakukan pada proses pengembangan perangkat lunak. Dalam proses planning terdapat beberapa hal yang akan direncanakan, diantaranya waktu yang diperlukan untuk pengembangan perangkat lunak serta estimasi waktu yang diperlukan untuk menjalankan masing-masing proses dalam pengembangan perangkat lunak.

3.4 Modeling

Pada proses *modeling* merupakan proses penggambaran analisis kebutuhan akan digambarkan pada suatu rancangan aplikasi dengan jelas sehingga pengembang akan dapat mudah mengimplementasikan proses kerja perangkat lunak atau mekanisme kerja perangkat lunak pada kode-kode program. Penggambaran rancangan sistem digambarkan pada beberapa model diantaranya *collaboration diagram*, *sequence diagram*, *class diagram*, dan *physical data model*. *Collaboration diagram* digunakan untuk mengetahui kelas-kelas yang akan diimplementasikan pada sistem. *Sequence diagram* digambarkan sebagai alur jalan suatu proses atau fungsi pada perangkat lunak mulai dari awal hingga akhir hingga mendapatkan tujuan yang diinginkan. Sedangkan *class Diagram* digambarkan untuk mengetahui hubungan antar kelas yang akan diimplementasikan pada sistem baik kelas *controller* dan kelas *model*.

3.5 Construction

Proses construction merupakan proses implementasi dan pengujian perangkat lunak dari hasil *modeling* yang didapatkan. Proses implementasi merupakan penulisan kode program dari perancangan yang sebelumnya dibuat. Implementasi pada penyelesaian laporan ini menggunakan *framework PHP* yakni *CodeIgniter* sehingga akan menghemat penggunaan kode program serta dapat membuat sebuah alur proses yang dapat dicermati dengan mudah sehingga jika terdapat kesalahan pada kode program pengembang dengan mudah akan dapat menemukan kesalahan kode program pada bagian mana.

Proses pengujian dilakukan menggunakan metode *Black-box* dan *user acceptance testing* (UAT) yakni percobaan perangkat lunak apakah telah berhasil memenuhi kebutuhan perangkat lunak yang didefinisikan sebelumnya mulai dari *input* hingga *output* perangkat lunak serta melihat sejauh mana penerimaan aplikasi oleh pengguna atau *stakeholder*. Untuk skenario pengujian ialah responden akan menggunakan program pada bagiannya masing-masing dan akan mengisi kuesioner yang diberikan. Kuesioner tersebut berisi pernyataan mengenai penerimaan aplikasi bagi pengguna.

Pengujian *Black-box* digunakan untuk melihat apakah seluruh fungsional yang telah didefinisikan telah diimplementasikan dan berjalan dengan baik. Sedangkan untuk melihat sejauh mana penerimaan pengguna digunakan metode UAT, yakni

user acceptance testing pada asesor dan siswa. Pengujian tersebut dengan memberikan beberapa pertanyaan terkait aplikasi yang telah dikembangkan. Dari hasil jawaban tersebut dapat diketahui bahwa aplikasi tersebut telah sesuai atau telah dapat diterima oleh pengguna maupun *stakeholder* aplikasi. Skenario yang diujikan yakni peneliti menebak kemungkinan *error* yang akan terjadi (*error guessing*) pada 4 (empat) fungsional utama sistem dan mendefinisikan keluaran yang diharapkan. Kemudian, kemungkinan *error* tersebut diujikan pada aplikasi secara langsung. Apabila keluaran program sesuai yang diharapkan maka hasil yang diharapkan adalah "valid" dan menginformasikan bahwa pengujian telah berhasil dilaksanakan dan menginformasikan bahwa kemungkinan *error* telah dapat tertangani dengan baik.

Setelah dilakukan percobaan, maka dari hasil percobaan tersebut dapat dilakukan proses analisis untuk menentukan apakah dengan perangkat lunak yang diuji telah layak digunakan di lapangan ataupun belum. Selain itu, analisis digunakan untuk menentukan bahwa perangkat lunak telah memenuhi semua kebutuhan sistem yang telah dijelaskan pada analisis kebutuhan dan perancangan sistem.

3.6 Pengambilan Kesimpulan dan Saran

Pada langkah terakhir ialah pengambilan kesimpulan dan saran. Kesimpulan yang didapat merupakan rangkuman atau hasil yang telah didapat mulai dari awal proses penyelesaian masalah hingga akhir. Dari hasil kesimpulan yang didapat berguna untuk digunakan pada penelitian selanjutnya dan untuk melihat apakah dari proses penyelesaian masalah tersebut telah sesuai dengan kebutuhan pengguna. Selain itu, terdapat saran yakni opini penulis terhadap proses penelitian maupun hasil pemecahan masalah sehingga dari opini tersebut dapat digunakan sebagai pertimbangan untuk pengembangan perangkat lunak lain yang serupa.

BAB 4 HASIL DAN PEMBAHASAN

Pada bab rekayasa kebutuhan menjelaskan tentang pernyataan masalah yang lebih mendetail dari penjelasan permasalahan pada bab pendahuluan. Selain itu, pada bab ini juga mengidentifikasi aktor dalam sistem serta perancangan kebutuhan sistem untuk digunakan dalam implementasi perangkat lunak.

4.1 Analisis Kebutuhan

4.1.1 Analisis Permasalahan

Proses analisis kebutuhan merupakan salah satu pada tahap pada proses *communication* dari SDLC *waterfall* yang digunakan dalam proses pengembangan perangkat lunak. Dari analisis permasalahan yang ada, didapatkan beberapa cara dalam langkah awal proses pemecahan masalah, yakni mengidentifikasi pemangku kepentingan atau pengguna dari sistem yang akan dibuat. Dari proses identifikasi tersebut, didapatkan 2 (dua) aktor yakni siswa pemohon sertifikasi dan guru yang telah mendapatkan lisensi sebagai asesor berdasarkan dokumen asesmen (lampiran A). Setelah pengguna sistem telah teridentifikasi, dilanjutkan dengan melakukan perancangan kebutuhan sistem. Kebutuhan fungsional ialah kebutuhan yang harus ada pada sistem agar sistem dapat berjalan dengan baik. Dari hasil analisis permasalahan, didapatkan 10 (sepuluh) kebutuhan fungsional.

Langkah selanjutnya ialah mendefinisikan kebutuhan program dalam pemodelan *use case* yakni dengan menggambarkan proses kerja sistem dalam *use case diagram* dari 10 (sepuluh) kebutuhan fungsional yang telah didapatkan dari proses identifikasi kebutuhan sebelumnya dan menjabarkan proses tersebut pada *use case scenario*. Pada *use case scenario* terdapat beberapa kondisi yang ada pada sistem baik sebelum atau sesudah pengguna melakukan suatu proses. Langkah terakhir yang dilakukan ialah menggambarkan proses yang lebih mendetail pada *activity diagram* untuk mengetahui langkah detail pengguna melakukan suatu proses dalam sistem.

4.1.2 Identifikasi Pengguna

Dalam proses identifikasi aktor atau pengguna sistem yang didapatkan dari analisis dokumen asesmen Lembaga Sertifikasi Profesi (LSP) SMK Negeri 4 Malang (lampiran A), ditemukan dua pengguna atau dua aktor dalam proses penggunaan sistem yakni siswa sebagai pemohon sertifikasi serta guru yang telah mendapatkan lisensi menjadi asesor dari Badan Nasional Sertifikasi Profesi (BNSP). Pengguna pertama yakni siswa yang berperan sebagai pemohon sertifikasi memiliki kepentingan untuk mendaftar sertifikasi serta melengkapi berkas-berkas pemohon sertifikasi dan mengisi lembar asesmen mandiri.

Sedangkan guru yang menjadi asesor berperan sebagai pemeriksa berkas-berkas pemohon sertifikasi serta melakukan asesmen terhadap hasil pekerjaan

dari tes yang diberikan kepada siswa dan mendata hasil asesmen tersebut dalam perangkat lunak. Selain itu, asesor (guru yang mendapatkan lisensi dari BNSP) juga dapat mencetak dokumen digital ke dalam kertas apabila diperlukan.

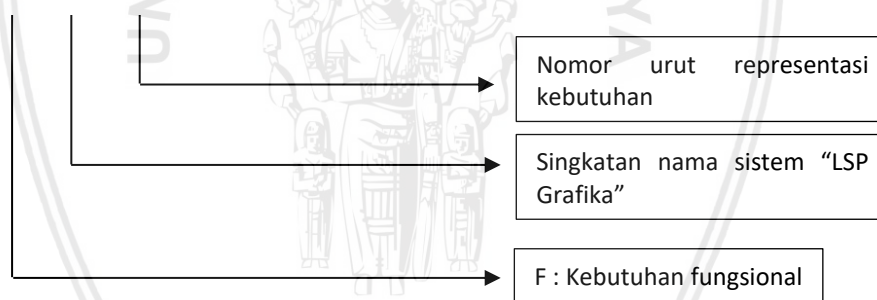
Dari identifikasi aktor di atas, dapat diketahui bahwa sistem digunakan oleh 2 (dua) aktor saja dan tanpa halaman publikasi kepada orang-orang yang tidak berkepentingan dalam proses sertifikasi profesi oleh Lembaga Sertifikasi Profesi (LSP) SMK Negeri 4 Malang.

4.1.3 Daftar Spesifikasi Kebutuhan

Pada spesifikasi kebutuhan dilakukan analisis kebutuhan fungsional dan yang akan dikembangkan dalam sistem. Kebutuhan fungsional merupakan kebutuhan utama dalam sistem sehingga sistem dapat bekerja dengan baik.

Kebutuhan fungsional adalah fungsionalitas atau hal apa saja yang dapat dilakukan oleh aktor kepada sistem. Kebutuhan fungsional didapatkan dari hasil analisis permasalahan dan analisis kebutuhan dari aktor-aktor yang telah diidentifikasi. Selain itu, kebutuhan fungsional didapatkan juga melalui masukan-masukan dari para *stakeholder* dengan menggunakan metode wawancara. Untuk penomoran kode dasar kebutuhan fungsional dapat dilihat pada Gambar 4.1.

KODE : F-LSPG-01



Gambar 4.1 Aturan Penomoran Kebutuhan Fungsional

Berikut pada tabel 4.1 merupakan hasil analisis kebutuhan fungsional:

Tabel 4.1 Identifikasi Kebutuhan Fungsional

No.	Kode	Deskripsi
1	F-LSPG-01	Sistem menyediakan <i>form</i> untuk pendaftaran pemohon sertifikasi oleh siswa
2	F-LSPG-02	Sistem menyediakan fitur untuk melakukan pengunggahan berkas pendukung pemohon sertifikasi oleh siswa
3	F-LSPG-03	Sistem menyediakan fitur untuk melakukan proses asesmen mandiri oleh siswa

Tabel 4.1 Identifikasi Kebutuhan Fungsional (lanjutan)

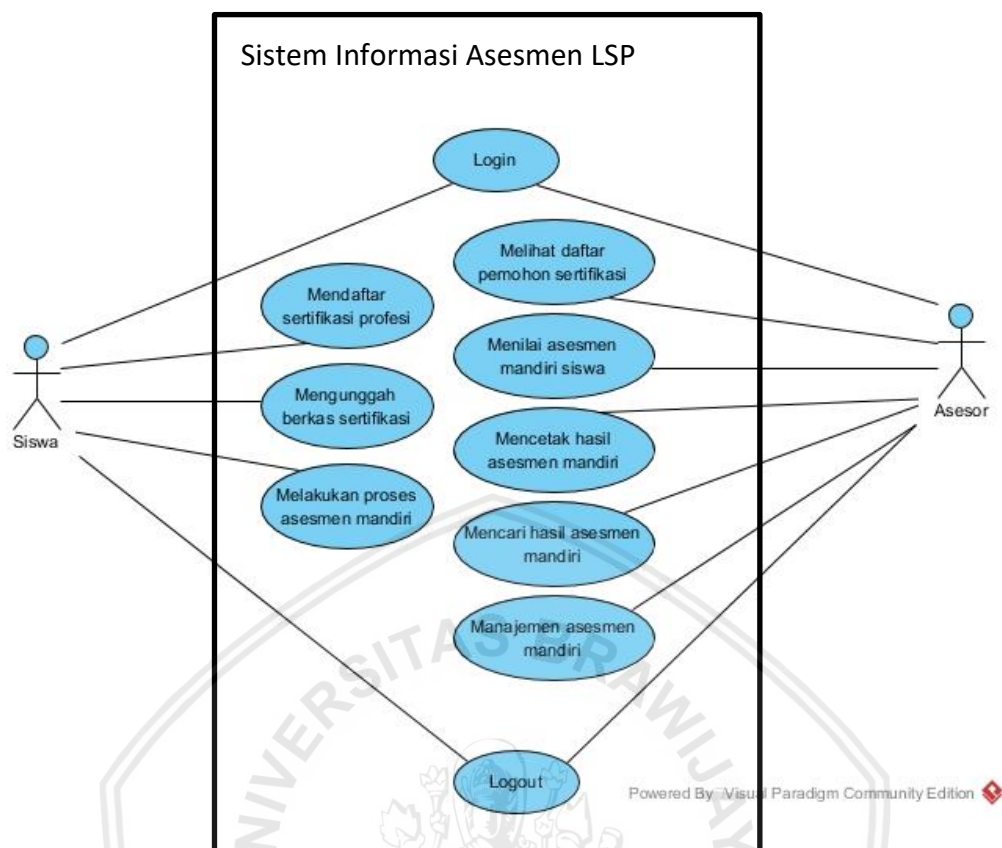
No.	Kode	Deskripsi
4	F-LSPG-04	Sistem dapat menampilkan daftar pemohon sertifikasi bagi asesor
5	F-LSPG-05	Sistem menyediakan fitur untuk melakukan penilaian asesmen mandiri bagi asesor
6	F-LSPG-06	Sistem dapat melakukan pencetakan hasil asesmen mandiri bagi asesor
7	F-LSPG-07	Sistem dapat melakukan pencarian hasil asesmen mandiri oleh asesor
8	F-LSPG-08	Fitur <i>login</i> untuk pengguna masuk ke dalam sistem sebagai siswa maupun asesor
9	F-LSPG-09	Fitur <i>logout</i> untuk pengguna keluar dari sistem untuk siswa dan asesor
10	F-LSPG-10	Fitur manajemen asesmen untuk asesor mengatur manajemen asesmen mandiri yang meliputi skema, unit, elemen, dan pertanyaan

4.1.4 Pemodelan *Use Case*

4.1.4.1 *Diagram Use Case*

Pada pemodelan *use case* digunakan untuk menggambarkan atau menampilkan hubungan antara fungsionalitas sistem masing-masing aktor. Pemodelan *use case* dapat dilihat pada Gambar 4.2. Pada gambar 4.2 adalah *use case diagram* dari sistem informasi asesmen Lembaga Sertifikasi Profesi (LSP) beserta aktor atau pengguna yang terdapat dalam sistem tersebut. Terdapat 10 (sepuluh) fungsional dan 2 (dua) aktor atau pengguna yang berinteraksi dengan sistem. Kebutuhan fungsional didapatkan dari identifikasi kebutuhan fungsional tabel 4.1 dan aktor atau pengguna didapatkan dari identifikasi pengguna sebelumnya.

Dari gambar 4.2 pula didapatkan keterangan bahwa kedua aktor harus melalui proses *login* untuk dapat mengakses sistem. Namun, apabila pengguna adalah pemohon atau siswa yang belum mendaftar, maka terdapat fitur untuk mendaftar sebagai pemohon sertifikasi profesi. Selain melakukan pendaftaran, siswa dapat melakukan proses *upload* berkas sertifikasi dan melakukan proses asesmen mandiri.



Gambar 4.2 Use Case Sistem Informasi Asesmen LSP

Sedangkan untuk asesor dapat melihat daftar pemohon sertifikasi, menilai asesmen mandiri siswa, mencetak hasil asesmen, dan mencari hasil asesmen yang telah dilakukan. Kedua aktor tersebut mempunyai fitur untuk *logout* yakni keluar dari sistem.

4.1.4.2 Use Case Scenario

Use case scenario merupakan penjelasan dari berbagai kebutuhan fungsional yang telah digambarkan dalam diagram *use case*. Dalam *use case scenario* terdapat penjelasan mengenai alur fungsional, *precondition*, *postcondition*, dan *alternative flow*. Berikut merupakan *use case scenario* masing-masing *use case*.

A. Use Case Scenario Mendaftar Sertifikasi Profesi

Berikut pada tabel 4.2 adalah penjelasan dari *use case scenario* mendaftar sertifikasi profesi.

Tabel 4.2 Use Case Scenario Mendaftar Sertifikasi Profesi

Item	Deskripsi
<i>Use case ID</i>	UC-LSPG-01
<i>Use case Name</i>	Mendaftar Sertifikasi Profesi

Tabel 4.2 *Use Case Scenario* Mendaftar Sertifikasi Profesi (Lanjutan)

Item	Deskripsi
<i>Functional ID</i>	F-LSPG-01
<i>Actor</i>	Siswa
<i>Description</i>	Siswa mendaftar sertifikasi profesi untuk mengikuti asesmen oleh Lembaga Sertifikasi Profesi (LSP)
<i>Use Goal</i>	Siswa terdaftar sebagai peserta sertifikasi profesi
<i>Preconditions</i>	Adanya koneksi internet dan siswa belum pernah mendaftar sebagai pemohon sertifikasi profesi
<i>Main flow</i>	1. Siswa memilih menu <i>register</i> pada halaman awal sistem informasi 2. Sistem menampilkan <i>form</i> pendaftaran 3. Siswa mengisi informasi utama pada <i>form</i> pendaftaran untuk mendaftar sebagai peserta sertifikasi profesi 4. Siswa menekan tombol <i>register</i> untuk mendaftar 5. Sistem menyimpan data pendaftaran siswa 6. Siswa mendapatkan <i>email</i> notifikasi pendaftaran berhasil 7. Sistem menampilkan informasi pendaftaran berhasil
<i>Alternative flow</i>	1. Jika terdapat data yang kosong, sistem akan menampilkan peringatan pada <i>field</i> yang kosong agar pengguna dapat mengisi <i>field</i> tersebut
<i>Postconditions</i>	Data pendaftaran sertifikasi profesi tersimpan dalam <i>database</i>

B. *Use Case Scenario* Mengunggah Berkas Sertifikasi

Berikut pada tabel 4.3 adalah penjelasan dari *use case scenario* mengunggah Berkas Sertifikasi.

Tabel 4.3 *Use Case Scenario* Mengunggah Berkas Sertifikasi

Item	Deskripsi
<i>Use case ID</i>	UC-LSPG-02
<i>Use case Name</i>	Mengunggah Berkas Sertifikasi
<i>Functional ID</i>	F-LSPG-02
<i>Actor</i>	Siswa

Tabel 4.3 *Use Case Scenario* Mengunggah Berkas Sertifikasi (Lanjutan)

Item	Deskripsi
<i>Use case ID</i>	UC-LSPG-02
<i>Use case Name</i>	Mengunggah Berkas Sertifikasi
<i>Functional ID</i>	F-LSPG-02
<i>Actor</i>	Siswa
<i>Description</i>	Siswa melakukan pengunggahan berkas pendukung seperti foto dan <i>scan</i> rekapitulasi nilai.
<i>Use Goal</i>	Berkas siswa dapat terunggah ke dalam server.
<i>Preconditions</i>	Siswa telah melakukan proses <i>login</i>
<i>Main flow</i>	1. Siswa memilih menu berkas 2. Sistem menampilkan <i>form upload</i> berkas-berkas siswa 3. Siswa melakukan <i>upload</i> berkas-berkas pada <i>form</i> yang disediakan sistem 4. Siswa menekan tombol <i>upload</i> untuk melakukan proses <i>upload</i> 5. Sistem menyimpan berkas siswa 6. Sistem menampilkan informasi <i>upload</i> berhasil
<i>Alternative flow</i>	1. Jika <i>file</i> siswa melebihi ukuran 1 <i>Megabytes</i>, sistem akan menampilkan peringatan <i>file</i> tidak boleh melebihi ukuran 1 <i>Megabytes</i>
<i>Postconditions</i>	Berkas pendukung sertifikasi siswa tersimpan dalam <i>database</i>

C. *Use Case Scenario* Melakukan Proses Asesmen Mandiri

Berikut pada tabel 4.4 adalah penjelasan dari *use case scenario* melakukan proses asesmen mandiri.

Tabel 4.4 *Use Case Scenario* Melakukan Proses Asesmen Mandiri

Item	Deskripsi
<i>Use case ID</i>	UC-LSPG-03
<i>Use case Name</i>	Melakukan Proses Asesmen Mandiri

Tabel 4.4 Use Case Scenario Melakukan Proses Asesmen Mandiri (Lanjutan)

Item	Deskripsi
<i>Functional ID</i>	F-LSPG-03
<i>Actor</i>	Siswa
<i>Description</i>	Siswa melakukan proses asesmen mandiri yang merupakan bagian dari lembar asesmen sertifikasi profesi
<i>Use Goal</i>	Siswa dapat mengisi lembar asesmen mandiri
<i>Preconditions</i>	Siswa telah melakukan proses <i>login</i> dan siswa telah mengerjakan uji kompetensi yang disediakan
<i>Main flow</i>	1. Siswa memilih menu asesmen mandiri 2. Sistem menampilkan lembar asesmen mandiri yang akan diisi oleh siswa 3. Siswa melakukan proses asesmen dengan mengisi <i>checklist</i> Kompeten (K) atau Belum Kompeten (BK) serta bukti pendukung 4. Siswa menekan tombol selesai untuk mengakhiri proses asesmen mandiri 5. Sistem menyimpan data hasil asesmen mandiri 6. Sistem menampilkan informasi asesmen mandiri selesai
<i>Alternative flow</i>	1. Jika terdapat data yang kosong, sistem akan menampilkan peringatan pada <i>field</i> yang kosong agar pengguna dapat mengisi <i>field</i> tersebut 2. Jika siswa masih belum disetujui atau diverifikasi oleh asesor, maka menu asesmen mandiri tidak tersedia dan kembali ke halaman awal
<i>Postconditions</i>	Data hasil asesmen mandiri siswa dapat tersimpan dalam <i>database</i>

D. Use Case Scenario Melihat Daftar Pemohon Sertifikasi

Berikut pada tabel 4.5 adalah penjelasan dari *use case scenario* melihat daftar pemohon sertifikasi.

Tabel 4.5 Use Case Scenario Melihat Daftar Pemohon Sertifikasi

Item	Deskripsi
<i>Use case ID</i>	UC-LSPG-04
<i>Use case Name</i>	Melihat Daftar Pemohon Sertifikasi
<i>Functional ID</i>	F-LSPG-04
<i>Actor</i>	Asesor
<i>Description</i>	Asesor dapat melihat daftar pemohon sertifikasi yang telah mendaftar sertifikasi profesi
<i>Use Goal</i>	Asesor dapat melihat daftar pemohon sertifikasi dan melihat kecukupan berkas yang telah diisi oleh pemohon sertifikasi (siswa)
<i>Preconditions</i>	Asesor telah melakukan proses <i>login</i>
<i>Main flow</i>	1. Asesor memilih menu daftar siswa 2. Sistem mengambil data siswa yang telah mendaftar dari <i>database</i> 3. Sistem menampilkan daftar siswa yang telah mendaftar beserta dokumen siswa yang telah di <i>upload</i>
<i>Alternative flow</i>	-
<i>Postconditions</i>	Data siswa yang telah mendaftar sertifikasi profesi dapat ditampilkan

E. Use Case Scenario Menilai Asesmen Mandiri Siswa

Berikut pada tabel 4.6 adalah penjelasan dari *use case scenario* menilai asesmen mandiri siswa.

Tabel 4.6 Use Case Scenario Menilai Asesmen Mandiri Siswa

Item	Deskripsi
<i>Use case ID</i>	UC-LSPG-05
<i>Use case Name</i>	Menilai Asesmen Mandiri Siswa
<i>Functional ID</i>	F-LSPG-05
<i>Actor</i>	Asesor
<i>Description</i>	Asesor dapat melakukan proses penilaian asesmen pada siswa yang telah melengkapi data diri serta dokumen pendukung sertifikasi profesi serta telah mengerjakan uji kompetensi yang diberikan

Tabel 4.6 *Use Case Scenario* Melakukan Proses Asesmen (Lanjutan)

Item	Deskripsi
<i>Use Goal</i>	Asesor dapat mengisi lembar asesmen melalui <i>form</i> yang telah disediakan sistem
<i>Preconditions</i>	Asesor telah melakukan proses <i>login</i> dan siswa yang akan dilakukan asesmen telah mengerjakan uji kompetensi yang diberikan
<i>Main flow</i>	1. Asesor memilih menu daftar siswa 2. Sistem mengambil data siswa yang telah mendaftar dari <i>database</i> 3. Sistem menampilkan daftar siswa yang telah mendaftar beserta dokumen siswa yang telah di <i>upload</i> 4. Asesor memilih menu asesmen pada salah satu siswa yang telah melengkapi berkas serta telah mengisi asesmen mandiri dan diverifikasi oleh asesor 5. Sistem menampilkan <i>form-form</i> pada masing-masing unit kompetensi untuk dilakukan asesmen 6. Asesor mengisi lembar asesmen yang ditampilkan sistem dalam <i>form-form</i> dengan melakukan <i>checklist</i> pada keterangan V (valid), A (asli), T (terkini) atau M (mandiri). 7. Asesor menekan tombol selesai setelah melakukan pengisian seluruh <i>form</i> asesmen yang diberikan 8. Sistem akan menyimpan data hasil asesmen ke dalam <i>database</i> 9. Sistem mengirimkan <i>e-mail</i> pada siswa bahwa asesmen mandiri telah dinilai oleh asesor 10. Sistem akan menampilkan informasi bahwa asesmen telah selesai dilakukan
<i>Alternative flow</i>	1. Jika terdapat data yang kosong, sistem akan menampilkan peringatan pada <i>field</i> yang kosong agar pengguna dapat mengisi <i>field</i> tersebut
<i>Postconditions</i>	Data hasil asesmen yang dilakukan asesor kepada siswa dapat tersimpan ke dalam <i>database</i>

F. Use Case Scenario Mencetak Hasil Asesmen Mandiri

Berikut pada tabel 4.7 adalah penjelasan dari *use case scenario* mencetak hasil asesmen mandiri.

Tabel 4.7 Use Case Scenario Mencetak Hasil Asesmen Mandiri

Item	Deskripsi
<i>Use case ID</i>	UC-LSPG-06
<i>Use case Name</i>	Mencetak Hasil Asesmen Mandiri
<i>Functional ID</i>	F-LSPG-06
<i>Actor</i>	Asesor
<i>Description</i>	Asesor dapat melakukan proses pencetakan dokumen dari hasil asesmen yang telah dilakukan
<i>Use Goal</i>	Dokumen asesmen dapat tercetak pada kertas
<i>Preconditions</i>	Asesor telah melakukan proses <i>login</i> dan telah melakukan proses asesmen
<i>Main flow</i>	1. Asesor memilih menu daftar siswa 2. Sistem mengambil data siswa yang telah mendaftar dari <i>database</i> 3. Sistem menampilkan daftar siswa yang telah mendaftar beserta dokumen siswa yang telah di <i>upload</i> 4. Asesor memilih dokumen yang akan dicetak pada siswa yang telah dilakukan proses asesmen 5. Sistem mengambil data hasil asesmen siswa tertentu yang telah dilakukan proses asesmen dari <i>database</i> 6. Sistem menampilkan pada halaman web hasil asesmen siswa tertentu yang akan dicetak 7. Sistem melakukan proses pencetakan melalui fitur cetak yang ada pada <i>browser</i>
<i>Alternative flow</i>	1. Jika dokumen yang akan dicetak belum tersedia, maka sistem akan menampilkan peringatan "Dokumen belum tersedia" dan membatalkan proses cetak
<i>Postconditions</i>	Data hasil asesmen yang telah dilakukan dapat tercetak ke dalam dokumen fisik

G. Use Case Scenario Mencari Hasil Asesmen Mandiri

Berikut pada tabel 4.8 adalah penjelasan dari *use case scenario* mencari hasil asesmen mandiri.

Tabel 4.8 Use Case Scenario Mencari Hasil Asesmen Mandiri

Item	Deskripsi
<i>Use case ID</i>	UC-LSPG-07
<i>Use case Name</i>	Mencari Hasil Asesmen Mandiri
<i>Functional ID</i>	F-LSPG-07
<i>Actor</i>	Asesor
<i>Description</i>	Asesor dapat melakukan pencarian terhadap data hasil asesmen siswa
<i>Use Goal</i>	Data hasil asesmen siswa tertentu dapat ditampilkan
<i>Preconditions</i>	Asesor telah melakukan proses <i>login</i> dan telah melakukan proses asesmen
<i>Main Flow</i>	1. Sistem mengambil data siswa yang telah mendaftar dari <i>database</i> 2. Sistem menampilkan daftar siswa yang telah mendaftar beserta dokumen siswa yang telah di <i>upload</i> 3. Asesor mengetikkan nama maupun kelas untuk dilakukan pencarian data hasil asesmen 4. Sistem mengambil data hasil asesmen siswa tertentu yang telah dilakukan proses asesmen dari <i>database</i> berdasarkan kata kunci yang diketikkan asesor 5. Sistem menampilkan pada halaman web data hasil asesmen siswa tertentu berdasarkan kata kunci yang diketikkan asesor
<i>Alternative flow</i>	1. Apabila asesor mengetikkan data kosong untuk dilakukan pencarian, maka sistem akan tetap menampilkan semua data yang ada 2. Jika asesor mengakses halaman dalam kondisi belum melakukan <i>login</i>, maka sistem akan menampilkan halaman <i>login</i> beserta peringatan "Anda belum melakukan proses login"
<i>Postconditions</i>	Data hasil asesmen siswa tertentu dapat ditampilkan berdasarkan kata kunci yang diketikkan asesor

H. Use Case Scenario Login

Berikut pada tabel 4.9 adalah penjelasan dari *use case scenario login*.

Tabel 4.9 Use Case Scenario Login

Item	Deskripsi
<i>Use case ID</i>	UC-LSPG-08
<i>Use case Name</i>	<i>Login</i>
<i>Functional ID</i>	F-LSPG-08
<i>Actor</i>	Asesor dan Siswa
<i>Description</i>	Asesor dan Siswa dapat masuk ke dalam sistem
<i>Use Goal</i>	Sistem dapat menampilkan fitur-fitur yang ada setelah aktor melakukan proses ini
<i>Preconditions</i>	Pengguna sudah terdaftar sebagai siswa ataupun asesor serta memiliki koneksi internet
<i>Main flow</i>	1. Pengguna membuka halaman web sistem informasi asesmen LSP 2. Sistem menampilkan <i>form</i> untuk pengisian <i>username</i> dan <i>password</i> 3. Pengguna mengisi <i>username</i> dan <i>password</i> pada <i>form</i> yang diberikan 4. Pengguna menekan tombol <i>login</i> 5. Sistem melakukan pengecekan apakah <i>username</i> dan <i>password</i> yang diisikan telah sesuai dengan data yang ada pada <i>database</i> 6. Apabila <i>username</i> dan <i>password</i> sesuai, maka sistem akan masuk ke halaman beranda Asesor maupun Siswa
<i>Alternative flow</i>	1. Apabila <i>username</i> dan <i>password</i> tidak sesuai, maka sistem akan menampilkan peringatan bahwa <i>username</i> atau <i>password</i> salah 2. Jika terdapat data yang kosong, sistem akan menampilkan peringatan pada <i>field</i> yang kosong agar pengguna dapat mengisi <i>field</i> tersebut
<i>Postconditions</i>	Pengguna telah melakukan proses <i>login</i> dan dapat mengakses fitur yang ada dalam sistem

I. Use Case Scenario Logout

Berikut pada tabel 4.10 adalah penjelasan dari *use case scenario logout*.

Tabel 4.10 Use Case Scenario Logout

Item	Deskripsi
<i>Use case ID</i>	UC-LSPG-09
<i>Use case Name</i>	<i>Logout</i>
<i>Functional ID</i>	F-LSPG-09
<i>Actor</i>	Asesor dan Siswa
<i>Description</i>	Asesor dan Siswa dapat keluar dari sistem
<i>Use Goal</i>	Sistem dapat mengeluarkan pengguna agar tidak digunakan pengguna lain yang tidak berkepentingan
<i>Preconditions</i>	Pengguna telah melakukan proses <i>login</i>
<i>Main flow</i>	1. Pengguna memilih menu <i>logout</i> 2. Sistem menampilkan peringatan apakah yakin akan melakukan proses <i>logout</i> 3. Apabila pengguna memilih pilihan ya, sistem melakukan proses <i>logout</i> dengan menghapus <i>session</i> dan <i>cookies</i> yang ada 4. Sistem menampilkan halaman <i>login</i>
<i>Alternative flow</i>	1. Apabila pengguna memilih pilihan tidak, sistem tidak akan melakukan proses <i>logout</i> dan kembali ke menu sebelumnya
<i>Postconditions</i>	Pengguna dapat keluar dari dalam sistem dan membatalkan proses <i>login</i>

J. Use Case Scenario Manajemen Asesmen Mandiri

Berikut pada tabel 4.11 adalah penjelasan dari *use case scenario* manajemen asesmen mandiri.

Tabel 4.11 Use Case Scenario Mencari Hasil Asesmen Mandiri

Item	Deskripsi
<i>Use case ID</i>	UC-LSPG-10
<i>Use case Name</i>	Manajemen Asesmen Mandiri
<i>Functional ID</i>	F-LSPG-10
<i>Actor</i>	Asesor

Tabel 4.11 *Use Case Scenario* Manajemen Asesmen Mandiri (Lanjutan)

Item	Deskripsi
<i>Description</i>	Asesor dapat melakukan manajemen pada asesmen mandiri yang meliputi skema, unit, elemen, dan pertanyaan asesmen mandiri
<i>Use Goal</i>	Asesmen mandiri dapat dikelola berdasarkan skema
<i>Preconditions</i>	Asesor telah melakukan proses <i>login</i>
<i>Main Flow</i>	1. Sistem mengambil data skema, unit, elemen, dan pertanyaan asesmen mandiri dari <i>database</i> 2. Sistem menampilkan menu manajemen skema, unit, elemen, dan pertanyaan 3. Asesor memilih untuk melakukan manajemen pada salah satu menu yakni skema, unit, elemen, dan pertanyaan 4. Sistem mengambil data berdasarkan menu yang dipilih oleh asesor 5. Asesor dapat melakukan tambah, ubah, dan hapus data pada menu yang telah dipilih
<i>Alternative flow</i>	3. Apabila asesor login dengan <i>role</i> asesor maka menu yang tersedia hanya menu pertanyaan. Sedangkan apabila asesor login dengan <i>role</i> manajemen maka seluruh menu yang tersedia dapat digunakan 4. Jika asesor mengakses halaman dalam kondisi belum melakukan <i>login</i>, maka sistem akan menampilkan halaman <i>login</i> beserta peringatan "Anda belum melakukan proses login"
<i>Postconditions</i>	Sistem informasi asesmen mandiri dapat terkelola berdasarkan skema yang tersedia

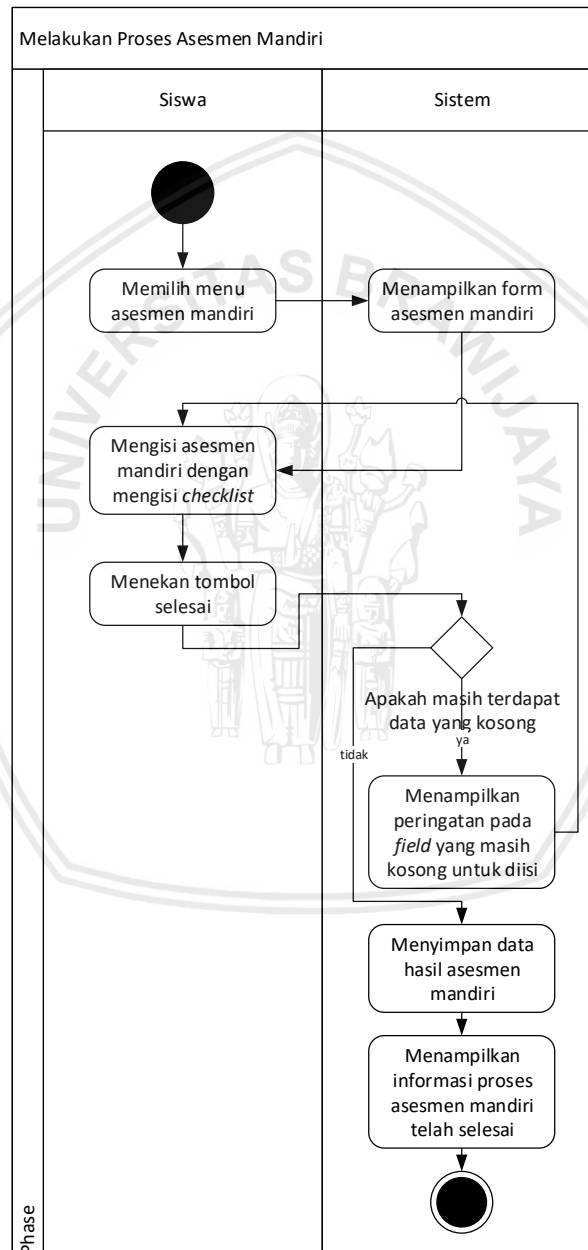
4.1.5 Activity Diagram

Activity diagram merupakan penggambaran aktivitas yang dilakukan oleh sistem. Selain itu, didalamnya juga menjelaskan detail siapa saja yang terlibat dalam suatu aktivitas dalam sistem. Berikut merupakan *activity diagram* yang terdapat dalam sistem.

4.1.5.1 Activity Diagram Melakukan Proses Asesmen Mandiri

Berikut pada gambar 4.3 adalah *activity diagram* melakukan proses asesmen mandiri yang dilakukan oleh siswa. Dari gambar 4.3 dapat dilihat bahwa terdapat

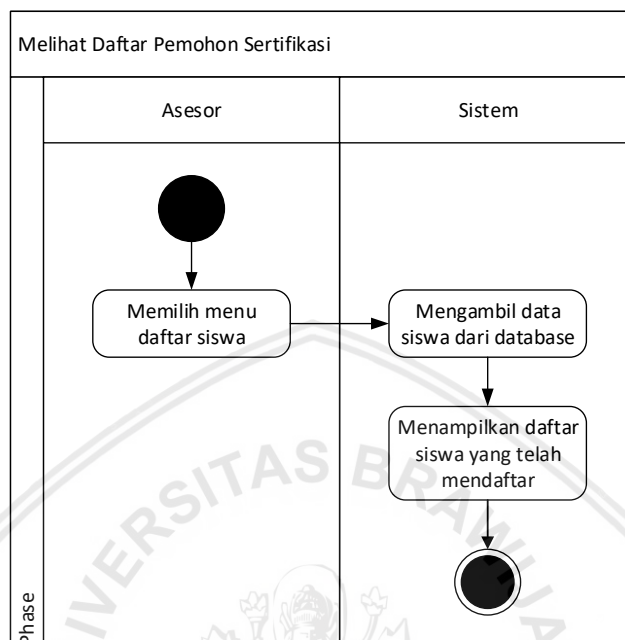
proses untuk mengisi asesmen mandiri yang dilakukan oleh siswa. Proses diawali oleh siswa memilih menu asesmen mandiri, setelah itu sistem akan menampilkan lembar asesmen mandiri kepada siswa. Selanjutnya, siswa akan mengisi lembar asesmen mandiri tersebut dan menekan tombol selesai. Jika terdapat data yang kosong, maka sistem akan menampilkan peringatan terdapat data yang kosong. Sebaliknya, jika semua data telah diisi, maka sistem akan menyimpan hasil asesmen mandiri dan menampilkan informasi kepada siswa apabila proses asesmen mandiri telah selesai dilaksanakan.



Gambar 4.3 Activity Diagram Melakukan Proses Asesmen Mandiri

4.1.5.2 Activity Diagram Melihat Daftar Pemohon Sertifikasi

Berikut pada gambar 4.4 adalah *activity diagram* melihat daftar pemohon sertifikasi yang dilakukan oleh asesor.

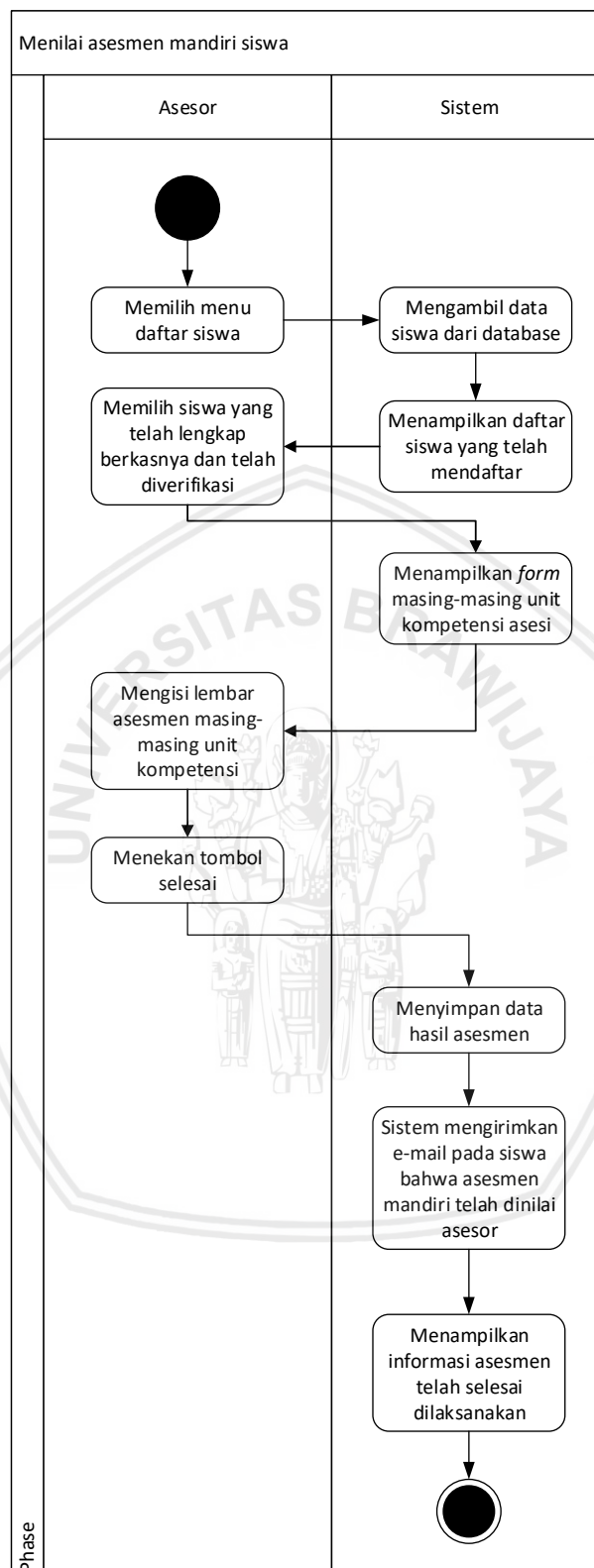


Gambar 4.4 Activity Diagram Melihat Daftar Pemohon Sertifikasi

Dari gambar 4.4 dapat dilihat bahwa terdapat proses untuk asesor melihat daftar pemohon aplikasi. Dari proses tersebut asesor memilih menu daftar siswa, kemudian sistem akan mengambil data siswa dari *database* untuk ditampilkan kepada asesor.

4.1.5.3 Activity Diagram Menilai Asesmen Mandiri Siswa

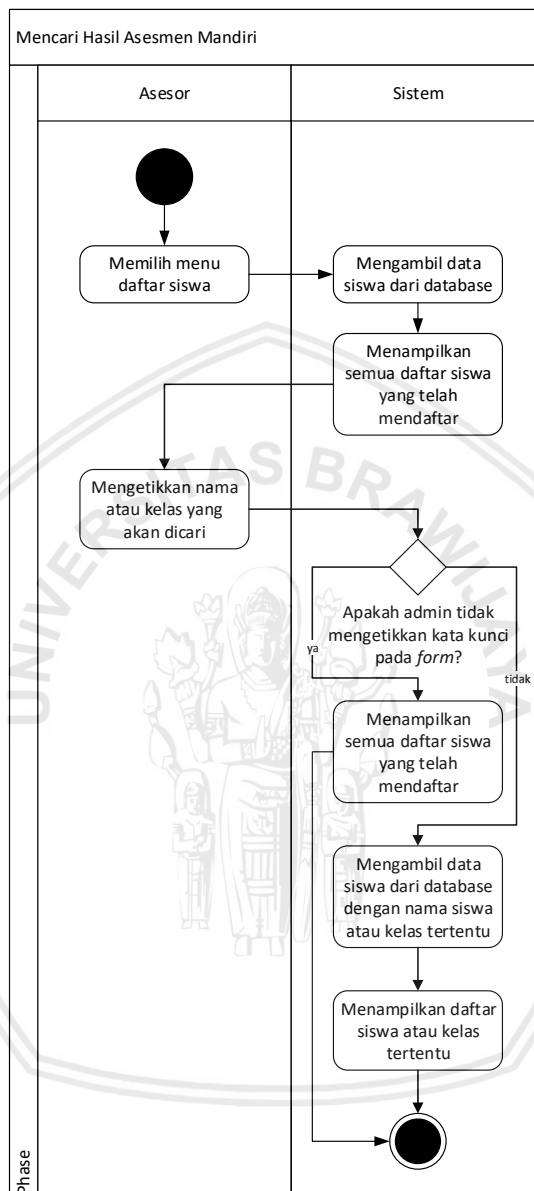
Berikut pada gambar 4.5 adalah *activity diagram* menilai asesmen mandiri siswa yang dilakukan oleh asesor. Dari gambar 4.5 dapat dilihat bahwa terdapat proses untuk asesor mengisi lembar asesmen kepada siswa. Langkah-langkah yang terjadi ialah asesor akan memilih salah satu siswa dari proses sebelumnya yakni melihat daftar pemohon sertifikasi dan memilih siswa yang telah lengkap persyaratannya. Setelah asesor memilih salah satu siswa untuk dilakukan proses asesmen, sistem akan menampilkan *form* untuk diisi oleh asesor sesuai dengan kecukupan persyaratan kompetensi yang diajukan kepada siswa. Setelah asesor mengisi semua *item* yakni V (valid), A (asli), T (terkini), dan M (memadai) asesor menekan tombol selesai.



Gambar 4.5 Activity Diagram Menilai Asesmen Mandiri Siswa

4.1.5.4 Activity Diagram Mencari Hasil Asesmen Mandiri

Berikut pada gambar 4.6 adalah *activity diagram* mencari hasil asesmen yang dilakukan oleh asesor.



Gambar 4.6 Activity Diagram Mencari Hasil Asesmen

Dari gambar 4.6 dapat dilihat bahwa terdapat proses untuk asesor mencari data siswa yang telah dilakukan proses asesmen. Pencarian tersebut dapat berdasarkan nama maupun kelas. Apabila asesor mengetikkan nama tertentu, maka sistem akan menampilkan hasil asesmen siswa sesuai apa yang diketikkan asesor. Namun apabila asesor kembali mengosongkan *form* isian pencarian tersebut, maka sistem akan menampilkan kembali seluruh data siswa yang telah mendaftar sebagai peserta sertifikasi profesi.

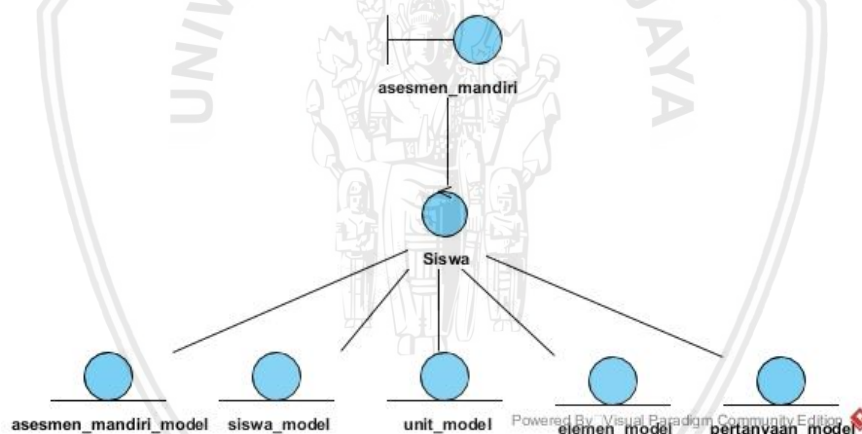
4.2 Perancangan Sistem

Perancangan sistem merupakan salah satu proses dalam tahapan *modeling* dari SDLC *waterfall*. Pada proses ini digunakan untuk merancang aplikasi yang akan diimplementasikan pada kode program. Proses perancangan menggunakan *unified modeling language* (UML) diantaranya *collaboration diagram*, *sequence diagram*, dan *class diagram*.

4.2.1 Collaboration Diagram

Dalam proses analisis kelas, digunakan *collaboration diagram* untuk mengetahui relasi atau komunikasi yang dilakukan kelas-kelas dalam suatu proses. Selain untuk mengetahui relasi kelas dalam suatu proses, *collaboration diagram* digunakan untuk mengetahui kelas-kelas apa saja yang akan digunakan di dalam sistem. Dalam penggambarannya, terdapat beberapa simbol yakni *model*, *controller*, dan *view*. Pada proses ini digunakan 4 (empat) fungsional utama program untuk digambarkan pada *collaboration diagram* yakni melakukan proses asesmen mandiri, melihat daftar pemohon sertifikasi, menilai asesmen mandiri siswa dan mencari hasil asesmen mandiri.

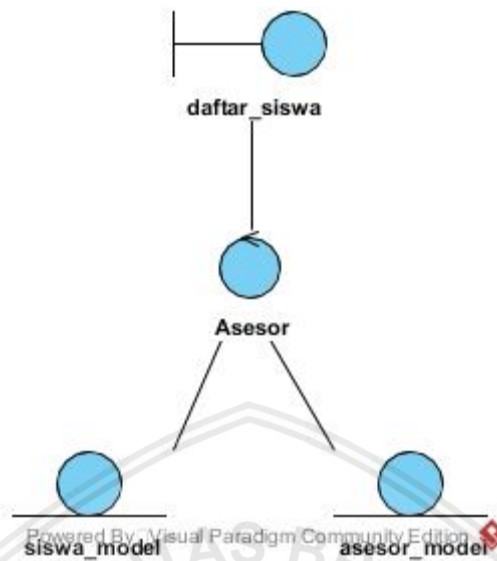
4.2.1.1 Collaboration Diagram Melakukan Proses Asesmen Mandiri



Gambar 4.7 Collaboration Diagram Melakukan Proses Asesmen Mandiri

Pada gambar 4.7 menunjukkan hasil analisis *collaboration diagram* asesmen mandiri yang dilakukan oleh siswa. Dari analisis tersebut mendapatkan hasil bahwa terdapat 1 (satu) *view*, 1 (satu) *controller*, dan 5 (lima) *model*. Terdapat satu *view* yakni *asesmen_mandiri* yang berfungsi sebagai halaman pengisian asesmen mandiri. Selanjutnya, pada bagian *controller* terdapat *controller* siswa yang bertugas mengatur proses asesmen mandiri. *Controller* siswa berhubungan dengan *model* unit, elemen, pertanyaan, siswa dan *asesmen_mandiri*. *Controller* siswa meminta masing-masing data dari unit, elemen, dan pertanyaan serta data siswa untuk ditampilkan dalam *view* *asesmen_mandiri*. Setelah itu, siswa dapat mengisi asesmen mandiri pada *view* *asesmen_mandiri*. Hasil asesmen mandiri dikirim *view* *asesmen_mandiri* menuju *controller* siswa kembali untuk disimpan dalam database melalui *model* *asesmen_mandiri_model*.

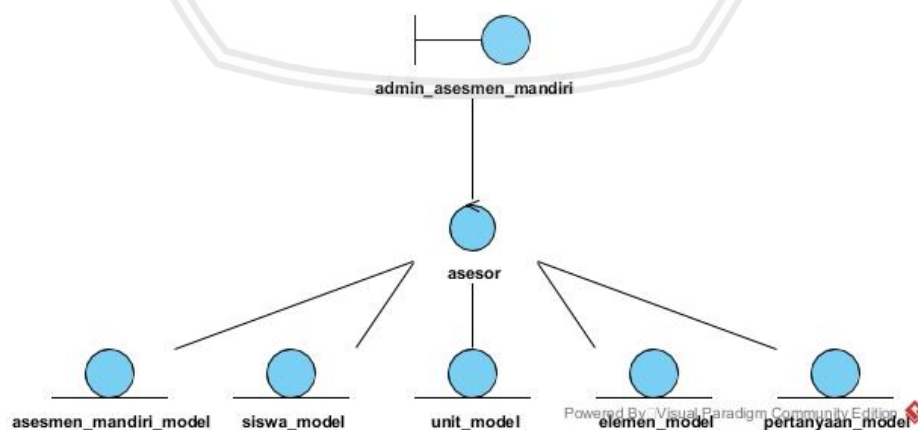
4.2.1.2 Collaboration Diagram Melihat Daftar Pemohon Sertifikasi



Gambar 4.8 Collaboration Diagram Melihat Daftar Pemohon Sertifikasi

Pada gambar 4.8 menunjukkan hasil analisis *collaboration diagram* melihat daftar pemohon sertifikasi yang dilakukan oleh asesor. Dari analisis tersebut mendapatkan hasil bahwa terdapat 1 (satu) *view*, 1 (satu) *controller*, serta 2 (dua) *model*. *View* daftar siswa bertugas menampilkan daftar siswa pemohon sertifikasi. *View* tersebut dikendalikan oleh *controller* asesor. *Controller* asesor tersebut bertugas mengambil data siswa dari *model* siswa dan data asesor dari *database*. *Model* siswa mengambil data dari *database* melalui dan dikendalikan oleh *controller* asesor. Begitu pula dengan *controller* asesor yang mengambil data dari *database* melalui *model* asesor_model.

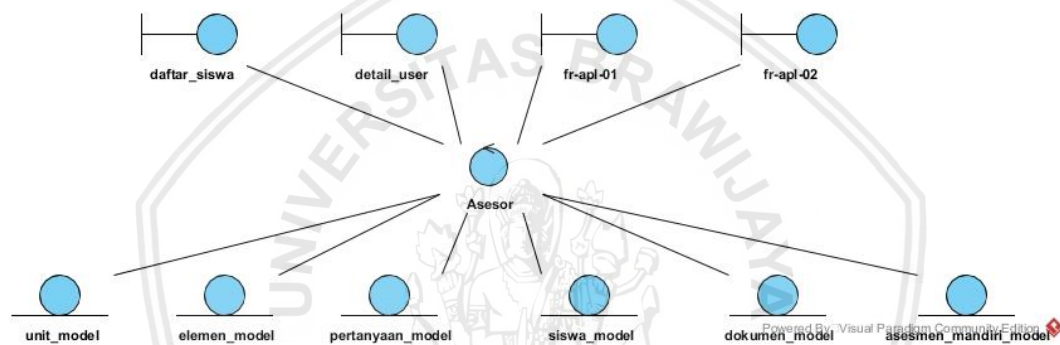
4.2.1.3 Collaboration Diagram Menilai Asesmen Mandiri Siswa



Gambar 4.9 Collaboration Diagram Menilai Asesmen Mandiri Siswa

Pada gambar 4.9 menunjukkan hasil analisis *collaboration diagram* menilai asesmen mandiri yang dilakukan oleh asesor. Dari analisis tersebut mendapatkan hasil bahwa terdapat 1 (satu) *view*, 1 (satu) *controller*, serta 6 (enam) *model*. *View* asesor_asesmen_mandiri berfungsi menampilkan hasil asesmen mandiri siswa yang akan dilakukan penilaian serta menampilkan *form* penilaian untuk asesor. *View* asesor_asesmen_mandiri dikendalikan oleh *controller* asesor yang bertugas mengambil data dari berbagai model yakni asesmen_mandiri_model, siswa_model, elemen_model, pertanyaan_model, dan unit_model. Setelah data untuk penilaian asesmen mandiri terkumpul, maka *controller* asesor akan menampilkan *view* asesor_asesmen_mandiri. Setelah asesor menyelesaikan penilaian asesmen mandiri, maka hasil penilaian tersebut akan disimpan oleh *model* asesmen_mandiri_model.

4.2.1.4 Collaboration Diagram Mencari Hasil Asesmen Mandiri



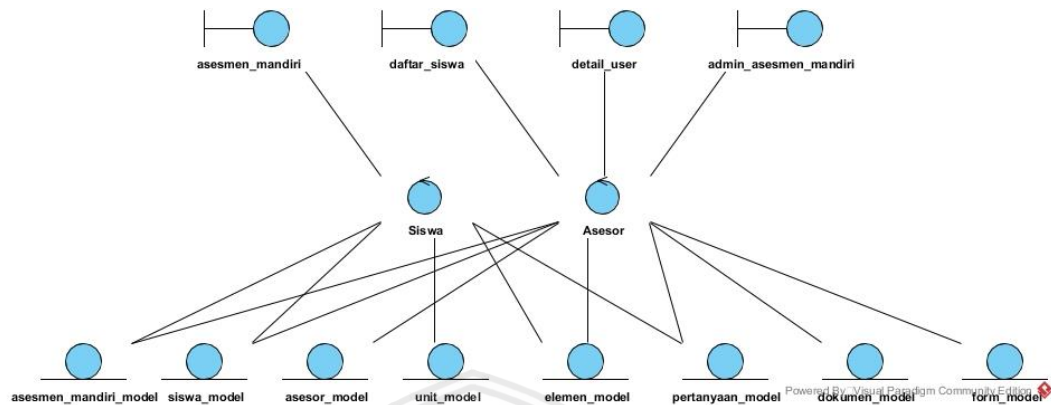
Gambar 4.10 Collaboration Diagram Mencari Hasil Asesmen Mandiri

Pada gambar 4.10 menunjukkan hasil analisis *collaboration diagram* mencari hasil asesmen yang dilakukan oleh asesor. Dari analisis tersebut mendapatkan hasil bahwa terdapat 2 (dua) *view*, 1 (satu) *controller*, serta 4 (empat) *model*. *View* pertama yakni daftar_siswa berfungsi menampilkan daftar siswa yang telah diambil dari *database* melalui *controller* asesor. Setelah siswa yang dilakukan pencarian telah tampil, maka aplikasi akan menampilkan *view* detail_user yang mengambil data dari *model* siswa, dokumen, dan *form* serta dikendalikan oleh *controller* asesor. Dalam *view* tersebut menampilkan detail keseluruhan siswa mulai dari biodata, dokumen pendukung, dan *form* biodata serta *form* hasil asesmen mandiri.

4.2.1.5 Unifikasi Kelas Analisis

Dalam tahap kedua proses analisis kelas yakni unifikasi kelas analisis, seluruh *collaboration diagram* yang telah digambarkan pada poin sebelumnya digabungkan menjadi kesatuan untuk melihat alur keseluruhan program. Selain untuk melihat gambaran utuh program, pada proses ini akan menunjukkan rancangan *sequence diagram* yang akan dibuat. Penggabungan *collaboration diagram* dapat dilihat pada gambar 4.11. Dari gambar tersebut terdapat 2 (dua) *controller* utama yakni siswa dan asesor. Kedua *controller* tersebut berfungsi

untuk menjalankan atau mengendalikan keseluruhan besar fungsional sistem dan berhubungan dengan *model* untuk mengambil data dari *database*.



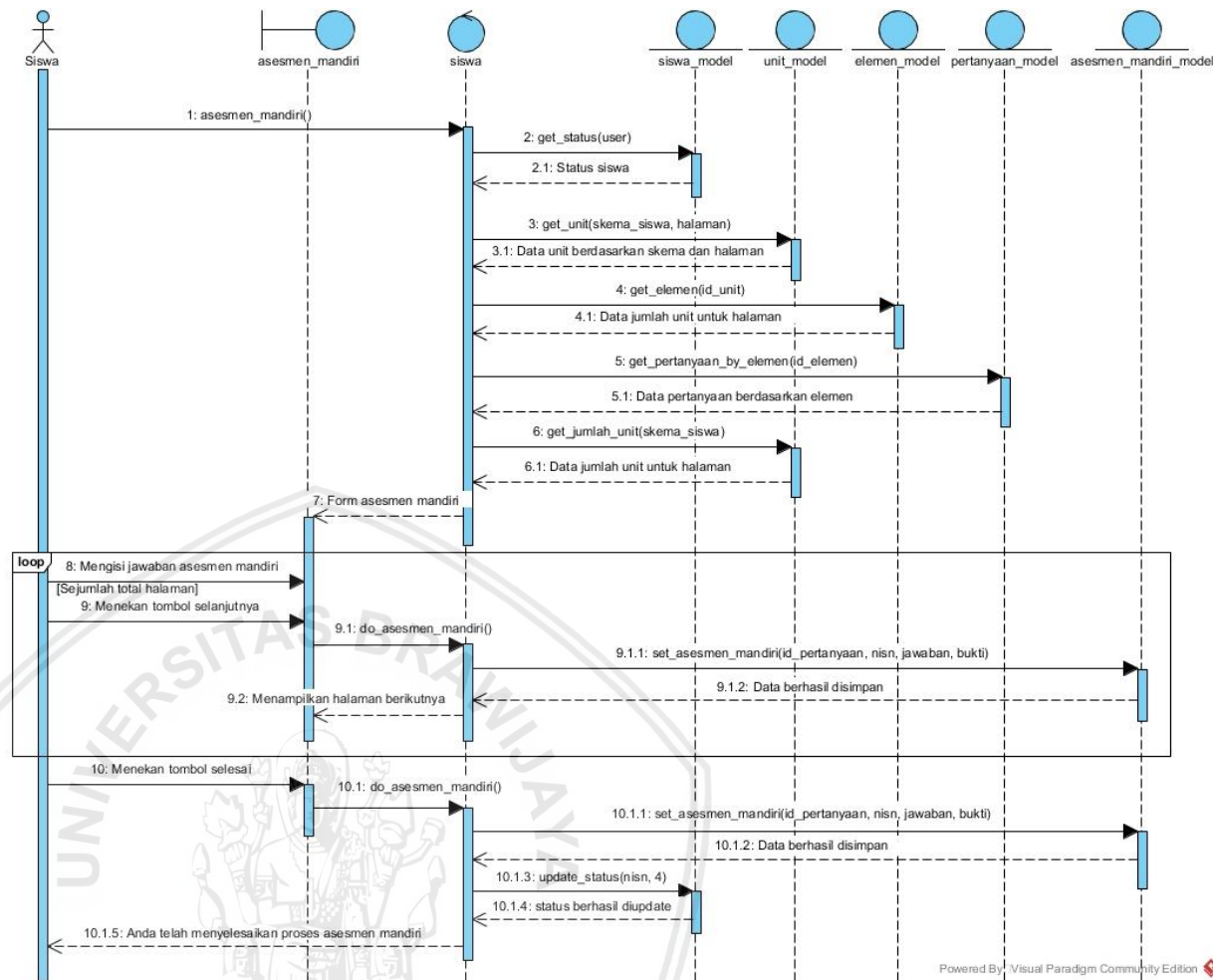
Gambar 4.11 Diagram Unifikasi Kelas Analisis

4.2.2 Sequence Diagram

Pada *Sequence diagram* adalah proses penggambaran alur program yang akan diimplementasikan pada kode program. Pada analisis perancangan program menggunakan sequence diagram, digunakan 4 (empat) fungsional utama yakni asesmen mandiri, melihat daftar pemohon sertifikasi, melakukan proses asesmen dan mencari hasil asesmen.

4.2.2.1 Sequence Diagram Melakukan Proses Asesmen Mandiri

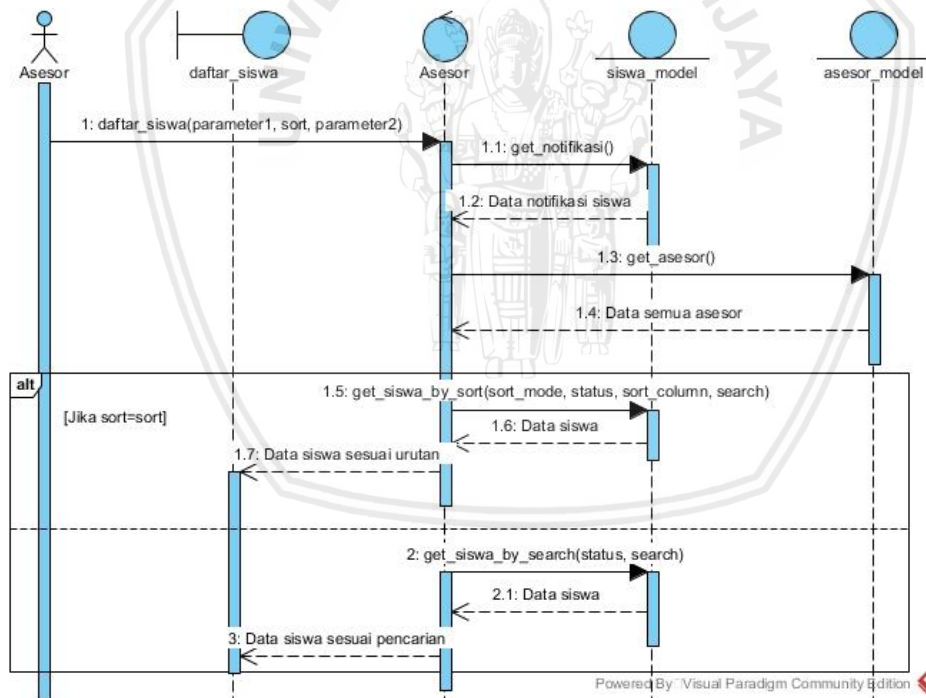
Pada gambar 4.12 adalah *sequence diagram* yang menggambarkan proses asesmen mandiri yang dilakukan oleh siswa. Dalam proses tersebut, siswa terlebih dahulu memilih menu asesmen mandiri pada navbar yang tersedia. Setelah itu, sistem akan menampilkan *form* asesmen mandiri yang harus diisi oleh siswa. *Form* tersebut akan mengambil data status, unit, elemen, dan pertanyaan yang akan diberikan kepada siswa. Setelah siswa selesai mengisi asesmen mandiri tersebut, siswa menekan tombol next yang akan mengirim data tersebut pada *database* untuk disimpan untuk 1 (unit yang telah diselesaikan). Apabila siswa tersebut tidak mengisi *form* tersebut secara lengkap, maka sistem akan memberikan peringatan pada *field* yang kosong bahwa masih ada data yang belum terisi. Setelah selesai, siswa menekan tombol selesai untuk menyudahi proses asesmen mandiri. Sistem akan merubah status siswa tersebut menjadi 3 (tiga) atau telah menyelesaikan asesmen mandiri. Sistem akan kembali menampilkan menu utama pada sistem informasi disertai peringatan bahwa siswa telah menyelesaikan asesmen mandiri.



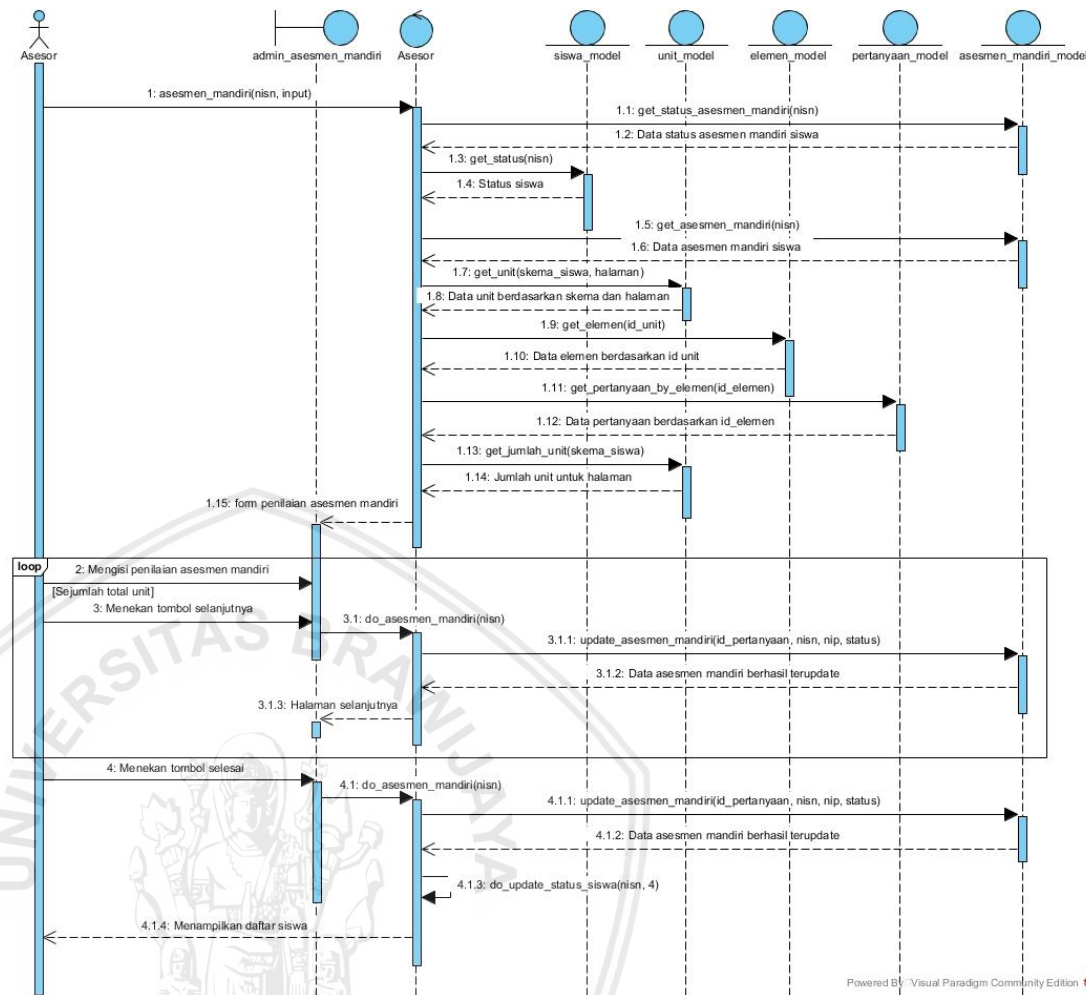
Gambar 4.12 Sequence Diagram Melakukan Proses Asesmen Mandiri

4.2.2.2 Sequence Diagram Melihat Daftar Pemohon Sertifikasi

Pada gambar 4.13 adalah *sequence diagram* yang menggambarkan proses sistem menyediakan daftar pemohon sertifikasi untuk asesor. Pada proses tersebut dimulai dengan asesor memilih menu daftar siswa pada navbar yang telah disediakan. Kemudian sistem akan memanggil fungsi `daftar_siswa` pada *controller* asesor. Pada *controller* tersebut akan mengumpulkan data notifikasi, data siswa sesuai urutan maupun sesuai pencarian dari *controller* siswa, serta daftar asesor yang ada. Apabila seluruh data telah terkumpul, maka sistem akan menampilkan daftar siswa pada asesor. Pada proses tersebut terdapat dua alternative yakni apabila asesor tidak mengetikkan kata kunci dalam *field* pencarian, maka akan tampil seluruh data siswa. Daftar siswa tersebut dapat diurutkan lagi menjadi *ascending* maupun *descending* sesuai *field* yang akan diurutkan oleh asesor. Apabila asesor mengetikkan kata kunci pada *field* pencarian yang tersedia dan menekan tombol cari, maka sistem akan menampilkan hanya data yang berhubungan dengan kata kunci yang dicari oleh asesor. Setelah data tersebut ditampilkan, dapat pula diurutkan berdasarkan *field* yang diinginkan oleh asesor.



Gambar 4.13 Sequence Diagram Melihat Daftar Pemohon Sertifikasi



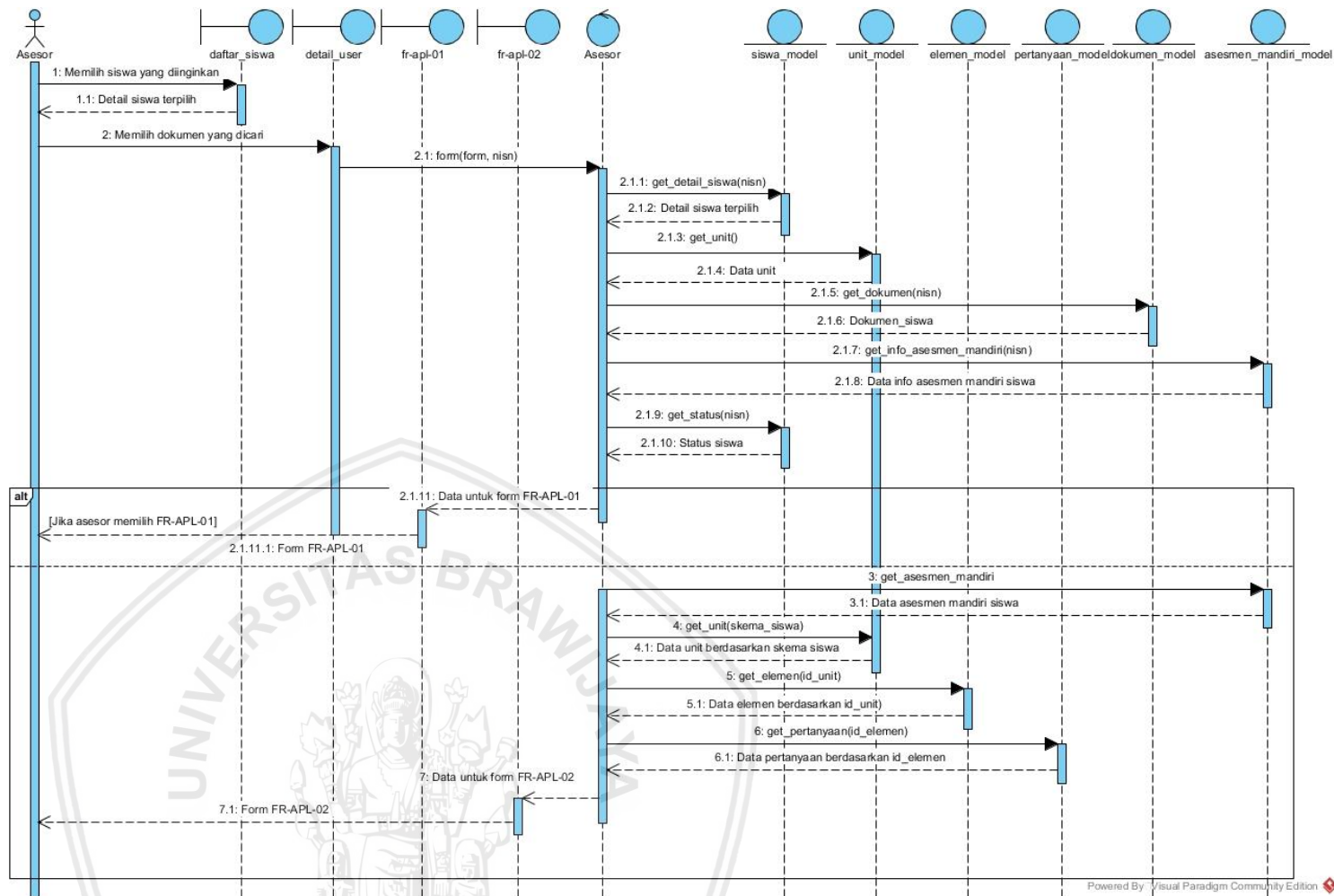
Gambar 4.14 Sequence Diagram Menilai Asesmen Mandiri Siswa

4.2.2.3 Sequence Diagram Menilai Asesmen Mandiri Siswa

Pada gambar 4.14 adalah *sequence diagram* yang menggambarkan proses sistem menyediakan daftar pemohon sertifikasi untuk asesor. Pada proses tersebut dimulai dengan asesor memilih menu daftar siswa pada navbar yang telah disediakan (proses menampilkan daftar siswa telah dijelaskan pada gambar 4.13). Asesor akan memilih siswa yang akan dilakukan asesmen dan sistem akan menampilkan detail mengenai siswa tersebut. Untuk menilai asesmen mandiri, asesor perlu untuk menekan tombol mulai asesmen. Setelah itu, sistem akan menampilkan hasil asesmen mandiri siswa yang telah dipilih untuk dilakukan penilaian oleh asesor. Asesor akan mulai untuk menilai hasil asesmen mandiri yang telah dilakukan oleh siswa. Setelah semua unit telah dinilai oleh asesor, maka sistem akan merubah status siswa tersebut menjadi 4 (empat) atau telah dilakukan asesmen mandiri dan dapat dilanjutkan untuk mengerjakan uji kompetensi langsung yang disediakan.

4.2.2.4 Sequence Diagram Mencari Hasil Asesmen Mandiri

Pada gambar 4.15 adalah *sequence diagram* yang menggambarkan proses mencari hasil asesmen mandiri siswa. Pencarian hasil asesmen dimulai dari mencari siswa yang akan dilihat dokumen maupun hasil asesmennya. Asesor akan memilih siswa dari daftar siswa yang telah disediakan oleh sistem. Setelah asesor memilih siswa, maka sistem akan menampilkan detail dari siswa tersebut. Pada menu detail siswa tersebut terdapat pilihan untuk melihat dokumen pendukung sertifikasi siswa serta dokumen hasil asesmen mandiri. Dokumen hasil asesmen mandiri antara lain FR-APL-01 yakni berisi biodata siswa serta informasi-informasi mengenai data diri siswa, serta FR-APL-02 yang berisi hasil asesmen mandiri siswa beserta hasil penilaian asesor. Pada kedua *form* hasil asesmen mandiri tersebut terdapat fungsi untuk langsung mencetak pada browser yang terbuka baik dapat dicetak langsung pada *printer* maupun diunduh menjadi file berekstensi .pdf.



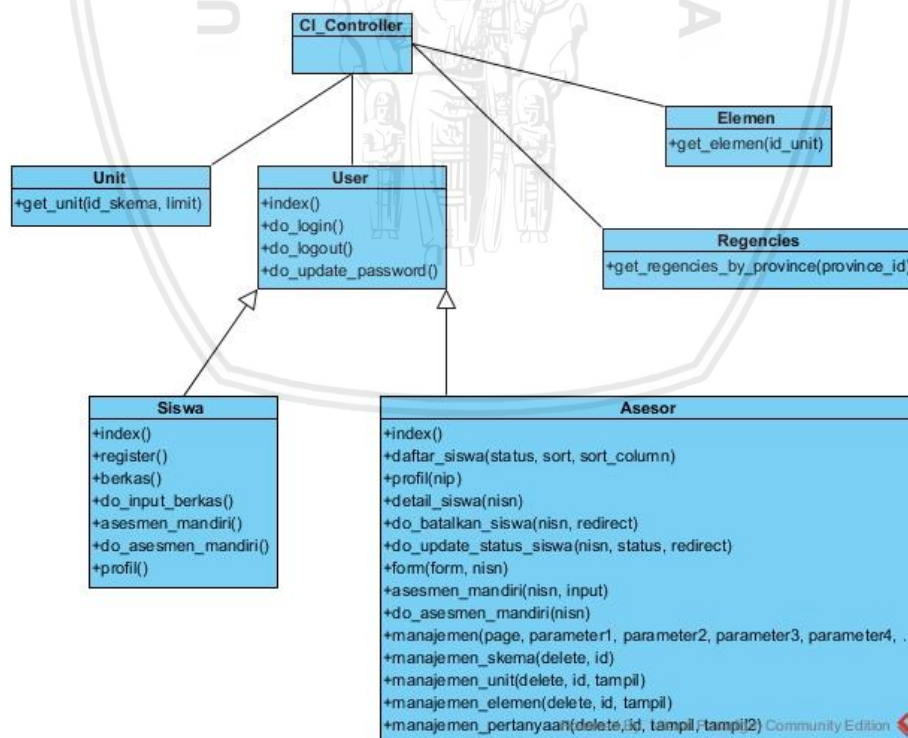
Gambar 4.15 Sequence Diagram Mencari Hasil Asesmen Mandiri

4.2.3 Class Diagram

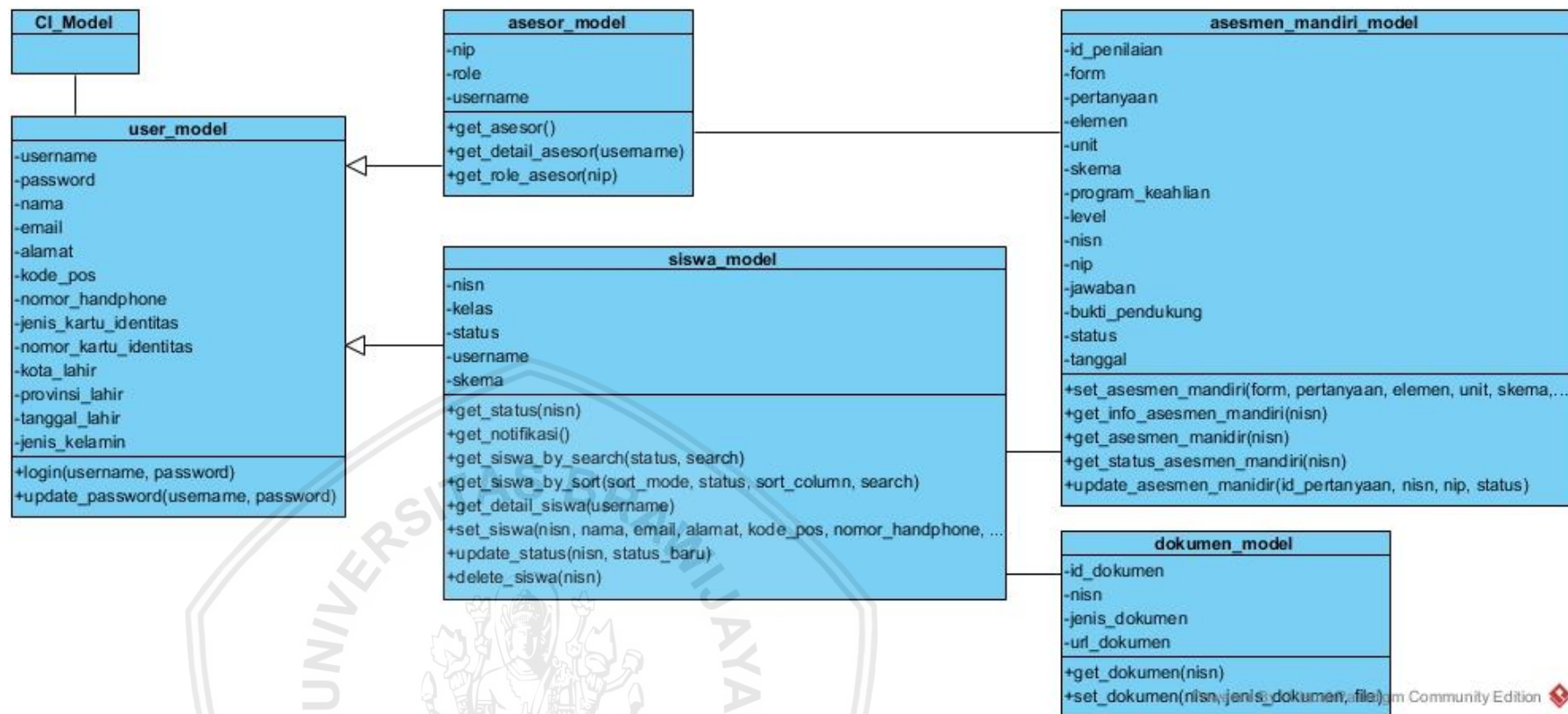
Class diagram (kelas diagram) merupakan penggambaran keseluruhan kelas-kelas yang ada pada program yang akan dibuat. Kelas diagram dibuat mengacu pada unifikasi kelas analisis yang terdapat pada gambar 4.11. Kelas diagram juga menjelaskan relasi antar kelas yang akan dibuat. Relasi antar kelas tersebut didapatkan dari masing-masing proses yang berjalan pada *sequence* diagram. Untuk hasil analisis *controller* kelas diagram sistem informasi asesmen Lembaga sertifikasi profesi dari 4 (empat) fungsional yang digunakan dapat dilihat pada gambar 4.16.

4.2.3.1 Class Diagram Controller

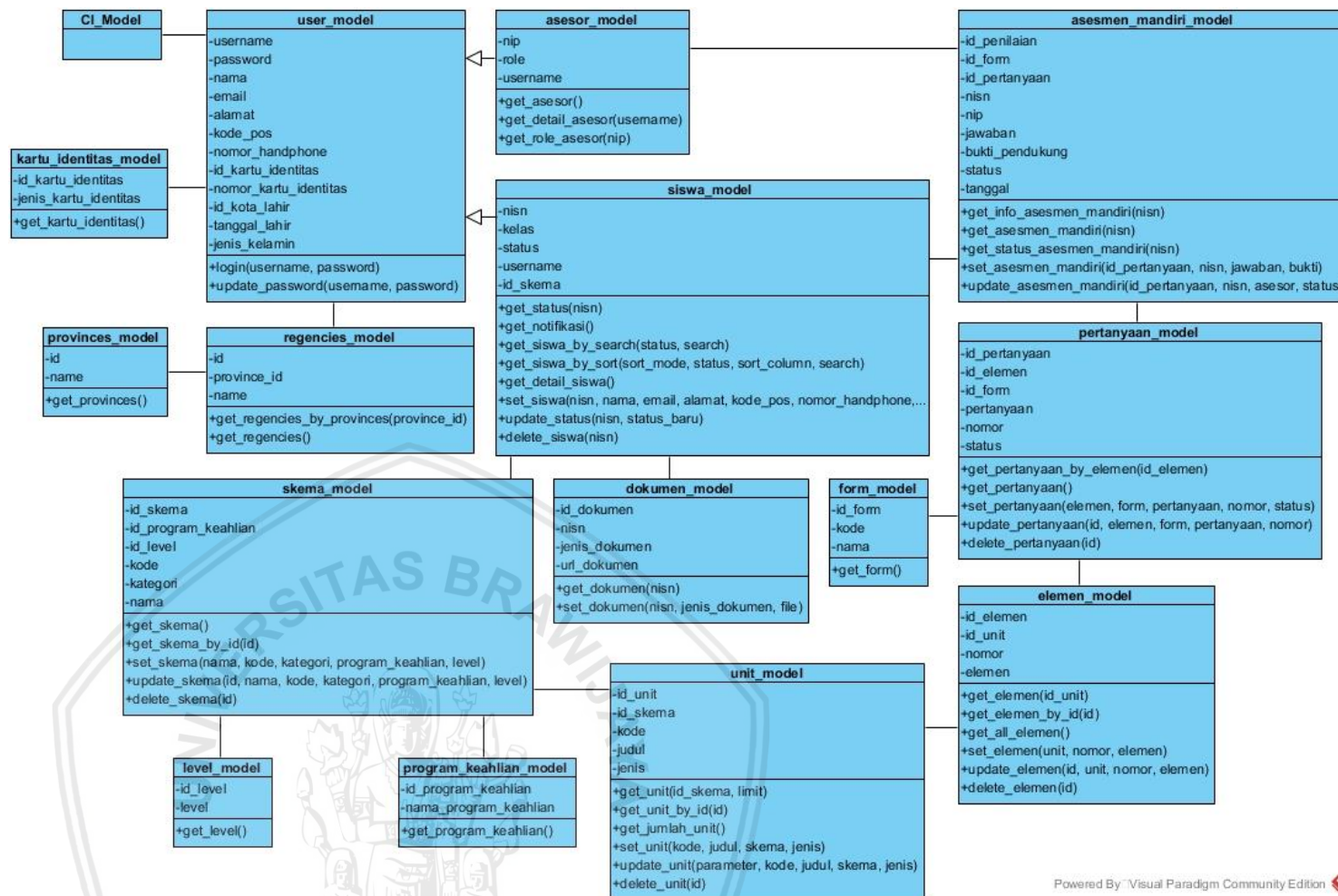
Pada gambar 4.16 adalah *class diagram controller* menunjukkan analisis controller yang akan dibuat pada implementasi aplikasi. Terdapat 2 (dua) *controller* utama dalam menjalankan aplikasi yakni *controller* siswa dan asesor yang merupakan generalisasi atau turunan dari *controller* user. Selain itu, terdapat 3 (tiga) *controller* lain sebagai pendukung aplikasi. *Controller* tersebut berfungsi sebagai penyedia data untuk *view* melalui mekanisme *json_encode*. Setelah *class diagram controller* didapatkan, maka dilanjutkan dalam perancangan *class diagram model*.



Gambar 4.16 Class Diagram Controller



Gambar 4.17 Class Diagram Model Awal



Powered By Visual Paradigm Community Edition

Gambar 4.18 Class Diagram Model Setelah Normalisasi

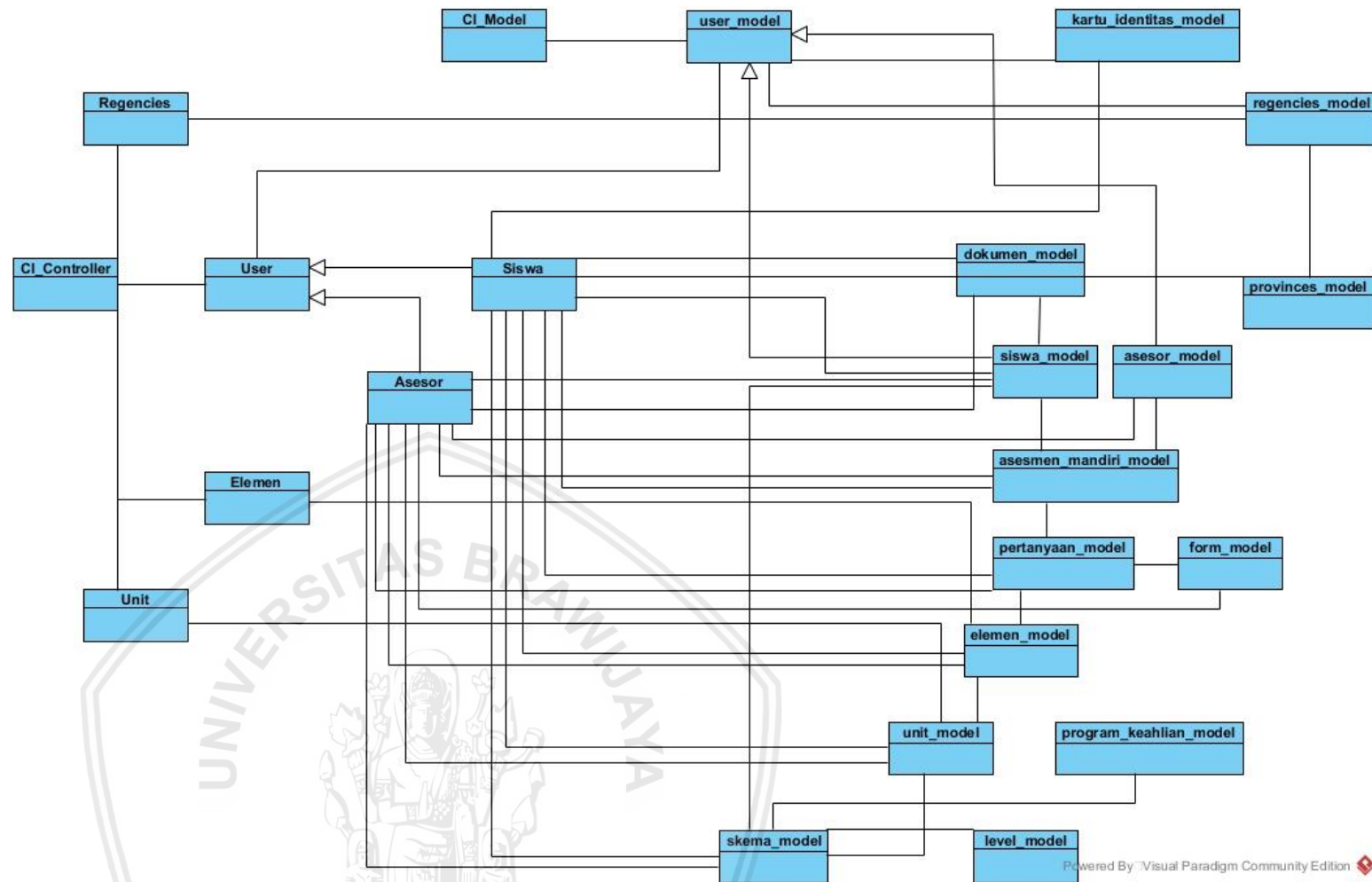
4.2.3.2 Class Diagram Model

Pada gambar 4.17 adalah *class diagram model* yang menunjukkan *class* yang akan berhubungan dengan *database*. Dari analisis awal *class diagram model*, terdapat 5 (lima) *class* yang diperlukan agar aplikasi dapat berhubungan dengan *database*. Dari kelima *class* tersebut terdapat atribut yang akan digunakan untuk perwujudan kolom dalam *database*. Selain atribut, terdapat *method* atau fungsi pada masing-masing *class* yakni fungsi yang digunakan aplikasi untuk berhubungan dengan data dalam *database*.

Dari gambar 4.17 tersebut kemudian dianalisis kembali dan dilakukan pemecahan pada berbagai komponen *class* guna untuk mempermudah manajemen dan pengolahan data. Pengolahan data yang dimaksud dapat dilihat pada *class* asesmen_mandiri pada gambar 4.17. Apabila data hasil asesmen mandiri telah masuk ke dalam tabel asesmen_mandiri, dan terjadi perubahan nama skema, maka semua data pada tabel tersebut harus dilakukan perubahan. Untuk hasil analisis dan penyempurnaan *class diagram model* dapat dilihat pada gambar 4.18. Dalam *class diagram* tersebut, apabila terdapat perubahan pada nama skema, maka pada tabel asesmen_mandiri tidak diperlukan adanya perubahan, namun hanya perlu merubah pada tabel skema yang telah terhubung dengan tabel asesmen_mandiri. Setelah *class diagram model* telah didapatkan, maka dilanjutkan dengan proses unifikasi *class diagram* untuk melihat keseluruhan *class diagram* baik *class diagram controller* dan *class diagram model* dan hubungan antar *class* tersebut.

4.2.3.3 Unifikasi Class Diagram

Pada gambar 4.19 adalah hasil unifikasi atau penyatuan *class diagram controller* dan *class diagram model*. Unifikasi *class diagram* diperlukan untuk melihat hubungan keseluruhan antara seluruh *class controller* dan *class model*. Dari unifikasi *class diagram* tersebut dapat dilihat bahwa terdapat dua *class controller* utama, yakni *controller* siswa dan asesor. Kedua *controller* tersebut merupakan *controller* yang mengendalikan mayoritas fungsional aplikasi. Setelah didapatkan hubungan antar *class* baik dari *class controller* maupun *class model*, dilanjutkan dengan membuat rancangan *database* berdasarkan *class diagram model* dan hubungan antar kelasnya.



Gambar 4.19 Unifikasi *Class Diagram*

4.2.4 Physical Data Model

Pada proses perancangan basis data, digunakan diagram *physical data model* (PDM) untuk menggambarkan struktur basis data yang dibutuhkan beserta relasinya. Rancangan tersebut didapatkan dari analisis *class diagram model* yang telah dibuat sebelumnya. Terdapat daftar kegunaan tabel yang mengacu pada jumlah *class diagram model* serta penjabaran kegunaan masing-masing tabel. Namun, setelah kembali dilakukan analisis, terdapat satu tambahan tabel yang berasal dari normalisasi tabel siswa yakni tabel status dengan kegunaan sebagai penyimpanan status dari siswa dalam proses asesmen yang dijalankan. Hasil seluruh perancangan kegunaan tabel dapat ditunjukkan oleh tabel 4.12

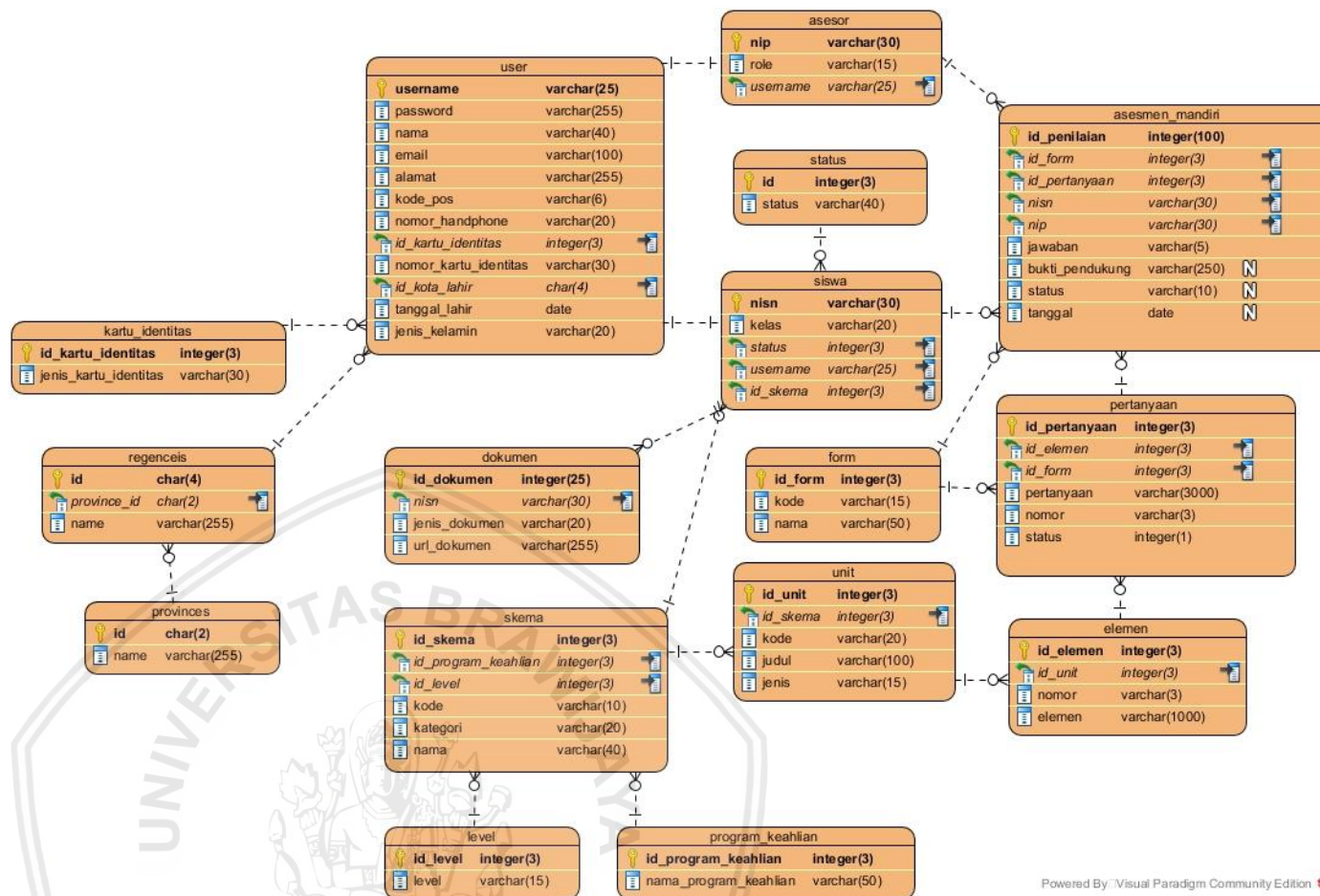
Tabel 4.12 Daftar Kegunaan Tabel

No	Nama Tabel	Kegunaan Tabel
1.	Asesmen_mandiri	Untuk menyimpan data hasil proses asesmen mandiri yang dilakukan oleh <i>user</i> dan asesor
2.	Asesor	Untuk menyimpan data asesor
3.	Biodata	Untuk menyimpan data siswa
4.	Dokumen	Untuk menyimpan data lokasi dan deskripsi dokumen pendukung asesmen yang diunggah oleh siswa.
5.	Elemen	Untuk menyimpan data masing-masing elemen pada <i>form</i> asesmen.
6.	<i>Form</i>	Untuk menyimpan data jenis-jenis <i>form</i> yang ada pada proses asesmen
7.	Kartu_identitas	Untuk menyimpan data jenis-jenis kartu identitas yang dapat digunakan untuk mendaftar sertifikasi
8.	Level	Untuk menyimpan data level asesmen yang sedang dilakukan
9.	Pertanyaan	Untuk menyimpan data pertanyaan pada <i>form-form</i> proses asesmen
10.	Program_keahlian	Untuk menyimpan data jenis-jenis program keahlian yang dapat melakukan proses asesmen
11.	Provinces	Untuk menyimpan data nama provinsi di seluruh Indonesia
12.	Regencies	Untuk menyimpan data nama Kota dan Kabupaten di seluruh Indonesia

Tabel 4.12 Daftar Kegunaan Tabel (Lanjutan)

No	Nama Tabel	Kegunaan Tabel
14.	Status	Untuk menyimpan data status siswa dalam proses asesmen
13.	Skema	Untuk menyimpan data jenis skema yang telah tersedia pada proses asesmen
14.	Unit	Untuk menyimpan kode dan judul unit kompetensi yang akan diujikan pada proses asesmen
15.	User	Untuk menyimpan data <i>user</i> dan <i>password</i> masing-masing <i>user</i> baik siswa maupun asesor

Sedangkan untuk *physical data model* yang telah dibuat berdasarkan daftar kegunaan tabel, dapat dilihat pada gambar 4.20. Dalam gambar tersebut menjelaskan seluruh entitas yang dibuat dan akan diimplementasikan pada *database*. Selain itu, terdapat garis relasi yang menunjukkan hubungan antar tabel serta jenis hubungan tersebut antara lain *one-to-one*, *one-to-many*, dan *many-to-one*.

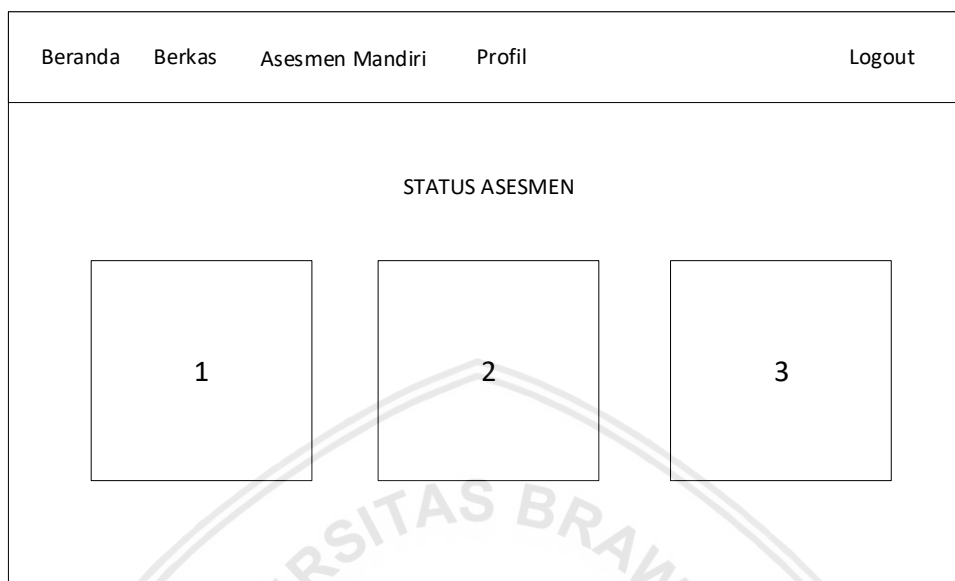


Powered By: Visual Paradigm Community Edition

Gambar 4.20 Physical Data Model

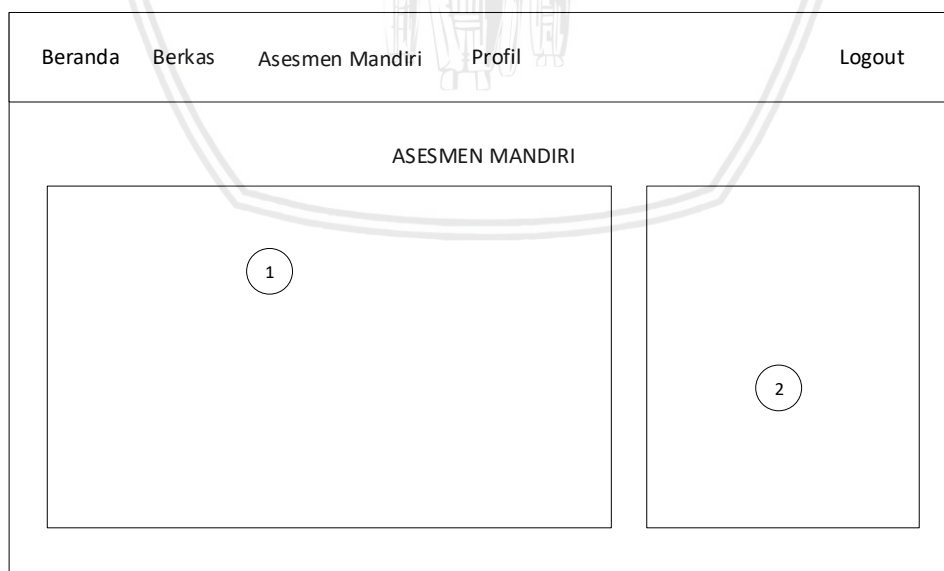
4.2.5 Perancangan Antarmuka

4.2.5.1 Rancangan Antarmuka Melakukan Proses Asesmen Mandiri



Gambar 4.21 Rancangan Antarmuka Beranda Siswa

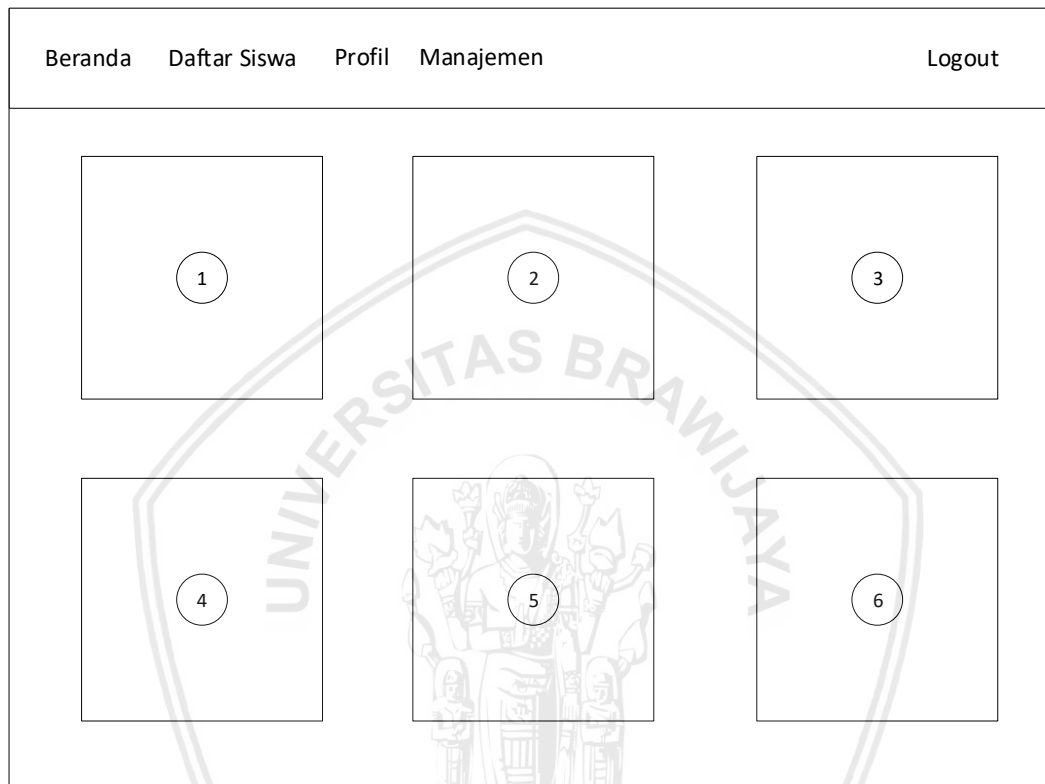
Pada gambar 4.21 merupakan rancangan antarmuka beranda siswa. Dalam antarmuka tersebut terdapat beberapa menu yakni beranda, berkas, asesmen mandiri, umpan balik, profil dan *logout*. Sedangkan 3 (tiga) persegi yang berada di tengah akan berfungsi sebagai penunjuk status proses asesmen yang dilakukan siswa tersebut. Untuk antarmuka asesmen mandiri, dapat dilihat pada gambar 4.22.



Gambar 4.22 Rancangan Antarmuka Asesmen Mandiri

Pada gambar 4.22 merupakan rancangan antarmuka asesmen mandiri. Untuk nomor 1 (satu) berfungsi sebagai *form* asesmen mandiri yang terdiri dari pertanyaan dan *form* jawaban untuk setiap kode unit kompetensi. Sedangkan nomor 2 (dua) berfungsi sebagai informasi halaman serta petunjuk pengisian form jawaban yang berada pada nomor 1 (satu).

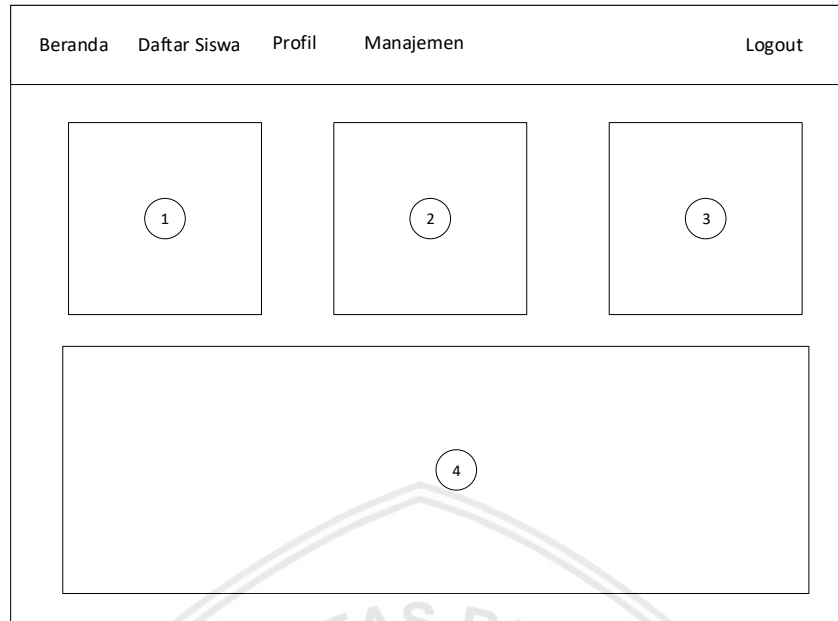
4.2.5.2 Rancangan Antarmuka Melihat Daftar Pemohon Sertifikasi



Gambar 4.23 Rancangan Antarmuka Beranda Asesor

Pada gambar 4.23 merupakan rancangan antarmuka beranda asesor. Terdapat 3 (tiga) menu yakni beranda, daftar siswa dan logout. Untuk persegi nomor 1 (satu) menunjukkan informasi jumlah pemohon sertifikasi yang belum melakukan proses *upload*. Nomor 2 (dua) menunjukkan jumlah siswa yang telah mengunggah dokumen dan menunggu verifikasi asesor. Nomor 3 (tiga) menunjukkan jumlah siswa yang melakukan proses pengisian asesmen mandiri. Nomor 4 (empat) menunjukkan jumlah siswa yang telah siap untuk dilakukan proses asesmen oleh asesor. Nomor 5 (lima) menunjukkan siswa yang sedang mengerjakan uji kompetensi. Sedangkan nomor 6 (enam) menunjukkan jumlah siswa yang telah selesai mengikuti semua proses asesmen.

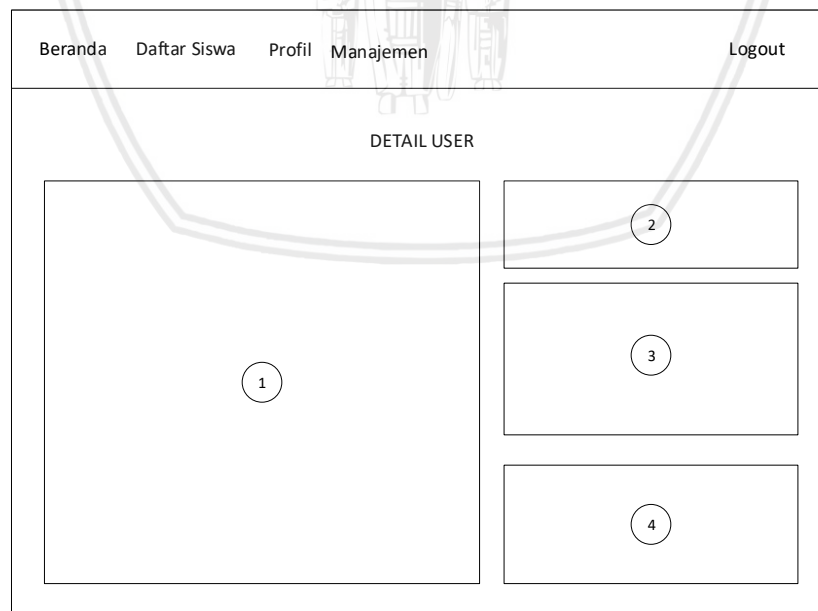
Sedangkan untuk melihat daftar siswa peserta sertifikasi profesi, asesor dapat menekan tombol daftar siswa dan sistem akan menampilkan halaman daftar siswa. Rancangan halaman daftar siswa dapat dilihat pada gambar 4.24.



Gambar 4.24 Rancangan Antarmuka Daftar Siswa

Pada gambar 4.24 merupakan rancangan antarmuka daftar siswa. Terdapat 4 (empat) fungsi utama pada halaman antarmuka daftar siswa tersebut. Fungsi pertama ditunjukkan pada nomor 1 (satu) ialah menu untuk menampilkan data siswa berdasarkan status asesmen. Nomor 2 (dua) ialah menu untuk menampilkan daftar asesor yang ada. Nomor 3 (tiga) ialah menu untuk melakukan pencarian data. Sedangkan nomor 4 (empat) ialah lokasi daftar siswa ditampilkan.

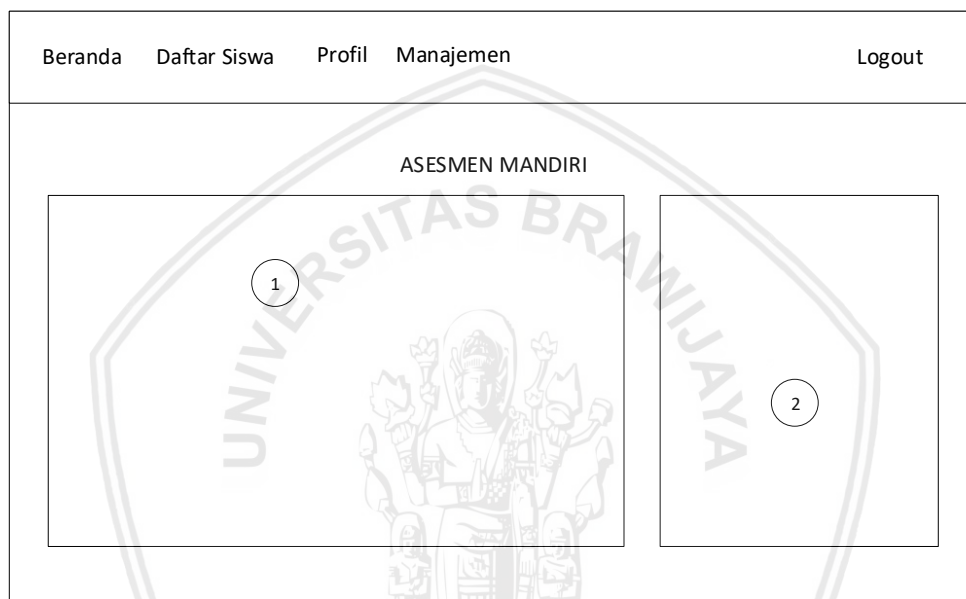
4.2.5.3 Rancangan Antarmuka Menilai Asesmen Mandiri Siswa



Gambar 4.25 Rancangan Antarmuka Detail *User*

Pada gambar 4.25 merupakan rancangan antarmuka detail *user*. Sebelum melakukan proses asesmen, asesor diharuskan membuka detail *user* yang berisi detail informasi siswa sebelum dilakukan asesmen. Nomor 1 (satu) menunjukkan detail biodata siswa, nomor 2 (dua) menunjukkan status beserta foto siswa tersebut dalam proses asesmen. Nomor 3 (tiga) menunjukkan dokumen pendukung yang telah diunggah siswa. Terakhir, nomor 4 (empat) menunjukkan daftar *form* kelengkapan proses asesmen.

Untuk melakukan proses asesmen, asesor mendapatkan informasi status siswa yang akan diasesmen dari bagian nomor 3 (tiga) dan menekan tombol mulai asesmen. Form asesmen dapat dilihat pada gambar 4.26.



Gambar 4.26 Rancangan Antarmuka Asesmen Mandiri (Asesor)

Pada awal proses asesmen yakni asesmen mandiri, tampilan antarmuka kurang lebih sama dengan tampilan antarmuka asesmen mandiri siswa. Namun, setelah menyelesaikan asesmen mandiri tersebut, asesor akan diarahkan menuju asesmen-asesmen lanjutan hingga proses asesmen selesai.

4.3 Implementasi

Implementasi merupakan salah satu proses dalam tahapan *construction* dari SDLC *waterfall*. Pada proses ini digunakan proses pembuatan dan pembangunan perangkat lunak dari hasil perancangan yang telah dilakukan. Hal yang dilakukan dalam tahap ini ialah implementasi antarmuka, fungsi, serta *database*.

4.3.1 Implementasi Antarmuka

Pada implementasi antarmuka menunjukkan hasil implementasi rancangan antarmuka yang telah dibuat sebelumnya. Berikut merupakan hasil implementasi antarmuka untuk asesmen mandiri, melihat daftar pemohon sertifikasi, melakukan proses asesmen dan mencari hasil asesmen.

4.3.1.1 Implementasi Antarmuka Melakukan Proses Asesmen Mandiri

Beranda
Berkas
Asesmen Mandiri
Profil
Logout

STATUS ASESMEN

1. Upload berkas pendukung
(Berkas pendukung digunakan untuk verifikasi pendaftaran asesmen)

Anda belum mengunggah berkas pendukung asesmen sertifikasi profesi, mohon segera mengunggah dokumen pendukung sertifikasi profesi [disini](#).

2. Melakukan asesmen mandiri
(Asesmen mandiri dilakukan jika status asesmen anda telah selesai diverifikasi)

Mohon menunggu untuk menyelesaikan proses sebelumnya.

3. Mengerjakan Uji Kompetensi
(Menu akan terbuka jika asesor telah menyelesaikan penilaian asesmen mandiri)

Mohon menunggu untuk menyelesaikan proses sebelumnya.

Gambar 4.27 Implementasi Antarmuka Beranda Siswa

Pada gambar 4.27 merupakan implementasi antarmuka beranda siswa. Dalam antarmuka tersebut terdapat beberapa menu yakni beranda, berkas, asesmen mandiri, umpan balik, profil dan logout. Untuk implementasi antarmuka asesmen mandiri, dapat dilihat pada gambar 4.28.

Beranda
Berkas
Asesmen Mandiri
Profil
Logout

ASESMEN MANDIRI

Kode Unit Kompetensi	TKK-JK01.006.01		
Judul Unit Kompetensi	Menerapkan prosedur kesehatan, keselamatan, dan keamanan kerja		
Elemen Kompetensi	1. Menetapkan hal-hal yang berhubungan dengan kesehatan dan keselamatan kerja di lingkungan kerja		
Daftar Pertanyaan (Asesmen Mandiri Self Assessment) Apakah anda dapat / mampu:			
	Penilaian		Bukti Pendukung
	K	BK	
1.1. Apakah anda dapat menetapkan pengawasan kesehatan dan keselamatan kerja sesuai undang-undang kesehatan dan keselamatan kerja.	☐	☐	
1.2. Apakah anda dapat meninjau dan melaporkan prosedur kesehatan dan keselamatan kerja di lingkungan kerja untuk memperbaiki masalah yang ada pada pengawasan	☐	☐	
1.3. Apakah anda dapat melaksanakan prosedur kesehatan dan keselamatan kerja untuk menjamin keamanan di lingkungan kerja	☐	☐	
Elemen Kompetensi	2. Mendokumentasikan dan menyebarkan syarat-syarat kesehatan dan keselamatan kerja		
Daftar Pertanyaan (Asesmen Mandiri Self Assessment) Apakah anda dapat / mampu:			
	Penilaian		Bukti Pendukung
	K	BK	
2.1. Apakah anda dapat mengumpulkan informasi yang berhubungan dengan peraturan kesehatan dan keselamatan kerja dan syarat-syaratnya	☐	☐	
2.2. Apakah anda dapat menetapkan dan mendokumentasikan peraturan-peraturan kesehatan dan keselamatan yang berkaitan dengan teknologi informasi di area klien	☐	☐	
2.3. Apakah anda dapat mengajukan dokumen pada pengawas untuk diverifikasi	☐	☐	
2.4. Apakah anda dapat menyebarkan dokumen kesehatan dan keselamatan kerja pada semua pos kerja	☐	☐	
2.5. Apakah anda dapat memperbaharui dan mendistribusikan dokumen kesehatan dan keselamatan kerja yang berhubungan dengan teknologi informasi	☐	☐	
Elemen Kompetensi	3. Menyediakan saran-saran ergonomis dasar		
Daftar Pertanyaan (Asesmen Mandiri Self Assessment) Apakah anda dapat / mampu:			
	Penilaian		Bukti Pendukung
	K	BK	
3.1. Apakah anda dapat menilai syarat-syarat ergonomis dari klien	☐	☐	
3.2. Apakah anda dapat menyediakan saran untuk klien berdasarkan persyaratan dari vendor, kejadian ditempat kerja, serta informasi kesehatan dan keselamatan kerja terbaru	☐	☐	
3.3. Apakah anda dapat mendokumentasikan dan memberikan saran pada klien dan pengawas	☐	☐	

PERINGATAN , Pastikan mengisi dengan benar, karena masing-masing jawaban yang telah diisi tidak dapat diulangi!

Pada bagian ini, anda diminta untuk menilai diri sendiri terhadap unit (unit-unit) kompetensi yang akan di-ases.

1. Pelajari seluruh standar Kriteria Unjuk Kerja (KUK), batasan variabel, panduan penilaian dan aspek kritis serta yakinkan bahwa anda sudah benar-benar memahami seluruh isinya.
2. Laksanakan penilaian mandiri dengan mempelajari dan menilai kemampuan yang anda miliki secara obyektif terhadap seluruh daftar pertanyaan yang ada, serta tentukan apakah sudah kompeten (K) atau belum kompeten (BK) serta tuliskan bukti-bukti pendukung yang anda anggap relevan terhadap setiap elemen/KUK unit kompetensi.

1 2 3 4 5 6 7 8

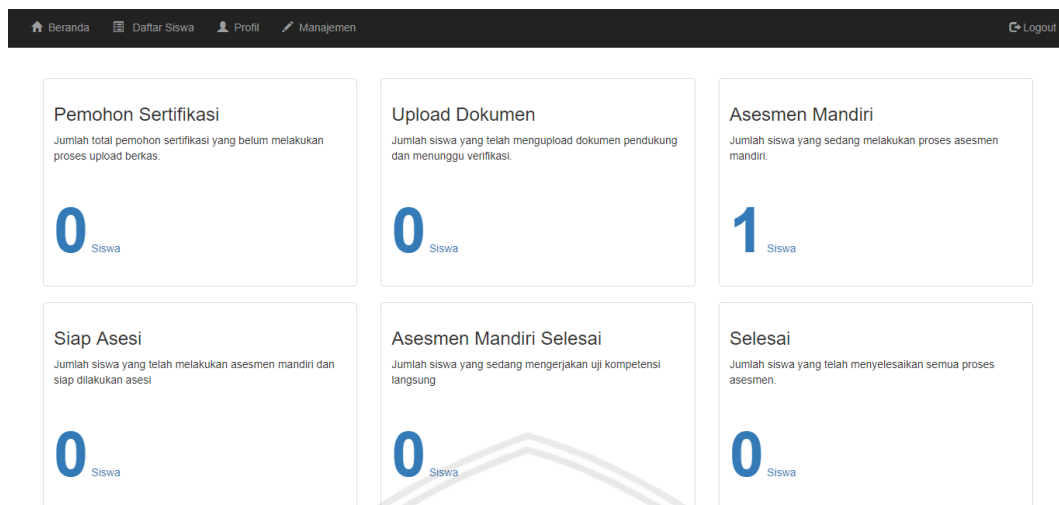
Keterangan Asesmen Mandiri

Selanjutnya

Asesmen Mandiri LSP SMK Negeri 4 Malang

Gambar 4.28 Implementasi Antarmuka Asesmen Mandiri

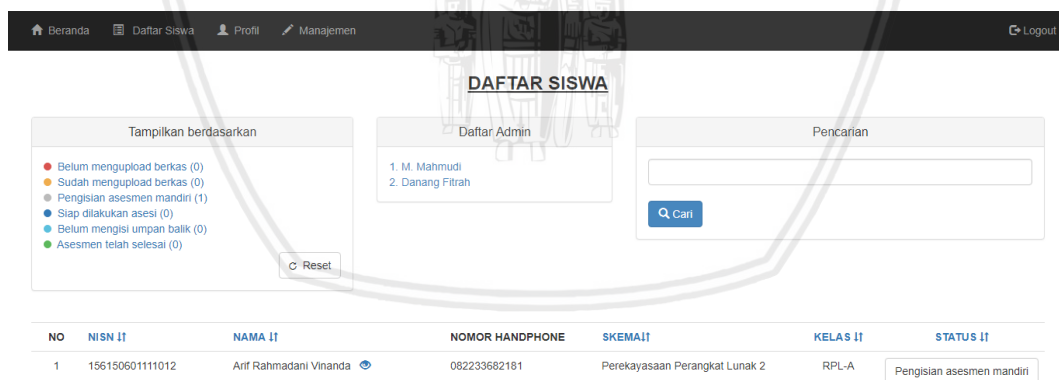
4.3.1.2 Implementasi Antarmuka Melihat Daftar Pemohon Sertifikasi



Gambar 4.29 Implementasi Antarmuka Beranda Asesor

Pada gambar 4.29 merupakan implementasi antarmuka beranda asesor. Pada beranda asesor tersebut terdapat 6 (enam) indikator asesmen dalam angka yang tersedia untuk mempermudah asesor.

Sedangkan untuk melihat daftar siswa peserta sertifikasi profesi, asesor dapat menekan tombol daftar siswa dan sistem akan menampilkan halaman daftar siswa. Implementasi halaman daftar siswa dapat dilihat pada gambar 4.30.



Gambar 4.30 Implementasi Antarmuka Daftar Siswa

4.3.1.3 Implementasi Antarmuka Menilai Asesmen Mandiri Siswa

DETAIL SISWA

NISN: 156150601111012

Nama: Arif Rahmadani Vinanda

Email: arifrahmadaniwinanda@gmail.com

Alamat: Jl. Ronggowuni 132, Tumpang

Kode Pos: 65156

Nomor Handphone: 082233682181

Jenis Kartu Identitas: Kartu Pelajar

Nomor Kartu Identitas: 156150601111012

Provinsi Lahir: JAWA TIMUR

Kota Lahir: KABUPATEN MALANG

Tanggal Lahir: 1997-02-09

Jenis Kelamin: Laki-laki

Skema: Perekayasaan Perangkat Lunak 2

Kelas: RPL-A

Peserta Asesi LSP SMK Negeri 4 Malang

Status: **Pengisian asesmen mandiri**

[Mulai Asesmen](#)

Status Asesmen

NO	Jenis Dokumen	Action
1	Raport Semester 1-5	Lihat

Dokumen Pendukung

[FR-APL-01](#) [Formulir Permohonan Sertifikasi Kompetensi](#)

[FR-APL-02](#) [Asesmen Mandiri](#)

[Daftar Dokumen](#)

Gambar 4.31 Implementasi Antarmuka Detail User

Pada gambar 4.31 merupakan implementasi antarmuka detail *user*. Sebelum melakukan proses asesmen, asesor diharuskan membuka detail *user* yang berisi detail informasi siswa sebelum dilakukan asesmen.

Untuk melakukan proses asesmen, asesor mendapatkan informasi status siswa yang akan diasesmen dari bagian status dan menekan tombol mulai asesmen. *Form* penilaian asesmen dapat dilihat pada gambar 4.32.

1. ASESMEN MANDIRI

Kode Unit Kompetensi	TIK-JK01.006.01									
Judul Unit Kompetensi	Menerapkan prosedur kesehatan, keselamatan, dan keamanan kerja									
Elemen Kompetensi	1. Menetapkan hal-hal yang berhubungan dengan kesehatan dan keselamatan kerja di lingkungan kerja									
Daftar Pertanyaan (Asesmen Mandiri/ Self Assessment) Apakah anda dapat / mampu:	Penilaian		Bukti Pendukung	Diisi Asesor						
	K	BK		V	A	T	M	Check All		
1.1. Apakah anda dapat menetapkan pengawasan kesehatan dan keselamatan kerja sesuai undang-undang kesehatan dan keselamatan kerja	✓			☐	☐	☐	☐	☐		
1.2. Apakah anda dapat meninjau dan melaporkan prosedur kesehatan dan keselamatan kerja di lingkungan kerja untuk memperbaiki masalah yang ada pada pengawas	✓			☐	☐	☐	☐	☐		
1.3. Apakah anda dapat melaksanakan prosedur kesehatan dan keselamatan kerja untuk menjamin keamanan di lingkungan kerja	✓			☐	☐	☐	☐	☐		
Elemen Kompetensi	2. Mendokumentasikan dan menyebarkan syarat-syarat kesehatan dan keselamatan kerja									
Daftar Pertanyaan (Asesmen Mandiri/ Self Assessment) Apakah anda dapat / mampu:	Penilaian		Bukti Pendukung	Diisi Asesor						
	K	BK		V	A	T	M	Check All		
2.1. Apakah anda dapat mengumpulkan informasi yang berhubungan dengan peraturan kesehatan dan keselamatan kerja dan syarat-syaratnya	✓			☐	☐	☐	☐	☐		
2.2. Apakah anda dapat menetapkan dan mendokumentasikan peraturan-peraturan kesehatan dan keselamatan yang berkaitan dengan teknologi informasi di area klien	✓			☐	☐	☐	☐	☐		
2.3. Apakah anda dapat mengajukan dokumen pada pengawas untuk diverifikasi	✓			☐	☐	☐	☐	☐		
2.4. Apakah anda dapat menyebarkan dokumen kesehatan dan keselamatan kerja pada semua pos kerja	✓			☐	☐	☐	☐	☐		
2.5. Apakah anda dapat memperbaharui dan mendiseminasi dokumen-dokumen kesehatan dan keselamatan kerja yang berhubungan dengan teknologi informasi	✓			☐	☐	☐	☐	☐		
Elemen Kompetensi	3. Menyediakan saran-saran ergonomis dasar									
Daftar Pertanyaan (Asesmen Mandiri/ Self Assessment) Apakah anda dapat / mampu:	Penilaian		Bukti Pendukung	Diisi Asesor						
	K	BK		V	A	T	M	Check All		
3.1. Apakah anda dapat menilai syarat-syarat ergonomis dari klien	✓			☐	☐	☐	☐	☐		
3.2. Apakah anda dapat menyediakan saran untuk klien berdasarkan persyaratan dari vendor, kebijakan ditempat kerja, serta informasi kesehatan dan keselamatan kerja terbaru	✓			☐	☐	☐	☐	☐		
3.3. Apakah anda dapat mendokumentasikan dan memberikan saran pada klien dan pengawas	✓			☐	☐	☐	☐	☐		

[Selanjutnya](#)

Asesmen Mandiri LSP SMK Negeri 4 Malang

Pada bagian ini, anda diminta untuk menilai diri sendiri terhadap unit (unit-unit) kompetensi yang akan di-ases.

1. Pelajari seluruh standar Kriteria Unjuk Kerja (KUK), batasan variabel, panduan penilaian dan aspek kritis serta yakinkan bahwa anda sudah benar-benar memahami seluruh isinya.
2. Laksanakan penilaian mandiri dengan mempelajari dan menilai kemampuan yang anda miliki secara obyektif terhadap seluruh daftar pertanyaan yang ada, serta tentukan apakah sudah kompeten (K) atau belum kompeten (BK) serta tuliskan bukti-bukti pendukung yang anda anggap relevan terhadap setiap elemen/KUK unit kompetensi.

1 2 3 4 5 6 7 8

Ketentuan Asesmen Mandiri

Gambar 4.32 Implementasi Antarmuka Penilaian Asesmen Mandiri Siswa

4.3.2 Implementasi Fungsi

Pada proses implementasi fungsi menjelaskan tentang implementasi kode program dari perancangan yang telah dilakukan. Berikut adalah hasil implementasi program untuk melakukan proses asesmen mandiri, melihat daftar pemohon sertifikasi, menilai asesmen mandiri siswa dan mencari hasil asesmen mandiri.

4.3.2.1 Implementasi Fungsi Melakukan Proses Asesmen Mandiri

Pada proses asesmen mandiri, terdapat satu *controller* yang bertugas untuk pemroses fungsi asesmen mandiri oleh siswa yakni *controller* siswa. Proses tersebut ditangani oleh dua *method* yakni *asesmen_mandiri()* yang bertugas untuk menampilkan soal dan *do_asesmen_mandiri()* yang bertugas menginputkan data. Kedua *controller* tersebut berhubungan dengan 5 (lima) *model* yang bertugas sebagai pemroses data menuju *database*. Berikut merupakan hasil implementasi dari kode program pada tabel 4.13.

Tabel 4.13 Implementasi Controller Siswa.php

No	Source Code
1	<code>public function asesmen_mandiri(){</code>
2	<code> \$halaman = \$this->input->get('halaman');</code>
3	<code> if(\$this->input->get('halaman')== ''){</code>
4	<code> \$url = base_url()."siswa/asesmen_mandiri?halaman=0";</code>
5	<code> echo "<script>>window.location.href='\$url';</script>";</code>
6	<code> }</code>
7	<code> \$user= \$this->session->userdata('user');</code>
8	<code> \$this->load->model('siswa_model');</code>
9	<code> \$status = \$this->siswa_model->get_status(\$user);</code>
10	<code> \$skema_siswa='0';</code>
11	<code> foreach(\$status as \$z){</code>
12	<code> \$skema_siswa=\$z['id_skema'];</code>
13	<code> }</code>
14	<code> \$this->load->model('unit_model');</code>
15	<code> \$data['unit'] = \$this->unit_model->get_unit(\$skema_siswa,</code>
16	<code> \$halaman);</code>
17	<code> foreach(\$data['unit'] as \$d):</code>
18	<code> \$this->load->model('elemen_model');</code>
19	<code> \$data['elemen'][\$d['id_unit']] = \$this->elemen_model-</code>
20	<code> >get_elemen(\$d['id_unit']);</code>
21	<code> foreach(\$data['elemen'][\$d['id_unit']] as \$e):</code>
22	<code> \$this->load->model('pertanyaan_model');</code>
23	<code> \$data['pertanyaan'][\$e['id_elemen']] = \$this-</code>
24	<code> >pertanyaan_model-</code>
25	<code> >get_pertanyaan_by_elemen(\$e['id_elemen']);</code>
26	<code> endforeach;</code>
27	<code> endforeach;</code>
28	<code> foreach(\$status as \$d):</code>
29	<code> if(\$d['id']<3){</code>
30	<code> \$url = base_url()."siswa";</code>
31	<code> echo "<script>alert('Anda belum berhak mengakses</code>
32	<code> halaman ini!');window.location.href='\$url';</script>";</code>
33	<code> }</code>

Tabel 4.13 Implementasi *Controller* Siswa.php (Lanjutan)

No	Source Code
29	else if(\$d['id']==3){
30	\$data['jumlah_unit']=\$this->unit_model->get_jumlah_unit(\$skema_siswa);
31	foreach(\$data['jumlah_unit'] as \$e):
32	\$data['jumlah_halaman'] = \$e['JUMLAH'];
33	endforeach;
34	\$this->load->view('navbar');
35	\$this->load->view('asesmen_mandiri', \$data);
36	}
37	else{
38	\$url = base_url()."siswa";
39	echo "<script>alert('Anda telah menyelesaikan tahap ini!');window.location.href='\$url';</script>";
40	}
41	endforeach;
42	}
43	public function do_asesmen_mandiri(){
44	\$nisn= \$this->session->userdata('user');
45	\$halaman = \$this->input->post('halaman');
46	\$jumlah_halaman = \$this->input->post('jumlah_halaman');
47	\$pertanyaan = \$this->input->post('pertanyaan');
48	\$x=0;
49	foreach(\$pertanyaan as \$id_pertanyaan){
50	\$bukti = \$this->input->post('bukti'.\$id_pertanyaan);
51	\$this->load->model('asesmen_mandiri_model');
52	\$this->asesmen_mandiri_model->set_asesmen_mandiri(\$id_pertanyaan, \$nisn, \$this->input->post(\$id_pertanyaan), \$bukti);
53	if((\$halaman+1)==\$jumlah_halaman){
54	\$this->load->model('siswa_model');
55	\$this->siswa_model->update_status(\$nisn, '4');
56	\$url = base_url()."siswa/";
57	echo "<script>alert('Anda telah menyelesaikan proses asesmen mandiri!');window.location.href='\$url';</script>";
58	}
59	else{
60	\$url = base_url()."siswa/asesmen_mandiri?halaman=" . (\$halaman+1);
61	echo "<script>window.location.href='\$url';</script>";
62	}
63	}
64	}

Pada fungsi melakukan proses asesmen mandiri tabel 4.13 terdapat dua *method* yang berjalan pada controller siswa yakni *asesmen_mandiri* pada baris 1 (satu) hingga 42 (empat puluh dua) dan *do_asesmen_mandiri* pada baris 43 (empat puluh tiga) hingga baris 64 (enam puluh empat). Untuk baris 2 (dua) hingga 6 (enam) merupakan proses pengecekan apakah parameter halaman bernilai kosong. Apabila parameter bernilai kosong maka halaman akan menuju tautan yang telah ditentukan (Halaman pertama asesmen mandiri).

Pada baris 7 (tujuh) hingga 9 (sembilan) merupakan proses untuk mendapatkan data user dan status. Parameter user didapatkan dari session "user" dan parameter status didapatkan dari database menggunakan model *siswa_model*. Kemudian pada baris 11 (sebelas) hingga 12 (dua belas) merupakan proses memasukkan nilai yang didapat dari database menuju variable *skema_siswa*. Pada baris 15 (lima belas) hingga 23 (dua puluh tiga) terdapat proses perulangan menggunakan *foreach* yakni perulangan untuk mendapatkan data dari unit, elemen, dan pertanyaan. Data tersebut didapatkan melalui model *unit_model*, *elemen_model*, dan *pertanyaan_model*. Pada baris 24 (dua puluh empat) hingga 42 (empat puluh dua) terdapat proses pengecekan status dari siswa. Apabila status siswa kurang dari 3 (tiga) maka sistem akan menampilkan peringatan "Anda belum berhak mengakses halaman ini. Sedangkan apabila status siswa adalah 3 (tiga) maka terdapat proses untuk menghitung jumlah halaman yang akan ditampilkan pada view serta terdapat proses untuk menampilkan view navbar dan *asesmen_mandiri*.

Sedangkan untuk method *do_asesmen_mandiri*, pada baris 44 (empat puluh empat) hingga 47 (empat puluh tujuh) terdapat proses untuk mengambil hasil asesmen mandiri dari view *asesmen_mandiri* yakni berupa *nisn*, *halaman*, *jumlah_halaman*, dan *pertanyaan*. Setelah itu, pada baris 50 (lima puluh) hingga 52 (lima puluh dua) terdapat proses untuk memasukkan hasil asesmen mandiri pada database melalui model *asesmen_mandiri_model* dengan method *set_asesmen_mandiri*. Terakhir, pada baris 53 (lima puluh tiga) hingga 62 (enam puluh dua) terdapat proses pengecekan apakah nilai variable *halaman* ditambah 1 (satu) bernilai sama dengan variable *jumlah_halaman*. Apabila benar, maka terdapat proses update status melalui method *update_status* model *siswa_model* dan akan diarahkan menuju tautan berikutnya disertai peringatan "Anda telah menyelesaikan proses asesmen". Apabila pengecekan tersebut bernilai salah, maka halaman akan berpindah pada tautan baru yakni halaman asesmen mandiri selanjutnya. Setelah membahas tentang isi *controller* siswa yang digunakan pada proses asesmen mandiri, selanjutnya akan dijelaskan isi masing-masing *model* yang digunakan meliputi *siswa_model*, *asesmen_mandiri_model*, *unit_model*, *elemen_model*, dan *pertanyaan_model*.

Pada *model* *siswa_model* tabel 4.14 terdapat *method* *get_status* yang berguna untuk mengambil status asesmen mandiri siswa berdasarkan *nisn*. Di dalam *method* tersebut terdapat perintah *query* yang berguna untuk menampilkan id siswa, status siswa dan *id_skema* siswa. Perintah tersebut akan berhubungan dengan *database* untuk mengambil data yang diinginkan. Setelah itu, data tersebut akan dikembalikan kepada *method* *get_status* untuk diakses melalui *controller*.

Tabel 4.14 Implementasi *Model* siswa_model.php

No	Source Code
1	public function get_status(\$nism){
2	\$query = \$this->db->query("SELECT S.id, S.status,
3	u.id_skema FROM siswa U LEFT JOIN status s ON u.status=s.id
4	WHERE u.username = '\$nism');
5	return \$query->result_array();
6	}

Pada tabel 4.15 menjelaskan tentang *model* unit_model. Pada *model* tersebut terdapat 2 (dua) *method* yang digunakan yakni *get_unit* dan *get_jumlah_unit*. *Method* *get_unit* pada baris 1 (satu) hingga 12 (dua belas) merupakan proses untuk mendapatkan data unit dari *database*. Dalam *method* tersebut terdapat pengecekan yakni apabila parameter *id_skema* bernilai kosong, maka akan menampilkan seluruh skema yang ada. Sedangkan apabila parameter *id_skema* mempunyai nilai namun parameter *limit* tidak mempunyai nilai, maka akan menampilkan seluruh skema berdasarkan *id_skema* yang telah ditentukan. Yang terakhir, apabila kedua parameter tersebut mempunyai nilai, maka akan menampilkan data unit berdasarkan skema tertentu dan ditampilkan terbatas berdasarkan nilai parameter *limit*. Sedangkan pada baris 13 (tiga belas) hingga 16 (enam belas) yakni *method* *get_jumlah_unit* merupakan fungsi untuk mendapatkan seluruh jumlah unit berdasarkan skema yang telah ditentukan.

Tabel 4.15 Implementasi *Model* unit_model.php

No	Source Code
1	public function get_unit(\$id_skema=null, \$limit=null){
2	if(\$id_skema==null){
3	\$query = \$this->db->query("SELECT u.id_unit, s.nama as
	skema, u.kode, u.judul, u.jenis FROM unit u LEFT JOIN
	skema s ON u.id_skema = s.id_skema");
4	}
5	else if(\$limit==null){
6	\$query = \$this->db->query("SELECT u.id_unit, s.nama as \
	skema, u.kode, u.judul, u.jenis FROM unit u LEFT JOIN
	skema s ON u.id_skema = s.id_skema WHERE
	u.id_skema='\$id_skema'");
7	}
8	else{
9	\$query = \$this->db->query("SELECT u.id_unit, s.nama as
	skema, u.kode, u.judul, u.jenis FROM unit u LEFT JOIN
	skema s ON u.id_skema = s.id_skema WHERE
	u.id_skema='\$id_skema' LIMIT \$limit, 1 ");
10	}
11	return \$query->result_array();
12	}
13	public function get_jumlah_unit(\$skema){
14	\$query = \$this->db->query("SELECT COUNT(*) AS JUMLAH FROM
	UNIT WHERE id_skema='\$skema'");
15	return \$query->result_array();
16	}

Pada tabel 4.16 merupakan implementasi *method* *get_elemen* pada *model* *elemen_model*. Isi dari *method* tersebut ialah mendapatkan data elemen dari database berdasarkan *id_unit* yang telah ditentukan.

Tabel 4.16 Implementasi *Model* *elemen_model.php*

No	Source Code
1	public function get_elemen(\$id_unit){
2	\$query = \$this->db->query("SELECT e.id_elemen, u.kode, e.nomor, e.elemen, s.nama as skema FROM elemen e LEFT JOIN unit u ON e.id_unit = u.id_unit LEFT JOIN skema s ON u.id_skema=s.id_skema WHERE u.id_unit='\$id_unit' ORDER BY e.nomor ASC");
3	return \$query->result_array();
4	}

Pada tabel 4.17 merupakan implementasi *method* *get_pertanyaan_by_elemen* pada *model* *pertanyaan_model*. Isi dari *method* tersebut ialah mendapatkan data pertanyaan serta *id_form* pertanyaan, elemen, dan kode unit dari *database* dan dicek apakah pertanyaan yang akan ditampilkan sesuai dengan parameter *id_elemen*.

Tabel 4.17 Implementasi *Model* *pertanyaan_model.php*

No	Source Code
1	public function get_pertanyaan_by_elemen(\$id_elemen){
2	\$query = \$this->db->query("SELECT p.id_pertanyaan, s.nama, u.kode, e.elemen, f.id_form, p.id_pertanyaan, p.pertanyaan, p.nomor, p.status FROM pertanyaan p LEFT JOIN elemen e on e.id_elemen=p.id_elemen LEFT JOIN form f on f.id_form=p.id_form LEFT JOIN unit u ON e.id_unit=u.id_unit LEFT JOIN skema s ON u.id_skema = s.id_skema WHERE p.id_elemen='\$id_elemen' ORDER BY p.id_elemen, p.nomor ASC");
3	return \$query->result_array();
4	}

Pada tabel 4.18 merupakan implementasi *method* *set_asesmen_mandiri* yakni *method* untuk menyimpan hasil jawaban asesmen mandiri pada *database*. Pada baris 2 (dua) terdapat *query* untuk menyimpan parameter pada tabel *asesmen_mandiri* kolom *id_form*, *id_pertanyaan*, *nisn*, *jawaban* dan *bukti_pendukung*.

Tabel 4.18 Implementasi *Model* *asesmen_mandiri_model.php*

No	Source Code
1	public function set_asesmen_mandiri(\$id_pertanyaan, \$nisn, \$jawaban, \$bukti_pendukung){
2	\$query = \$this->db->query("INSERT INTO asesmen_mandiri (id_form, id_pertanyaan, nisn, jawaban, bukti_pendukung) VALUES ('2', '\$id_pertanyaan', '\$nisn', '\$jawaban', '\$bukti_pendukung')");
3	}

4.3.2.2 Implementasi Fungsi Melihat Daftar Pemohon Sertifikasi

Pada proses melihat daftar siswa, terdapat satu *controller* yang bertugas untuk pemroses fungsi melihat daftar pemohon sertifikasi oleh asesor yakni *controller* asesor. Proses tersebut ditangani oleh satu *method* yakni *daftar_siswa()* yang bertugas untuk menampilkan daftar pemohon sertifikasi dan dapat dilihat pada tabel 4.19.

Pada baris 2 (empat) hingga 9 (sembilan) merupakan proses untuk mendapatkan data notifikasi, daftar admin, status, mode, dan search. Setelah itu, pada baris 11 (sebelas) hingga 20 (dua puluh) terdapat pengecekan apakah nilai parameter sort bernilai "sort". Apabila benar maka sistem akan mengambil data siswa dari *method* *get_siswa_by_sort*. Apabila salah, maka sistem akan mengambil data siswa dari *method* *get_siswa_by_search*. Setelah itu, sistem akan menampilkan view navbar dan daftar_siswa beserta data yang didapat.

Tabel 4.19 Implementasi Controller Asesor.php

No	Source Code
1	<code>public function daftar_siswa(\$status= NULL, \$sort = NULL, \$sort_column= NULL){</code>
2	<code> \$this->load->model('siswa_model');</code>
3	<code> \$notifikasi=\$this->siswa_model->get_notifikasi();</code>
4	<code> \$data['notifikasi']=\$notifikasi;</code>
5	<code> \$data['param1'] = \$status;</code>
6	<code> \$this->load->model('asesor_model');</code>
7	<code> \$data['daftar_admin'] = \$this->asesor_model->get_asesor();</code>
8	<code> \$mode = \$this->input->get('mode', TRUE);</code>
9	<code> \$search = \$this->input->get('search', TRUE);</code>
10	<code> if(isset(\$status)){</code>
11	<code> if(\$sort=='sort'){</code>
12	<code> \$mode = \$this->input->get('mode', TRUE);</code>
13	<code> \$search = \$this->input->get('search', TRUE);</code>
14	<code> \$result=\$this->siswa_model->get_siswa_by_sort(\$mode, \$status, \$sort_column, \$search);</code>
15	<code> \$data['daftar_siswa'] = \$this->siswa_model->get_siswa_by_sort(\$mode, \$status, \$sort_column, \$search);</code>
16	<code> }</code>
17	<code> else{</code>
18	<code> \$search = \$this->input->get('search', TRUE);</code>
19	<code> \$result=\$this->siswa_model->get_siswa_by_search(\$status, \$search);</code>
20	<code> \$data['daftar_siswa'] = \$this->siswa_model->get_siswa_by_search(\$status, \$search);</code>
21	<code> }</code>
22	<code> } else{</code>
23	<code> \$url = base_url()."asesor/daftar_siswa/0";</code>
24	<code> echo "<script>window.location.href='\$url';</script>";</code>
25	<code> }</code>
26	<code> \$this->load->view('navbar');</code>
27	<code> \$this->load->view('daftar_siswa', \$data);}</code>

Pada tabel 4.20 terdapat implementasi *method* *get_notifikasi* dari *model* *siswa_model* baris 1 (satu) hingga 4 (empat). *Method* tersebut berfungsi untuk mengambil data status siswa dan dikelompokkan berdasarkan *id_status* mulai dari 1 (satu) hingga 6 (enam). Selain itu, terdapat *method* *get_siswa_by_search* baris 5 (lima) hingga 23 (dua puluh tiga) dan *get_siswa_by_sort* baris 24 (dua puluh empat) hingga 42 (empat puluh dua).

Method *get_siswa_by_search* merupakan *method* untuk menampilkan semua daftar siswa atau menampilkan seluruh data siswa pada *view* *daftar_siswa*. Pada baris 7 (tujuh) hingga 12 (dua belas) merupakan fungsi yang dijalankan apabila parameter *search* bernilai kosong. Pada kondisi tersebut, terdapat pengecekan apakah nilai parameter *status* bernilai 0 (nol) atau kosong. Apabila benar, maka sistem akan mengambil data seluruh siswa. Apabila salah, maka sistem akan mengambil data seluruh siswa dengan *status* berdasarkan parameter *status*. Lanjut pada baris 15 (lima belas) hingga 20 (dua puluh) merupakan fungsi yang dijalankan apabila parameter *search* mempunyai nilai. Pada kondisi tersebut, terdapat pengecekan apakah nilai parameter *status* bernilai 0 (nol) atau kosong. Apabila benar, maka sistem akan mengambil data siswa berdasarkan kata kunci yang telah ditentukan. Apabila salah, maka sistem akan mengambil data siswa berdasarkan *status* tertentu dan kata kunci yang telah ditentukan.

Terakhir, *Method* *get_siswa_by_sort* merupakan *method* untuk menampilkan semua daftar siswa atau menampilkan seluruh data siswa pada *view* *daftar_siswa* berdasarkan urutan. Pada baris 26 (dua puluh enam) hingga 31 (tiga puluh satu) merupakan fungsi yang dijalankan apabila parameter *search* bernilai kosong. Pada kondisi tersebut, terdapat pengecekan apakah nilai *variable filter* tidak bernilai 0 (nol) atau kosong. Apabila benar, maka sistem akan mengambil data seluruh siswa berdasarkan *status* yang telah ditentukan dan urut sesuai urutan yang dikehendaki. Apabila salah, maka sistem akan mengambil data seluruh siswa dan diurutkan berdasarkan urutan yang telah dikehendaki. Lanjut pada baris 34 (tiga puluh empat) hingga 40 (empat puluh) merupakan fungsi yang dijalankan apabila parameter *search* mempunyai nilai. Pada kondisi tersebut, terdapat pengecekan apakah nilai *filter* mempunyai nilai atau tidak kosong. Apabila benar, maka sistem akan mengambil data siswa berdasarkan kata kunci yang telah ditentukan dan berdasarkan *status* yang telah ditentukan serta urut sesuai dengan urutan yang telah dikehendaki. Apabila salah, maka sistem akan mengambil data siswa berdasarkan kata kunci yang telah ditentukan dan urut sesuai urutan yang telah dikehendaki.

Tabel 4.20 Implementasi *Model* siswa_model.php

No	Source Code
1	public function get_notifikasi(){
2	<pre> \$query = \$this->db->query("select t1.jumlah1, t2.jumlah2, t3.jumlah3, t4.jumlah4, t5.jumlah5, t6.jumlah6 from (select COUNT(status) as jumlah1 FROM siswa WHERE status='1') as t1, (select COUNT(status) as jumlah2 FROM siswa WHERE status='2') as t2, (select COUNT(status) as jumlah3 FROM siswa WHERE status='3') as t3, (select COUNT(status) as jumlah4 FROM siswa WHERE status='4') as t4, (select COUNT(status) as jumlah5 FROM siswa WHERE status='5') as t5, (select COUNT(status) as jumlah6 FROM siswa WHERE \ status='6') as t6"); return \$query->result_array(); } </pre>
3	
4	
5	public function get_siswa_by_search(\$status=NULL,
6	\$search=NULL){
7	<pre> if(\$search==''){ if(\$status==0 \$status==null){ </pre>
8	<pre> \$query = \$this->db->query("SELECT B.nisn, u.nama, u.nomor_handphone, P.nama as skema, B.kelas, S.id, S.status FROM siswa B, skema P, user U, status S WHERE B.id_skema=P.id_skema and b.nisn = u.username AND b.status = s.id "); </pre>
9	
10	<pre> } else{ </pre>
11	<pre> \$query = \$this->db->query("SELECT B.nisn, u.nama, u.nomor_handphone, P.nama as skema, B.kelas, S.id, S.status FROM siswa B, skema P, user U, status S WHERE B.id_skema=P.id_skema and b.nisn = u.username AND b.status = s.id AND b.status = '\$status' "); </pre>
12	
13	<pre> } } else{ </pre>
14	
15	<pre> if(\$status==0 \$status==null){ </pre>
16	<pre> \$query = \$this->db->query("SELECT B.nisn, u.nama, u.nomor_handphone, P.nama as skema, B.kelas, S.id, S.status FROM siswa B LEFT JOIN skema P ON B.id_skema=P.id_skema LEFT JOIN user U ON b.nisn = u.username LEFT JOIN status S ON b.status = s.id WHERE B.nisn LIKE '".\$search."' OR u.nama LIKE '".\$search."' OR u.nomor_handphone LIKE '".\$search."' OR u.email LIKE '".\$search."' OR u.alamat LIKE '".\$search."' OR u.nomor_kartu_identitas LIKE '".\$search."' OR u.tanggal_lahir LIKE '".\$search."' OR u.jenis_kelamin LIKE '".\$search."' OR p.nama LIKE '".\$search."'"); </pre>
17	
18	<pre> } else{ </pre>
19	<pre> \$query = \$this->db->query("SELECT B.nisn, u.nama, u.nomor_handphone, P.nama as skema, B.kelas, S.id, </pre>

Tabel 4.20 Implementasi *Model* siswa_model.php (Lanjutan)

No	Source Code
	<pre> S.status FROM siswa B LEFT JOIN skema P ON \ B.id_skema=P.id_skema LEFT JOIN user U ON b.nisn = u.username LEFT JOIN status S ON b.status = s.id WHERE (b.status='\$status') AND (B.nisn LIKE '".\$search."' OR u.nama LIKE '".\$search."' OR u.nomor_handphone LIKE '".\$search."' OR u.email LIKE '".\$search."' OR u.alamat LIKE '".\$search."' OR u.nomor_kartu_identitas LIKE '".\$search."' OR u.tanggal_lahir LIKE '".\$search."' OR u.jenis_kelamin LIKE '".\$search."' OR p.nama LIKE '".\$search."'); 20 } 21 } 22 return \$query->result_array(); 23 } 24 public function get_siswa_by_sort(\$sort_mode, \$status=null, \$sort_column, \$search=null){ 25 if(\$search==''){ 26 if(\$status!=0){ 27 \$query = \$this->db->query("SELECT B.nisn, u.nama, u.nomor_handphone, P.nama as skema, B.kelas, S.id, S.status FROM siswa B, skema P, user U, status S WHERE B.id_skema=P.id_skema and b.nisn = u.username AND b.status = s.id AND b.status='\$status' ORDER BY \$sort_column \$sort_mode"); 28 } 29 else{ 30 \$query = \$this->db->query("SELECT b.nisn, u.nama, u.nomor_handphone, P.nama as skema, B.kelas, S.id, S.status FROM siswa B, skema P, user U, status S WHERE B.id_skema=P.id_skema and b.nisn = u.username AND b.status = s.id ORDER BY \$sort_column \$sort_mode"); 31 } 32 } 33 else{ 34 if(\$status!=0){ 35 \$query = \$this->db->query("SELECT B.nisn, u.nama, u.nomor_handphone, P.nama as skema, B.kelas, S.id, S.status FROM siswa B LEFT JOIN skema P ON B.id_skema=P.id_skema LEFT JOIN user U ON b.nisn = u.username LEFT JOIN status S ON b.status = s.id WHERE (b.status='\$status) AND (B.nisn LIKE '".\$search."' OR OR u.nama LIKE '".\$search."' OR u.nomor_handphone LIKE '".\$search."' OR u.email LIKE '".\$search."' OR u.alamat LIKE '".\$search."' OR u.nomor_kartu_identitas LIKE '".\$search."' OR u.tanggal_lahir LIKE '".\$search."' OR u.jenis_kelamin LIKE '".\$search."' OR p.nama LIKE '".\$search."') ORDER BY \$sort_column \$sort_mode"); 36 } 37 else{ </pre>

Tabel 4.20 Implementasi *Model* siswa_model.php (Lanjutan)

No	Source Code
38	<code>\$query = \$this->db->query("SELECT B.nisn, u.nama, u.nomor_handphone, P.nama as skema, B.kelas, S.id, S.status FROM siswa B LEFT JOIN skema P ON B.id_skema=P.id_skema LEFT JOIN user U ON b.nisn = u.username LEFT JOIN status S ON b.status = s.id WHERE B.nisn LIKE '%" . \$search . "%' OR u.nama LIKE '%" . \$search . "%' OR u.nomor_handphone LIKE '%" . \$search . "%' OR u.email LIKE '%" . \$search . "%' OR u.alamat LIKE '%" . \$search . "%' OR u.nomor_kartu_identitas LIKE '%" . \$search . "%' OR u.tanggal_lahir LIKE '%" . \$search . "%' OR u.jenis_kelamin LIKE '%" . \$search . "%' OR p.nama LIKE '%" . \$search . "%' ORDER BY \$sort_column \$sort_mode");</code>
39	<code>}</code>
40	<code>}</code>
41	<code>return \$query->result_array();</code>
42	<code>}</code>

4.3.2.3 Implementasi Fungsi Menilai Asesmen Mandiri Siswa

Pada proses penilaian asesmen mandiri siswa, terdapat satu *controller* yang bertugas untuk menampilkan hasil asesmen mandiri siswa serta menilai asesmen mandiri tersebut. Proses tersebut ditangani oleh dua *method* yakni *asesmen_mandiri()* yang bertugas untuk menampilkan hasil asesmen mandiri siswa beserta pertanyaan asesmen mandiri serta *do_asesmen_mandiri()* yang bertugas untuk menyimpan hasil penilaian asesmen mandiri oleh asesor. *Controller* tersebut berhubungan dengan 6 (enam) *model* yang bertugas sebagai pemroses data menuju *database*. Berikut merupakan hasil implementasi dari kode program pada tabel 4.21. Untuk *model* dalam proses menampilkan soal asesmen akan sama dengan tabel 4.14 hingga 4.17. Namun yang membedakan ialah pada baris 19 (Sembilan belas) yakni terdapat *method* untuk mendapatkan hasil asesmen siswa berdasarkan nisn. Data asesmen mandiri digunakan untuk asesor mempertimbangkan penilaian asesmen mandirinya. Selain itu, *view* yang digunakan ialah navbar dan admin_asesmen_mandiri serta pada *method* *do_asesmen_mandiri* *method* yang digunakan untuk menyimpan data bukan *set_asesmen_mandiri* namun menggunakan *method* *update_asesmen_mandiri*.

Tabel 4.21 Implementasi *Controller* Asesor.php

No	Source Code
1	<code>public function asesmen_mandiri(\$nisn=null, \$input=null){</code>
2	<code>\$this->load->model('asesmen_mandiri_model');</code>
3	<code>\$status_asesmen = \$this->asesmen_mandiri_model->get_status_asesmen_mandiri(\$nisn);</code>
4	<code>foreach(\$status_asesmen as \$d):</code>
5	<code>if(!isset(\$d['asesor'])){</code>
6	<code>\$halaman = \$this->input->get('halaman');</code>
7	<code>if(\$this->input->get('halaman')=='){</code>
8	<code></code>

Tabel 4.21 Implementasi *Controller* asesor.php (Lanjutan)

No	Source Code
8	<code>\$url = base_url()."asesor/asesmen_mandiri/</code>
9	<code>\\$nism?halaman=0";</code>
10	<code>echo "<script>window.location.href='\$url';</code>
11	<code></script>";</code>
12	<code>} \$user= \$nism;</code>
13	<code>\$this->load->model('siswa_model');</code>
14	<code>\$status = \$this->siswa_model->get_status(\$nism);</code>
15	<code>\$skema_siswa='0';</code>
16	<code>foreach(\$status as \$z){</code>
17	<code> \$skema_siswa=\$z['id_skema'];</code>
18	<code>}</code>
19	<code>\$data['nism'] = \$nism;</code>
20	<code>\$data['mandiri'] = \$this->asesmen_mandiri_model-</code>
21	<code>>get_asesmen_mandiri(\$nism);</code>
22	<code>\$this->load->model('unit_model');</code>
23	<code>\$data['unit'] = \$this->unit_model-</code>
24	<code>>get_unit(\$skema_siswa, \$halaman);</code>
25	<code>foreach(\$data['unit'] as \$d):</code>
26	<code> \$this->load->model('elemen_model');</code>
27	<code> \$data['elemen'][\$d['id_unit']] = \$this-</code>
28	<code> >elemen_model->get_elemen(\$d['id_unit']);</code>
29	<code> foreach(\$data['elemen'][\$d['id_unit']] as \$e):</code>
30	<code> \$this->load->model('pertanyaan_model');</code>
31	<code> \$data['pertanyaan'][\$e['id_elemen']] = \$this-</code>
32	<code> >pertanyaan_model->get_pertanyaan_by_elemen</code>
33	<code> (\$e['id_elemen']);</code>
34	<code> endforeach;</code>
35	<code>endforeach;</code>
36	<code>foreach(\$status as \$d):</code>
37	<code> if(\$d['id']<4){</code>
38	<code> \$url = base_url()."asesor/detail_siswa/\$nism";</code>
39	<code> echo "<script>alert('Anda belum berhak mengakses</code>
40	<code> halaman ini!');window.location.href='\$url';</code>
41	<code> </script>";</code>
42	<code> }</code>
43	<code> else if(\$d['id']==4){</code>
44	<code> \$this->load->model('unit_model');</code>
45	<code> \$data['jumlah_unit']= \$this->unit_model-</code>
46	<code> >get_jumlah_unit(\$skema_siswa);</code>
47	<code> foreach(\$data['jumlah_unit'] as \$e):</code>
48	<code> \$data['jumlah_halaman'] = \$e['JUMLAH'];</code>
49	<code> endforeach;</code>
50	<code> \$this->load->view('navbar');</code>
	<code> \$this->load->view('admin_asesmen_mandiri', \$data);</code>
	<code> }</code>
	<code> else{</code>
	<code> \$url = base_url()."asesor/detail_siswa/\$nism";</code>
	<code> echo "<script>alert('Anda telah menyelesaikan tahap</code>
	<code> ini!');window.location.href='\$url';</script>";</code>
	<code> }</code>
	<code>endforeach;</code>
	<code>}</code>
	<code>else{</code>

Tabel 4.21 Implementasi *Controller* asesor.php (Lanjutan)

No	Source Code
51	<code>\$url = base_url()."asesor/detail_siswa/\$nispn";</code>
52	<code>echo "<script>alert('Anda telah menyelesaikan tahap ini!');window.location.href='\$url';</script>";</code>
53	<code>}</code>
54	<code>endforeach;</code>
55	<code>}</code>
56	<code>public function do_asesmen_mandiri(\$nispn){</code>
57	<code>\$halaman = \$this->input->post('halaman');</code>
58	<code>\$jumlah_halaman = \$this->input->post('jumlah_halaman');</code>
59	<code>\$bukti = \$this->input->post('bukti');</code>
60	<code>\$pertanyaan = \$this->input->post('pertanyaan');</code>
61	<code>\$selesai = \$this->input->post('selesai');\$x=0;</code>
62	<code>foreach(\$pertanyaan as \$id_pertanyaan){</code>
63	<code>\$this->load->model('asesmen_mandiri_model');</code>
64	<code>\$total_jawaban="";</code>
65	<code>foreach(\$this->input->post(\$id_pertanyaan) as \$jawaban){</code>
66	<code>\$total_jawaban=\$total_jawaban."+ ".\$jawaban;</code>
67	<code>}</code>
68	<code>\$this->asesmen_mandiri_model->update_asesmen_mandiri(\$id_pertanyaan, \$nispn, \$this->session->userdata('user'), \$total_jawaban);</code>
69	<code>if((\$halaman+1)==\$jumlah_halaman){</code>
70	<code>\$this->do_update_status_siswa(\$nispn, '4');</code>
71	<code>\$url = base_url()."asesor/daftar_siswa";</code>
72	<code>echo "<script>window.location.href='\$url';</script>";</code>
73	<code>}</code>
74	<code>else{</code>
75	<code>\$url = base_url()."asesor/asesmen_mandiri/\$nispn?halaman="</code>
76	<code>."+(\$halaman+1);</code>
77	<code>echo "<script>window.location.href='\$url';</script>";</code>
78	<code>}</code>
79	<code>}</code>

Pada tabel 4.22 merupakan implementasi *method* `get_asesmen_mandiri`, `get_status_asesmen_mandiri`, dan `update_status_asesmen_mandiri`. *Method* `get_status_asesmen_mandiri` baris 1 (satu) hingga 4 (empat) merupakan *method* yang berfungsi untuk mendapatkan data hasil asesmen mandiri siswa berdasarkan nispn siswa yang dimaksud. Sedangkan *method* `get_status_asesmen_mandiri` baris 5 (lima) hingga 8 (delapan) adalah *method* yang berfungsi untuk mendapatkan status asesmen mandiri siswa yang ditunjukkan dengan telah terisinya kolom nip yang menandi bahwa hasil asesmen mandiri siswa telah dinilai oleh asesor. Yang terakhir ialah *method* `update_asesmen_mandiri` baris 9 (sembilan) hingga 12 (dua belas) adalah *method* yang berfungsi untuk mengubah hasil asesmen mandiri siswa dan ditambahkan penilaian dari asesor. Perubahan tersebut meliputi penambahan nip asesor, status, dan tanggal penilaian asesmen.

Tabel 4.22 Implementasi *Model* asesmen_mandiri_model.php

No	Source Code
1	public function get_asesmen_mandiri(\$nisn){
2	\$query = \$this->db->query("SELECT id_pertanyaan, nisn, jawaban, bukti_pendukung, status FROM asesmen_mandiri WHERE nisn='\$nisn'");
3	return \$query->result_array();
4	}
5	public function get_status_asesmen_mandiri(\$nisn){
6	\$query = \$this->db->query("SELECT nip FROM asesmen_mandiri WHERE nisn='\$nisn' LIMIT 1");
7	return \$query->result_array();
8	}
9	public function update_asesmen_mandiri(\$id_pertanyaan, \$nisn, \$asesor, \$status){
10	\$date = date("Y-m-d");
11	\$query = \$this->db->query("UPDATE asesmen_mandiri SET nip='\$asesor', status='\$status', tanggal='\$date' WHERE id_pertanyaan='\$id_pertanyaan' AND nisn='\$nisn'");
12	}

4.3.2.4 Implementasi Fungsi Mencari Hasil Asesmen Mandiri

Pada proses mencari hasil asesmen mandiri, terdapat satu *controller* yang bertugas untuk hasil asesmen mandiri dan biodata siswa peserta sertifikasi. Proses tersebut ditangani oleh dua *method* yakni *detail_siswa()* yang berfungsi untuk menampilkan detail biodata siswa serta *from()* yang bertugas untuk menampilkan form hasil asesmen mandiri yakni FR-APL-01 dan FR-APL-02. *Controller* tersebut berhubungan dengan 4 (empat) *model* yang bertugas sebagai pemroses data dari *database*. Berikut merupakan hasil implementasi dari kode program. Untuk pencarian siswa dapat dilihat pada sub bab 4.3.2.2 yang menjabarkan tentang melihat daftar peserta sertifikasi beserta pencarian siswa.

Tabel 4.23 Implementasi *Controller* Asesor.php

No	Source Code
1	public function detail_siswa(\$nisn){
2	\$this->load->model('siswa_model');
3	\$data['profil'] = \$this->siswa_model->get_detail_siswa(\$nisn);
4	\$data['status'] = \$this->siswa_model->get_status(\$nisn);
5	\$this->load->model('dokumen_model');
6	\$data['dokumen'] = \$this->dokumen_model->get_dokumen(\$nisn);
7	\$this->load->model('form_model');
8	\$data['form'] = \$this->form_model->get_form(\$nisn);
9	\$data['nisn'] = \$nisn;
10	\$this->load->view('navbar');
11	\$this->load->view('detail_user', \$data);
12	}

Tabel 4.23 Implementasi *Controller* Asesor.php (Lanjutan)

No	Source Code
12	public function form(\$form, \$nisn){
13	\$this->load->model('siswa_model');
14	\$data['siswa'] = \$this->siswa_model->get_detail_siswa(\$nisn);
15	\$this->load->model('unit_model');
16	\$data['unit'] = \$this->unit_model->get_unit();
17	\$this->load->model('dokumen_model');
18	\$data['dokumen'] = \$this->dokumen_model->get_dokumen(\$nisn);
19	\$this->load->model('asesmen_mandiri_model');
20	\$data['info'] = \$this->asesmen_mandiri_model->get_info_asesmen_mandiri(\$nisn);
21	\$status = \$this->siswa_model->get_status(\$nisn);
22	\$skema_siswa='0';
23	foreach(\$status as \$z){
24	\$skema_siswa=\$z['id_skema'];
25	}
26	\$s=0;
27	foreach(\$status as \$d):
28	\$s=\$d['id'];
29	endforeach;
30	if(\$form=='FR-APL-01'&&\$s>=2){
31	\$this->load->view('fr-apl-01', \$data);
32	}
33	else if(\$form=='FR-APL-02'&&\$s>4){
34	\$user= \$this->session->userdata('user');
35	\$data['mandiri'] = \$this->asesmen_mandiri_model->get_asesmen_mandiri(\$nisn);
36	\$data['unit'] = \$this->unit_model->get_unit(\$skema_siswa);
37	foreach(\$data['unit'] as \$d):
38	\$this->load->model('elemen_model');
39	\$data['elemen'][\$d['id_unit']] = \$this->elemen_model->get_elemen(\$d['id_unit']);
40	foreach(\$data['elemen'][\$d['id_unit']] as \$e):
41	\$this->load->model('pertanyaan_model');
42	\$data['pertanyaan'][\$e['id_elemen']] = \$this->pertanyaan_model->get_pertanyaan_by_elemen(\$e['id_elemen']);
43	endforeach;
44	endforeach;
45	\$this->load->view('fr-apl-02', \$data);
46	}
47	else{
48	\$url = base_url()."asesor/detail_siswa/\$nisn";
49	echo "<script>window.close();alert('Dokumen belum tersedia!');window.location.href='\$url';</script>";
50	}
51	}

Pada tabel 4.23 Terdapat *method* yakni *detail_siswa* baris 1 (satu) hingga 11 (sebelas) dan *method form* baris 12 (dua belas) hingga 51 (lima puluh satu). Pada *method detail_siswa* terdapat proses untuk mendapatkan detail siswa, status siswa, dokumen pendukung siswa, serta *form-form* yang ada pada proses asesmen. Sedangkan pada *method form* yang merupakan proses yang menangani pencetakan *form* hasil asesmen. Pada baris 13 (tiga belas) hingga 29 (dua puluh sembilan) merupakan proses untuk mendapatkan data siswa, data unit, dokumen siswa, status asesmen dan info asesmen mandiri siswa. Pada baris 30 (tiga puluh) hingga 51 (lima puluh satu) terdapat proses pengecekan apakah nilai dari parameter *form* bernilai "FR-APL-01" dan *variable s* bernilai lebih dari 2 (dua). Jika benar, maka halaman akan menampilkan *view fr-apl-01* yang berisi biodata siswa beserta nama dokumen yang dilampirkan. Namun, apabila nilai dari parameter *form* adalah "FR-APL-02" dan *variable s* bernilai lebih dari 4 (empat), maka sistem akan menampilkan *view fr-apl-02* lengkap beserta hasil asesmen mandiri siswa sesuai dengan skema yang ada dan hasil penilaian asesor. Namun, apabila parameter *form* tidak bernilai keduanya, maka halaman akan menampilkan peringatan "Dokumen belum tersedia".

Pada tabel 4.24 merupakan implementasi *model siswa_model* yang mempunyai *method get_status* baris 1 (satu) hingga 4 (empat) dan *get_detail_siswa* baris 5 (lima) hingga 8 (delapan). *Method get_status* merupakan *method* yang berfungsi untuk mengambil data status asesmen siswa dari *database* dan *method get_detail_siswa* merupakan *method* yang berfungsi untuk menampilkan seluruh biodata siswa dengan lengkap.

Tabel 4.24 Implementasi *Model siswa_model.php*

No	Source Code
1	<code>public function get_status(\$nisn){</code>
2	<code> \$query = \$this->db->query("SELECT S.id, S.status,</code>
	<code> u.id_skema FROM siswa U LEFT JOIN status s ON u.status=s.id</code>
	<code> WHERE u.username = '\$nisn');</code>
3	<code> return \$query->result_array();</code>
4	<code>}</code>
5	<code>public function get_detail_siswa(\$username){</code>
6	<code> \$query = \$this->db->query("SELECT s.nisn, u.password,</code>
	<code> u.nama, u.email, u.alamat, u.kode_pos, u.nomor_handphone,</code>
	<code> k.jenis_kartu_identitas, u.nomor_kartu_identitas, P.name</code>
	<code> as provinsi, R.name as Kota, u.tanggal_lahir,</code>
	<code> u.jenis_kelamin, u.password, j.nama as skema, s.kelas FROM</code>
	<code> user u, KARTU_IDENTITAS K, PROVINCES P, REGENCIES R, skema</code>
	<code> J, siswa S WHERE u.username='\$username' AND u.username =</code>
	<code> s.username AND u.id_kartu_identitas = K.id_kartu_identitas</code>
	<code> AND u.id_kota_lahir = R.id AND R.province_id =P.id AND</code>
	<code> s.id_skema = j.id_skema ");</code>
7	<code> return \$query->result_array();</code>
8	<code>}</code>

Pada tabel 4.25 merupakan implementasi *model* dokumen_model dan mempunyai *method* get_dokumen. *Method* tersebut berfungsi untuk menampilkan seluruh dokumen yang telah diunggah siswa pada sistem berdasarkan nisan siswa yang dipilih.

Tabel 4.25 Implementasi Model dokumen_model.php

No	Source Code
1	public function get_dokumen(\$nisan){
2	\$query = \$this->db->query("SELECT * FROM dokumen WHERE nisan='\$nisan'");
3	return \$query->result_array();
4	}

Pada tabel 4.26 merupakan implementasi *model* dokumen_model dan mempunyai *method* get_form. *Method* tersebut berfungsi untuk mengambil seluruh data *form* dari *database*.

Tabel 4.26 Implementasi Model dokumen_model.php

No	Source Code
1	public function get_form(){
2	\$query = \$this->db->query("SELECT * FROM form");
3	return \$query->result_array();
4	}

Pada tabel 4.27 merupakan implementasi *model* asesmen_mandiri_model dan mempunyai *method* get_info_asesmen_mandiri. *Method* tersebut berfungsi untuk mendapatkan data nama dan tanggal asesmen mandiri siswa sebagai informasi pada dokumen FR-APL-02.

Tabel 4.27 Implementasi Model asesmen_mandiri_model.php

No	Source Code
1	public function get_info_asesmen_mandiri(\$nisan){
2	\$query = \$this->db->query("SELECT u.nama, s.tanggal FROM user u left join asesmen_mandiri s on u.username = s.nip WHERE nisan='\$nisan' LIMIT 1");
3	return \$query->result_array();
4	}

4.3.3 Implementasi *Database*

Pada proses implementasi *database* merupakan implementasi dari hasil perancangan *physical data model* yang telah dibuat sebelumnya. implementasi yang dilakukan antara lain pembuatan *database*, tabel, dan kolom serta relasi masing-masing tabel. Perintah yang digunakan dalam implementasi ini adalah *data definition language* (DDL) yang digunakan untuk proses manajemen objek *database*.

4.3.3.1 Implementasi Pembuatan *Database*

Tabel 4.28 Implementasi Pembuatan *Database*

No	Source Code
1	CREATE DATABASE db_lspsmkn4;

Pada tabel 4.28 terdapat perintah untuk implementasi pembuatan database yang bernama db_lspsmkn4. Perintah *query* tersebut dijalankan pada awal implementasi sebelum masuk menuju implementasi-implementasi tabel di dalamnya.

4.3.3.2 Implementasi Pembuatan Tabel *Provinces*

Tabel 4.29 Implementasi Pembuatan Tabel *Provinces*

No	Source Code
1	CREATE TABLE provinces (id char(2) NOT NULL, name varchar(255) NOT NULL, PRIMARY KEY (id));

Pada tabel 4.29 terdapat perintah untuk implementasi pembuatan tabel *provinces* yang berfungsi untuk menyimpan data provinsi. Pada *query* tersebut dapat diketahui bahwa kolom id dengan tipe data char sejumlah 2 (dua) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain. Selain itu terdapat kolom *name* untuk menyimpan data nama provinsi dengan panjang maksimal 255 (dua ratus lima puluh lima)

4.3.3.3 Implementasi Pembuatan Tabel *Level*

Tabel 4.30 Implementasi Pembuatan Tabel *Level*

No	Source Code
1	CREATE TABLE level (id_level int(3) NOT NULL AUTO_INCREMENT, level varchar(15) NOT NULL, PRIMARY KEY (id_level));

Pada tabel 4.30 terdapat perintah untuk implementasi pembuatan tabel *level* yang berfungsi untuk menyimpan data *level* asesmen. Pada *query* tersebut dapat diketahui bahwa kolom id dengan tipe data integer sejumlah 3 (tiga) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada

pertambahan data. Selain itu terdapat kolom *level* untuk menyimpan data nama *level* dengan panjang maksimal 15 (lima belas) karakter.

4.3.3.4 Implementasi Pembuatan Tabel Status

Tabel 4.31 Implementasi Pembuatan Tabel Status

No	Source Code
1	CREATE TABLE status (id int(3) NOT NULL AUTO_INCREMENT, status varchar(40) NOT NULL, PRIMARY KEY (id));

Pada tabel 4.31 terdapat perintah untuk implementasi pembuatan tabel status yang berfungsi untuk menyimpan data status proses asesmen. Pada *query* tersebut dapat diketahui bahwa kolom id dengan tipe data integer sejumlah 3 (tiga) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada pertambahan data. Selain itu terdapat kolom status untuk menyimpan data nama status dengan panjang maksimal 40 (empat puluh) karakter.

4.3.3.5 Implementasi Pembuatan Tabel Form

Tabel 4.32 Implementasi Pembuatan Tabel Form

No	Source Code
1	CREATE TABLE form (id_form int(3) NOT NULL AUTO_INCREMENT, kode varchar(15) NOT NULL, nama varchar(50) NOT NULL, PRIMARY KEY (id_form));

Pada tabel 4.32 terdapat perintah untuk implementasi pembuatan tabel form yang berfungsi untuk menyimpan data form asesmen. Pada *query* tersebut dapat diketahui bahwa kolom id_form dengan tipe data integer sejumlah 3 (tiga) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada pertambahan data. Selain itu terdapat kolom kode untuk menyimpan data kode *form* dengan panjang maksimal 15 (lima belas) karakter dan kolom nama dengan panjang maksimal (50) karakter untuk menyimpan data nama *form*.

4.3.3.6 Implementasi Pembuatan Tabel Program Keahlian

Tabel 4.33 Implementasi Pembuatan Tabel Program Keahlian

No	Source Code
1	CREATE TABLE program_keahlian (id_program_keahlian int(3) NOT NULL AUTO_INCREMENT, nama_program_keahlian varchar(50) NOT NULL, PRIMARY KEY (id_program_keahlian));

Pada tabel 4.33 terdapat perintah untuk implementasi pembuatan tabel program keahlian yang berfungsi untuk menyimpan data program keahlian peserta asesmen. Pada *query* tersebut dapat diketahui bahwa kolom id_program_keahlian dengan tipe data integer sejumlah 3 (tiga) bertindak sebagai

primary key yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada penambahan data. Selain itu terdapat kolom *nama_program_keahlian* untuk menyimpan data jenis program keahlian dengan panjang maksimal 50 (lima puluh) karakter.

4.3.3.7 Implementasi Pembuatan Tabel Kartu Identitas

Tabel 4.34 Implementasi Pembuatan Tabel Kartu Identitas

No	Source Code
1	CREATE TABLE kartu_identitas (id_kartu_identitas int(3) NOT NULL AUTO_INCREMENT, jenis_kartu_identitas varchar(30) NOT NULL, PRIMARY KEY (id_kartu_identitas));

Pada tabel 4.34 terdapat perintah untuk implementasi pembuatan tabel kartu identitas yang berfungsi untuk menyimpan data kartu identitas peserta asesmen. Pada *query* tersebut dapat diketahui bahwa kolom *id_kartu_identitas* dengan tipe data integer sejumlah 3 (tiga) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada penambahan data. Selain itu terdapat kolom *jenis_kartu_identitas* untuk menyimpan data kartu identitas peserta sertifikasi dengan panjang maksimal 30 (tiga puluh) karakter.

4.3.3.8 Implementasi Pembuatan Tabel Regencies

Tabel 4.35 Implementasi Pembuatan Tabel Regencies

No	Source Code
1	CREATE TABLE regencies (id char(4) NOT NULL, province_id char(2) NOT NULL, name varchar(255) NOT NULL, PRIMARY KEY (id), FOREIGN KEY (province_id) REFERENCES provinces (id));

Pada tabel 4.35 terdapat perintah untuk implementasi pembuatan tabel *regencies* yang berfungsi untuk menyimpan Kabupaten atau Kota di Indonesia. Pada *query* tersebut dapat diketahui bahwa kolom *id* dengan tipe data char sejumlah 4 (empat) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain. Selain itu terdapat kolom *name* untuk menyimpan data nama Kabupaten atau Kota dengan panjang maksimal 255 (dua ratus lima puluh lima) karakter. Sebelum kolom *name*, terdapat kolom *province_id* dengan tipe data char sejumlah 2 (dua) yang menyimpan data id provinsi. Data dalam kolom tersebut merujuk pada tabel *provinces* kolom *id*.

4.3.3.9 Implementasi Pembuatan Tabel Skema

Tabel 4.36 Implementasi Pembuatan Tabel Skema

No	Source Code
1	<pre>CREATE TABLE skema (id_skema int(3) NOT NULL AUTO_INCREMENT, id_program_keahlian int(3) NOT NULL, id_level int(3) NOT NULL, kode varchar(10) NOT NULL, kategori varchar(20) NOT NULL, nama varchar(40) NOT NULL, PRIMARY KEY (id_skema), FOREIGN KEY (id_program_keahlian) REFERENCES program_keahlian (id_program_keahlian), FOREIGN KEY (id_level) REFERENCES level (id_level));</pre>

Pada tabel 4.36 terdapat perintah untuk implementasi pembuatan tabel skema yang berfungsi untuk menyimpan data skema asesmen. Pada *query* tersebut dapat diketahui bahwa kolom *id_skema* dengan tipe data integer sejumlah 3 (tiga) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada penambahan data. Selain itu terdapat kolom *kode* untuk menyimpan data kode skema dengan panjang maksimal 10 (sepuluh) karakter, kolom *kategori* untuk menyimpan data kategori skema dengan panjang maksimal 20 (dua puluh) karakter, kolom *nama* untuk menyimpan data nama skema dengan panjang maksimal 40 (empat puluh) karakter. Terdapat pula kolom *id_program_keahlian* bertipe data integer dengan panjang 3 (tiga) yang merujuk pada tabel *program_keahlian* kolom *id_program keahlian* dan terdapat kolom *id_level* yang merujuk pada tabel *level* kolom *id_level*.

4.3.3.10 Implementasi Pembuatan Tabel Unit

Tabel 4.37 Implementasi Pembuatan Tabel Unit

No	Source Code
1	<pre>CREATE TABLE unit (id_unit int(3) NOT NULL AUTO_INCREMENT, id_skema int(3) NOT NULL, kode varchar(20) NOT NULL, judul varchar(100) NOT NULL, jenis varchar(15) NOT NULL, PRIMARY KEY (id_unit), FOREIGN KEY (id_skema) REFERENCES skema (id_skema));</pre>

Pada tabel 4.37 terdapat perintah untuk implementasi pembuatan tabel unit yang berfungsi untuk menyimpan data unit asesmen. Pada *query* tersebut dapat diketahui bahwa kolom *id_unit* dengan tipe data integer sejumlah 3 (tiga) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada penambahan data. Selain itu terdapat kolom *kode* untuk menyimpan data kode unit dengan panjang maksimal 20 (dua puluh) karakter, kolom *judul* untuk menyimpan data judul unit dengan panjang maksimal 100 (seratus) karakter, kolom *jenis* untuk menyimpan data jenis unit dengan panjang maksimal 15 (lima belas) karakter. Terdapat pula kolom *id_skema* bertipe data integer dengan panjang 3 (tiga) yang merujuk pada tabel *skema* kolom *id_skema*.

4.3.3.11 Implementasi Pembuatan Tabel Elemen

Tabel 4.38 Implementasi Pembuatan Tabel Elemen

No	Source Code
1	<pre>CREATE TABLE elemen (id_elemen int(3) NOT NULL AUTO_INCREMENT, id_unit int(3) NOT NULL, nomor varchar(3) NOT NULL, elemen varchar(1000) NOT NULL, PRIMARY KEY (id_elemen), FOREIGN KEY (id_unit) REFERENCES unit (id_unit));</pre>

Pada tabel 4.38 terdapat perintah untuk implementasi pembuatan tabel elemen yang berfungsi untuk menyimpan data elemen asesmen. Pada *query* tersebut dapat diketahui bahwa kolom *id_elemen* dengan tipe data integer sejumlah 3 (tiga) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada pertambahan data. Selain itu terdapat kolom nomor untuk menyimpan data urutan elemen dengan panjang maksimal 3 (tiga) karakter, kolom elemen untuk menyimpan data detail elemen dengan panjang maksimal 1000 (seribu) karakter. Terdapat pula kolom *id_unit* bertipe data integer dengan panjang 3 (tiga) yang merujuk pada tabel unit kolom *id_unit*.

4.3.3.12 Implementasi Pembuatan Tabel Pertanyaan

Tabel 4.39 Implementasi Pembuatan Tabel Pertanyaan

No	Source Code
1	<pre>CREATE TABLE pertanyaan (id_pertanyaan int(3) NOT NULL AUTO_INCREMENT, id_elemen int(3) NOT NULL, id_form int(3) NOT NULL, pertanyaan varchar(2000) NOT NULL, nomor varchar(3) NOT NULL, status int(1) NOT NULL, PRIMARY KEY (id_pertanyaan), FOREIGN KEY (id_elemen) REFERENCES elemen (id_elemen), FOREIGN KEY (id_form) REFERENCES form (id_form));</pre>

Pada tabel 4.39 terdapat perintah untuk implementasi pembuatan tabel pertanyaan yang berfungsi untuk menyimpan data pertanyaan asesmen. Pada *query* tersebut dapat diketahui bahwa kolom *id_pertanyaan* dengan tipe data integer sejumlah 3 (tiga) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada pertambahan data. Selain itu terdapat kolom nomor untuk menyimpan data urutan pertanyaan dengan panjang maksimal 3 (tiga) karakter, kolom status untuk menyimpan data status pertanyaan apakah telah diverifikasi oleh manajemen atau belum dengan panjang maksimal 1 (satu) karakter, kolom pertanyaan untuk menyimpan butir pertanyaan dengan Panjang maksimal 2000 (dua ribu) karakter. Terdapat pula kolom *id_elemen* bertipe data integer dengan panjang 3 (tiga) yang merujuk pada tabel elemen kolom *id_elemen*, dan kolom *id_form* bertipe data integer dengan panjang 3 (tiga) yang merujuk pada tabel form kolom *id_form*.

4.3.3.13 Implementasi Pembuatan Tabel *User*

Tabel 4.40 Implementasi Pembuatan Tabel *User*

No	Source Code
1	<pre>CREATE TABLE user (username varchar(25) NOT NULL, password varchar(255) NOT NULL, nama` varchar(40) NOT NULL, email varchar(100) NOT NULL, alamat varchar(255) NOT NULL, kode_pos varchar(6) NOT NULL, nomor_handphone varchar(20) NOT NULL, id_kartu_identitas int(3) NOT NULL, nomor_kartu_identitas varchar(30) NOT NULL, id_Kota_lahir char(4) NOT NULL, tanggal_lahir date NOT NULL, jenis_kelamin varchar(20) NOT NULL, PRIMARY KEY (username), FOREIGN KEY (id_kartu_identitas) REFERENCES kartu_identitas (id_kartu_identitas), FOREIGN KEY (id_Kota_lahir) REFERENCES `regencies` (`id`));</pre>

Pada tabel 4.40 terdapat perintah untuk implementasi pembuatan tabel *user* yang berfungsi untuk menyimpan biodata pengguna aplikasi baik asesor maupun siswa. Pada *query* tersebut dapat diketahui bahwa kolom *username* dengan tipe data *varchar* sejumlah 25 (dua puluh lima) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada penambahan data. Selain itu terdapat berbagai kolom untuk menyimpan biodata, diantaranya kolom *password*, *nama*, *email*, *alamat*, *kode_pos*, *nomor-handphone*, *nomor_kartu_identitas*, *tanggal_lahir*, dan *jenis_kelamin*. Tabel *user* juga terhubung kepada tabel *kartu_identitas* dan tabel *regencies*. Tabel *kartu_identitas* terhubung melalui kolom *id_kartu_identitas*, sedangkan tabel *regencies* terhubung melalui kolom *id_Kota_lahir* yang menunjukkan Kabupaten atau Kota tempat lahir *user*.

4.3.3.14 Implementasi Pembuatan Tabel Asesor

Tabel 4.41 Implementasi Pembuatan Tabel Asesor

No	Source Code
1	<pre>CREATE TABLE asesor (nip varchar(30) NOT NULL, role varchar(15) NOT NULL, username varchar(25) NOT NULL, PRIMARY KEY (nip), FOREIGN KEY(username) REFERENCES user (username));</pre>

Pada tabel 4.41 terdapat perintah untuk implementasi pembuatan tabel *asesor* yang berfungsi untuk menyimpan data asesor. Pada *query* tersebut dapat diketahui bahwa kolom *nip* dengan tipe data *varchar* sejumlah 30 (tiga puluh) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain. Selain itu terdapat kolom *role* untuk menyimpan data jenis asesor yakni asesor murni dan manajemen dengan panjang maksimal 15 (lima belas) karakter. Terdapat pula kolom *username* bertipe data *varchar* dengan panjang 25 (dua puluh lima) yang merujuk pada tabel *user* kolom *username*.

4.3.3.15 Implementasi Pembuatan Tabel Siswa

Tabel 4.42 Implementasi Pembuatan Tabel Siswa

No	Source Code
1	<pre>CREATE TABLE siswa (nisan varchar(30) NOT NULL, kelas varchar(20) NOT NULL, status int(3) NOT NULL, username varchar(25) NOT NULL, id_skema int(3) NOT NULL, PRIMARY KEY (nisan), FOREIGN KEY (username) REFERENCES user (username), FOREIGN KEY (status) REFERENCES status (id), FOREIGN KEY (id_skema) REFERENCES skema (id_skema));</pre>

Pada tabel 4.42 terdapat perintah untuk implementasi pembuatan tabel siswa yang berfungsi untuk menyimpan data siswa peserta sertifikasi. Pada *query* tersebut dapat diketahui bahwa kolom nisan dengan tipe data varchar sejumlah 30 (tiga puluh) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain. Selain itu terdapat kolom kelas untuk menyimpan data jenis kelas siswa dengan panjang maksimal 20 (dua puluh) karakter. Terdapat pula kolom status bertipe data integer dengan panjang 3 (tiga) yang merujuk pada tabel status kolom id, kolom *username* bertipe data varchar dengan Panjang 25 (dua puluh lima) yang merujuk pada tabel *user* kolom *username*, dan kolom id_skema bertipe data integer dengan panjang 3 (tiga) yang merujuk pada tabel skema kolom id_skema.

4.3.3.16 Implementasi Pembuatan Tabel Dokumen

Tabel 4.43 Implementasi Pembuatan Tabel Dokumen

No	Source Code
1	<pre>CREATE TABLE dokumen (id_dokumen int(25) NOT NULL AUTO_INCREMENT, nisan varchar(30) NOT NULL, jenis_dokumen varchar(20) NOT NULL, url_dokumen varchar(255) NOT NULL, PRIMARY KEY (id_dokumen), FOREIGN KEY (nisan) REFERENCES siswa (nisan));</pre>

Pada tabel 4.43 terdapat perintah untuk implementasi pembuatan tabel dokumen yang berfungsi untuk menyimpan data dokumen pendukung yang diunggah siswa. Pada *query* tersebut dapat diketahui bahwa kolom id_dokumen dengan tipe data integer sejumlah 25 (dua puluh lima) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada pertambahan data. Selain itu terdapat kolom jenis_dokumen untuk menyimpan data jenis dokumen yang diunggah siswa dengan panjang maksimal 20 (dua puluh) karakter, kolom url_dokumen untuk menyimpan data tautan menuju dokumen yang telah diunggah siswa dengan panjang maksimal 255 (dua ratus lima puluh lima) karakter. Terdapat pula kolom nisan bertipe data varchar dengan panjang 25 (dua puluh lima) yang merujuk pada tabel siswa kolom nisan.

4.3.3.17 Implementasi Pembuatan Tabel Asesmen Mandiri

Tabel 4.44 Implementasi Pembuatan Tabel Asesmen Mandiri

No	Source Code
1	<pre>CREATE TABLE asesmen_mandiri (id_penilaian int(100) NOT NULL AUTO_INCREMENT, id_form int(3) NOT NULL, id_pertanyaan int(3) NOT NULL, nisan varchar(30) NOT NULL, nip varchar(30) DEFAULT NULL, jawaban varchar(5) NOT NULL, bukti_pendukung varchar(250) DEFAULT NULL, status varchar(10) DEFAULT NULL, tanggal date NOT NULL, PRIMARY KEY (id_penilaian), FOREIGN KEY (id_pertanyaan) REFERENCES pertanyaan (id_pertanyaan), FOREIGN KEY (id_form) REFERENCES form (id_form), FOREIGN KEY (nip) REFERENCES asesor (nip), FOREIGN KEY (nisan) REFERENCES siswa (nisan));</pre>

Pada tabel 4.44 terdapat perintah untuk implementasi pembuatan tabel asesmen mandiri yang berfungsi untuk data hasil asesmen mandiri siswa maupun hasil penilaian asesmen mandiri oleh asesor. Pada *query* tersebut dapat diketahui bahwa kolom *id_penilaian* dengan tipe data integer sejumlah 100 (seratus) bertindak sebagai *primary key* yang digunakan untuk membedakan data satu dengan yang lain serta bersifat *AUTO_INCREMENT* yakni bertambah satu setiap ada penambahan data. Selain itu terdapat beberapa kolom yang menunjang dalam penyimpanan asesmen mandiri yakni jawaban, bukti_pendukung, status, dan tanggal. Tabel *user* juga terhubung kepada tabel form, pertanyaan, siswa, dan tabel asesor. Tabel form terhubung melalui kolom *id_form*, tabel pertanyaan terhubung melalui kolom *id_pertanyaan*, tabel siswa terhubung melalui kolom *nisan* sedangkan tabel asesor terhubung melalui kolom *nip* yang menunjukkan siapa asesor yang telah menilai butir hasil asesmen mandiri tersebut.

4.4 Pengujian dan Analisis

Pengujian dan analisis merupakan salah satu proses dalam tahapan *construction* dari SDLC *waterfall*. Pada proses ini digunakan untuk melihat sejauh mana aplikasi yang telah dikembangkan telah dapat digunakan oleh pengguna. Proses pengujian menggunakan dua metodi yakni *black-box testing* dan *user acceptance testing* (UAT)

4.4.1 Pengujian *Black-box*

Pada pengujian *Black-box* penelitian ini menguji 4 (empat) fungsional utama dari aplikasi yang dikembangkan, yakni melakukan proses asesmen mandiri, melihat daftar pemohon sertifikasi, menilai asesmen mandiri siswa, dan mencari hasil asesmen mandiri. Skenario pengujian tersebut peneliti menebak kemungkinan *error* yang akan terjadi (*error guessing*) pada 4 (empat) fungsional utama aplikasi dan mendefinisikan keluaran yang diharapkan. Kemudian, kemungkinan *error* tersebut diujikan pada aplikasi secara langsung. Apabila keluaran program sesuai yang diharapkan maka hasil yang diharapkan adalah "valid" dan menginformasikan bahwa pengujian telah berhasil dilaksanakan dan

menginformasikan bahwa kemungkinan *error* telah dapat tertangani dengan baik serta fungsional yang diujikan telah layak digunakan oleh pengguna.

4.4.1.1 Pengujian *Black-box* Melakukan Proses Asesmen Mandiri

Pada gambar 4.35 merupakan *form* pengisian asesmen mandiri. Pada *form* tersebut siswa diwajibkan mengisi penilaian anantara K dan BK namun tidak diwajibkan untuk mengisi bukti pendukung.

Dari gambar 4.35 diperoleh kemungkinan kesalahan (*error guessing*) yang dapat dilihat pada tabel 4.46. Pada tabel tersebut dijabarkan kemungkinan *error* yang akan terjadi pada saat perangkat lunak digunakan.

Beranda
Berkas
Asesmen Mandiri
Profil
Logout

ASESMEN MANDIRI

Kode Unit Kompetensi	TIK-JK01.006.01		
Judul Unit Kompetensi	Menerapkan prosedur kesehatan, keselamatan, dan keamanan kerja		
Elemen Kompetensi	1. Menetapkan hal-hal yang berhubungan dengan kesehatan dan keselamatan kerja di lingkungan kerja		
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu:		Penilaian K BK	Bukti Pendukung
1.1. Apakah anda dapat menetapkan pengawasan kesehatan dan keselamatan kerja sesuai undang-undang kesehatan dan keselamatan kerja		☐	☐
1.2. Apakah anda dapat meninjau dan melaporkan prosedur kesehatan dan keselamatan kerja di lingkungan kerja untuk memperbaiki masalah yang ada pada pengawasan		☐	☐
1.3. Apakah anda dapat melaksanakan prosedur kesehatan dan keselamatan kerja untuk menjamin keamanan di lingkungan kerja		☐	☐
Elemen Kompetensi	2. Mendokumentasikan dan menyebarkan syarat-syarat kesehatan dan keselamatan kerja		
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu:		Penilaian K BK	Bukti Pendukung
2.1. Apakah anda dapat mengumpulkan informasi yang berhubungan dengan peraturan kesehatan dan keselamatan kerja dan syarat-syaratnya		☐	☐
2.2. Apakah anda dapat menetapkan dan mendokumentasikan peraturan-peraturan kesehatan dan keselamatan yang berkaitan dengan teknologi informasi di area klien		☐	☐
2.3. Apakah anda dapat mengajukan dokumen pada pengawas untuk diverifikasi		☐	☐
2.4. Apakah anda dapat menyebarkan dokumen kesehatan dan keselamatan kerja pada semua pos kerja		☐	☐
2.5. Apakah anda dapat memperbaharui dan mendiseminasi dokumen-dokumen kesehatan dan keselamatan kerja yang berhubungan dengan teknologi informasi		☐	☐
Elemen Kompetensi	3. Menyediakan saran-saran ergonomis dasar		
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu:		Penilaian K BK	Bukti Pendukung
3.1. Apakah anda dapat menilai syarat-syarat ergonomis dari klien		☐	☐
3.2. Apakah anda dapat menyediakan saran untuk klien berdasarkan persyaratan dari vendor, kebijakan ditempat kerja, serta informasi kesehatan dan keselamatan kerja terbaru		☐	☐
3.3. Apakah anda dapat mendokumentasikan dan memberikan saran pada klien dan pengawas		☐	☐

Selanjutnya

Asesmen Mandiri LSP SMK Negeri 4 Malang

PERINGATAN : Pastikan mengisi dengan benar, karena masing-masing halaman yang telah diisi tidak dapat diulangi!

Pada bagian ini, anda diminta untuk menilai diri sendiri terhadap unit (unit-unit) kompetensi yang akan di-ases.

1. Pelajari seluruh standar Kriteria Unjuk Kerja (KUK), batasan variabel, panduan penilaian dan aspek kritis serta yakinkan bahwa anda sudah benar-benar memahami seluruh isinya.

2. Laksanakan penilaian mandiri dengan mempelajari dan menilai kemampuan yang anda miliki secara obyektif terhadap seluruh daftar pertanyaan yang ada, serta tentukan apakah sudah kompeten (K) atau belum kompeten (BK) serta tuliskan bukti-bukti pendukung yang anda anggap relevan terhadap setiap elemen/KUK unit kompetensi.

1 2 3 4 5 6 7 8

Keterangan Asesmen Mandiri

Gambar 4.35 Contoh *Form* Melakukan Proses Asesmen Mandiri

Berikut merupakan tabel 4.45 yang berisi hasil dari pengujian *error guessing* melakukan proses asesmen mandiri.

Tabel 4.45 Error Guessing Melakukan Proses Asesmen Mandiri

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Status
1	Pada proses penilaian, terdapat <i>field</i> yang dibiarkan kosong	<i>Error</i>	Sistem menampilkan peringatan pada <i>field</i> penilaian yang masih kosong dan tidak dapat dilanjutkan	Valid
2	<i>Field</i> Bukti pendukung asesmen mandiri dibiarkan kosong	Tidak <i>Error</i>	Sistem tetap dapat menyimpan data meskipun <i>field</i> bukti pendukung dibiarkan kosong	Valid

Dari hasil pengujian pada proses melakukan proses asesmen mandiri yang ditunjukkan tabel 4.45 menunjukkan bahwa seluruh pengujian untuk mendapatkan *error* telah dijalankan. Dari hasil tersebut, didapatkan bahwa seluruh pengujian telah dijalankan dan mendapatkan status valid atau sesuai dengan hasil yang diharapkan.

4.4.1.2 Pengujian *Black-box* Melihat Daftar Pemohon Sertifikasi

Pada gambar 4.36 merupakan halaman untuk melihat daftar pemohon sertifikasi. Pada halaman tersebut asesor dapat melihat daftar pemohon sertifikasi sesuai dengan status maupun dapat melakukan pencarian dan pengurutan siswa.

NO	NISN IT	NAMA IT	NOMOR HANDPHONE	PROGRAM KEAHLIAN IT	KELAS IT	STATUS IT
1	123456	Arif Rahmadani Vinanda	082233682181	Rekayasa Perangkat Lunak	A	Slap dilakukan ases
2	156150600111024	Fakhrizal Arif Pratidina	081233444050	Rekayasa Perangkat Lunak	A	Belum mengupload berkas
3	654321	Fahmi Aquinas	082233682181	Rekayasa Perangkat Lunak	A	Pengisian asesmen mandiri

Gambar 4.36 Contoh Form Melihat Daftar Pemohon Sertifikasi

Dari gambar 4.36 diperoleh kemungkinan kesalahan (*error guessing*) yang dapat dilihat pada tabel 4.46. Pada tabel tersebut dijabarkan kemungkinan *error* yang akan terjadi pada saat perangkat lunak digunakan.

Tabel 4.46 Error Guessing Melihat Daftar Pemohon Sertifikasi

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Status
1	Pada kolom pencarian diinputkan karakter huruf sebanyak-banyaknya	Tidak <i>Error</i>	Sistem tetap melakukan pencarian meskipun kata kunci yang diinputkan berjumlah banyak	Valid
2	Pada kolom pencarian dimasukkan karakter kosong	Tidak <i>Error</i>	Sistem menampilkan seluruh data dengan status yang dimaksud apabila pada kolom pencarian dimasukkan karakter kosong	Valid
3	Saat menekan tombol <i>reset</i> maka sistem akan menampilkan seluruh pemohon sertifikasi	Tidak <i>Error</i>	Sistem menampilkan seluruh data pemohon sertifikasi mulai dari status awal hingga akhir.	Valid

Dari hasil pengujian pada proses melihat daftar pemohon sertifikasi yang ditunjukkan tabel 4.46 menunjukkan bahwa seluruh pengujian untuk mendapatkan *error* telah dijalankan. Dari hasil tersebut, didapatkan bahwa seluruh pengujian telah dijalankan dan mendapatkan status valid atau sesuai dengan hasil yang diharapkan.

4.4.1.3 Pengujian *Black-box* Menilai Asesmen Mandiri Siswa

Pada gambar 4.37 merupakan halaman untuk asesor menilai hasil asesmen mandiri siswa. Pada halaman tersebut asesor dapat melakukan penilaian pada kolom V, A, T, atau M. Asesor melakukan penilaian berdasarkan hasil asesmen mandiri siswa yang telah selesai. Asesor tidak akan bisa menilai siswa apabila siswa belum terverifikasi.

1. ASESMEN MANDIRI

Kode Unit Kompetensi	TIK-JK01.006.01									
Judul Unit Kompetensi	Menerapkan prosedur kesehatan, keselamatan, dan keamanan kerja									
Elemen Kompetensi	1. Menetapkan hal-hal yang berhubungan dengan kesehatan dan keselamatan kerja di lingkungan kerja									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu:	Penilaian		Bukti Pendukung	Diisi Asesor					Check All	
	K	BK		V	A	T	M			
1.1. Apakah anda dapat menetapkan pengawasan kesehatan dan keselamatan kerja sesuai undang-undang kesehatan dan keselamatan kerja	✓									
1.2. Apakah anda dapat meninjau dan melaporkan prosedur kesehatan dan keselamatan kerja di lingkungan kerja untuk memperbaiki masalah yang ada pada pengawas	✓									
1.3. Apakah anda dapat melaksanakan prosedur kesehatan dan keselamatan kerja untuk menjamin keamanan di lingkungan kerja	✓									
Elemen Kompetensi	2. Mendokumentasikan dan menyebarkan syarat-syarat kesehatan dan keselamatan kerja									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu:	Penilaian		Bukti Pendukung	Diisi Asesor					Check All	
	K	BK		V	A	T	M			
2.1. Apakah anda dapat mengumpulkan informasi yang berhubungan dengan peraturan kesehatan dan keselamatan kerja dan syarat-syaratnya	✓									
2.2. Apakah anda dapat menetapkan dan mendokumentasikan peraturan-peraturan kesehatan dan keselamatan yang berkaitan dengan teknologi informasi di area klien	✓									
2.3. Apakah anda dapat mengajukan dokumen pada pengawas untuk diverifikasi	✓									
2.4. Apakah anda dapat menyebarkan dokumen kesehatan dan keselamatan kerja pada semua pos kerja	✓									

Pada bagian ini, anda diminta untuk menilai diri sendiri terhadap unit (unit-unit) kompetensi yang akan di-ases.

1. Pelajari seluruh standar Kriteria Unjuk Kerja (KUK), batasan variabel, panduan penilaian dan aspek kritis serta yakinkan bahwa anda sudah benar-benar memahami seluruh isinya.
2. Laksanakan penilaian mandiri dengan mempelajari dan menilai kemampuan yang anda miliki secara obyektif terhadap seluruh daftar pertanyaan yang ada, serta tentukan apakah sudah kompeten (**K**) atau belum kompeten (**BK**) serta tuliskan bukti-bukti pendukung yang anda anggap relevan terhadap setiap elemen/KUK unit kompetensi.

1
2
3
4
5
6
7
8

Ketentuan Asesmen Mandiri

Gambar 4.37 Contoh Form Menilai Asesmen Mandiri Siswa

Dari gambar 4.37 diperoleh kemungkinan kesalahan (*error guessing*) yang dapat dilihat pada tabel 4.47. Pada tabel tersebut dijabarkan kemungkinan *error* yang akan terjadi pada saat perangkat lunak digunakan.

Tabel 4.47 Error Guessing Menilai Asesmen Mandiri Siswa

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Status
1	Pada proses penilaian, terdapat <i>field</i> penilaian yang dibiarkan kosong	<i>Tidak Error</i>	Sistem dapat memasukkan penilaian asesor walau terdapat <i>checkbox</i> yang kosong	Valid
2	Membuka halaman secara manual walau siswa belum menyelesaikan proses asesmen mandiri	<i>Error</i>	Sistem menampilkan peringatan bahwa asesor belum dapat mengakses halaman tersebut dan halaman akan pindah menuju ke halaman daftar siswa	Valid

Dari hasil pengujian pada proses menilai asesmen mandiri siswa yang ditunjukkan tabel 4.47 menunjukkan bahwa seluruh pengujian untuk mendapatkan *error* telah dijalankan. Dari hasil tersebut, didapatkan bahwa seluruh pengujian telah dijalankan dan mendapatkan status valid atau sesuai dengan hasil yang diharapkan.

4.4.1.4 Pengujian *Black-box* Mencari Hasil Asesmen Mandiri

Pada gambar 4.37 merupakan halaman untuk asesor mencetak hasil asesmen mandiri. Proses yang dilakukan sebelumnya ialah mencari siswa pada halaman daftar siswa. Apabila data siswa terlalu banyak, maka asesor dapat memanfaatkan fitur pencarian dan pengurutan siswa pada halaman tersebut. Setelah siswa yang dimaksud untuk dicari dokumen asesmen mandiri terpilih, maka sistem akan memunculkan halaman detail siswa. Pada halaman tersebut asesor dapat menemukan tautan menuju dokumen asesmen mandiri yang dimaksud baik FR-APL-01 yang berisi biodata siswa dan FR-APL-02 yang berisi hasil asesmen mandiri dan telah dinilai oleh asesor.

The screenshot shows a web application interface for a university. The top navigation bar includes links for Beranda, Daftar Siswa, Profil, Manajemen, and Logout. The main content area is titled 'DETAIL SISWA' and displays the following information:

- NISN:** 15615060111012
- Nama:** Arif Rahmadani Vinanda
- Email:** arifrahmadanivinanda@gmail.com
- Alamat:** JL. Ronggowuni 132, Tumpang
- Kode Pos:** 65156
- Nomor Handphone:** 082233682181
- Jenis Kartu Identitas:** Kartu Pelajar
- Nomor Kartu Identitas:** 15615060111012
- Provinsi Lahir:** JAWA TIMUR
- Kota Lahir:** KABUPATEN MALANG
- Tanggal Lahir:** 1997-02-09
- Jenis Kelamin:** Laki-laki
- Skema:** Perekrayaan Perangkat Lunak 2
- Kelas:** RPL-A

On the right side, there is a circular profile picture of a young man. Below it, the status is 'Pengisian asesmen mandiri' with a 'Mulai Asesmen' button. A 'Status Asesmen' section is also present. A table lists documents:

NO	Jenis Dokumen	Action
1	Raport Semester 1-5	Lihat

Below the table, there is a 'Dokumen Pendukung' section with links for 'FR-APL-01 Formulir Permohonan Sertifikasi Kompetensi' and 'FR-APL-02 Asesmen Mandiri'. At the bottom, there is a 'Daftar Dokumen' button.

Gambar 4.38 Contoh Form Mencari Hasil Asesmen Mandiri

Dari gambar 4.38 diperoleh kemungkinan kesalahan (*error guessing*) yang dapat dilihat pada tabel 4.48. Pada tabel tersebut dijabarkan kemungkinan *error* yang akan terjadi pada saat perangkat lunak digunakan.

Tabel 4.48 *Error Guessing* Mencari Hasil Asesmen Mandiri

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Status
1	Membuka dokumen FR-APL-01 pada saat siswa belum mengunggah berkas sertifikasi	<i>Error</i>	Sistem menampilkan peringatan bahwa dokumen belum tersedia	Valid
2	Membuka dokumen FR-APL-02 pada saat hasil asesmen siswa belum dinilai asesor atau belum melaksanakan asesmen mandiri	<i>Error</i>	Sistem menampilkan peringatan bahwa dokumen belum tersedia	Valid

Dari hasil pengujian pada proses mencari hasil asesmen mandiri yang ditunjukkan tabel 4.48 menunjukkan bahwa seluruh pengujian untuk mendapatkan *error* telah dijalankan. Dari hasil tersebut, didapatkan bahwa seluruh pengujian telah dijalankan dan mendapatkan status valid atau sesuai dengan hasil yang diharapkan.

4.4.2 Analisis Hasil Pengujian *Black-box*

Dari hasil pengujian *error guessing* tersebut sudah dapat dipastikan bahwa proses suatu fungsional telah dapat berjalan dengan baik sebelum dilakukan pengujian *error guessing*. Pada dasarnya, pengujian *error guessing* ialah melakukan pengujian pada suatu fungsional yang berjalan dengan suatu kondisi yang diperkirakan akan membuat program tidak berjalan semestinya atau *error*.

Analisis dari hasil pengujian *Black-box* menggunakan teknik *error guessing* yang dilakukan pada 4 (empat) fungsional yang diujikan menunjukkan bahwa seluruh pengujian telah mendapatkan hasil yang valid. Hal tersebut menunjukkan bahwa seluruh kemungkinan kesalahan program atau *error* yang mungkin terjadi pada saat program digunakan telah dapat tertangani dengan baik.

4.4.3 Pengujian *User Acceptance Testing*

Pada proses pengujian *user acceptance testing* menggunakan dua jenis responden yakni asesor dan siswa. Kedua responden tersebut digunakan karena responden tersebut merupakan sasaran pengguna aplikasi. Pengujian dilakukan dengan menggunakan kuesioner yang akan diisi responden setelah responden

menggunakan aplikasi. Untuk hasil pengujian *user acceptance testing* yang telah selesai diisi oleh responden dapat dilihat pada lampiran D untuk hasil pengujian *user acceptance testing* siswa, sedangkan untuk hasil pengujian *user acceptance testing* asesor dapat dilihat pada lampiran E. Berikut pada tabel 4.49 merupakan hasil jawaban dari *user acceptance testing* siswa (Responden 1 dan 2) dan asesor (Responden 3).

Tabel 4.49 Tabel Hasil User Acceptance Testing

No	Pertanyaan	Hasil			Komentar
		Responden 1	Responden 2	Responden 3	
1	Siswa dan asesor: Sistem Informasi Asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang telah berjalan sesuai dengan kebutuhan yang didefinisikan	Ya	Ya	Ya	
2	Asesor: Proses penilaian asesmen mandiri siswa oleh asesor telah berjalan dengan baik Siswa: Proses pendaftaran sertifikasi profesi dapat berjalan dengan baik	Ya	Ya	Ya	
3	Asesor: Proses pencarian siswa dan pencetakan dokumen asesmen mandiri dapat dilayani oleh sistem dengan baik Siswa: Proses pengisian asesmen mandiri siswa dapat dilakukan dengan baik	Ya	Ya	Ya	Responden 1: Sudah bagus, dapat dikembangkan kedepannya.

Tabel 4.49 Tabel Hasil *User Acceptance Testing* (Lanjutan)

No	Pertanyaan	Hasil			Komentar
		Responden 1	Responden 2	Responden 3	
4	Asesor: Proses manajemen asesmen yang meliputi manajemen skema, unit, elemen, dan pertanyaan dapat ditangani sistem dengan baik Siswa: Pengguna menerima e-mail apabila terjadi perubahan status pada proses asesmen	Ya	Ya	Ya	
5	Siswa dan asesor: Informasi maupun tampilan yang diberikan sistem dapat dengan mudah dimengerti	Ya	Ya	Ya	Responden 1: Informasi tersampaikan dengan jelas, tampilan kurang menarik Responden 3: Belum ada perbedaan <i>form</i> yang aktif dengan yang belum aktif.

Dari tabel 4.49 yakni hasil *user acceptance testing*, seluruh pertanyaan mendapatkan jawaban ya, yang menunjukkan bahwa aplikasi dapat diterima sepenuhnya oleh siswa maupun asesor. Namun, terdapat 2 (dua) komentar dari responden 1 atau siswa mengenai poin 3 (tiga) yakni pada pengisian asesmen mandiri siswa yang dapat ditingkatkan kembali dan pada poin 5 (lima) bahwa informasi dari sistem telah dapat tersampaikan dengan baik, namun untuk antar muka atau tampilan sistem dirasa salah satu responden kurang menarik. Serta terdapat 1 (satu) komentar atau masukan yakni pada poin 5 (lima) mengenai tampilan sistem dari responden 3 atau asesor, pada aplikasi belum adanya perbedaan antara *form* yang aktif dengan yang belum aktif. Untuk contoh apabila sistem menampilkan halaman asesmen mandiri, menu pada aplikasi tidak menunjukkan halaman tersebut tengah terbuka.

4.4.4 Analisis Hasil Pengujian *User Acceptance Testing*

Setelah hasil pengujian didapatkan, maka langkah selanjutnya adalah melakukan proses *acceptance decision* atau pengambilan keputusan terkait penerimaan. Dari hasil yang didapatkan, diketahui bahwa seluruh poin pertanyaan mendapatkan jawaban “ya” yang menunjukkan bahwa seluruh fungsional atau dari segi *usefulness* sistem dapat diterima secara penuh oleh pengguna baik siswa maupun asesor. Dengan hasil tersebut dapat ditarik kesimpulan bahwa sistem telah dapat digunakan oleh pengguna untuk membantu proses asesmen mandiri baik untuk siswa maupun asesor. Catatan yang ditinggalkan responden dapat digunakan sebagai saran yang membangun untuk pengembangan aplikasi kedepannya.



BAB 5 PENUTUP

5.1 Kesimpulan

Untuk kesimpulan yang dapat diambil dari proses penelitian yang dilakukan adalah sebagai berikut:

1. Dari hasil analisis kebutuhan sistem informasi asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang bersarkan observasi dokumen asesmen dan wawancara didapatkan 2 (dua) pengguna sistem yakni siswa sebagai pemohon dan peserta sertifikasi dan asesor sebagai penilai peserta sertifikasi. Setelah sasaran pengguna sistem didapatkan, maka dilakukan analisis kebutuhan aplikasi untuk menentukan kebutuhan fungsional yang harus ada pada sistem. Hasil analisis kebutuhan tersebut mendapatkan beberapa fungsional sistem yang digambarkan pada *use case diagram* yang didapatkan 3 (tiga) fungsional siswa, 5 (lima) fungsional asesor, dan 2 (dua) fungsional yang dimiliki kedua aktor. Setelah seluruh kebutuhan didapatkan, maka dilanjutkan dengan pembuatan *use case scenario* yang menjabarkan alur proses masing-masing kebutuhan fungsional. Terakhir, dilakukan proses pembuatan *activity diagram* untuk meringkas hasil *use case scenario* menjadi lebih jelas dalam hal pembagian proses antara pengguna dan sistem.
2. Pada proses perancangan sistem informasi asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang berdasarkan hasil analisis kebutuhan dilakukan proses pembuatan perancangan menggunakan *collaboration diagram*, *sequence diagram*, *class diagram*, dan *physical data model*. *collaboration diagram*, *sequence diagram*, *class diagram* digunakan untuk merancang sistem yang akan dibuat, sedangkan *physical data model* digunakan untuk merancang *database* yang akan dibuat berdasarkan *class diagram* yang telah dibuat. Dari berbagai proses perancangan perangkat lunak tersebut dapat didapatkan sejumlah 6 (enam) kelas *controller* dan 15 (lima belas) kelas *model*. Seluruh hasil perancangan tersebut diimplementasikan menggunakan bahasa pemrograman PHP (*PHP Hypertext Protocol*), *JavaScript*, dan *Ajax*. Implementasi juga dipermudah dengan penggunaan *framework CodeIgniter* yang berstruktur *model*, *view*, *controller* (MVC).
3. Untuk hasil pengujian menggunakan metode *Black-box* dengan teknik *error guessing* mendapatkan hasil bahwa semua kemungkinan kesalahan program atau *error* telah sesuai dengan hasil yang diharapkan dan seluruh pengujian mendapatkan status valid. Sedangkan untuk hasil pengujian menggunakan metode *user acceptance testing* dari segi *usefulness* mendapatkan hasil bahwa sistem telah dapat diterima oleh seluruh responden baik siswa maupun asesor. Hal tersebut ditunjukkan dengan jawaban seluruh responden pada masing-masing butir menghasilkan

jawaban “ya”. Dari hasil pengujian *user acceptance testing* tersebut dapat diambil kesimpulan bahwa menghasilkan *go decision* yakni program telah dapat digunakan secara langsung oleh pengguna.

5.2 Saran

Berikut merupakan saran yang bisa dilakukan untuk mengembangkan penelitian ini lebih lanjut adalah sebagai berikut:

1. Berdasarkan masukan dari responden dapat digunakan sebagai saran ialah peningaktan pada menu asesmen mandiri siswa peningkatan pada *user interface* atau tampilan sistem sehingga lebih *user friendly*.
2. Perlunya pembeda untuk *form* yang sedang aktif dan *form* yang tidak aktif pada *navigation bar* sehingga pengguna dapat mengetahui jenis halaman atau menu yang terbuka.
3. Untuk pengembangan sistem informasi asesmen Lembaga Sertifikasi Profesi (LSP) SMK Negeri 4 Malang seutuhnya dapat dikembangkan menuju asesmen-asesmen lainnya tidak terbatas pada asesmen mandiri.
4. Dapat dikembangkan pula sistem informasi asesmen Lembaga Sertifikasi Profesi (LSP) SMK Negeri 4 Malang pada pengembangan aplikasi *native* berbasis *android*. Dilansir dari *tekno.kompas.com* jumlah pengguna aktif sistem operasi *android* mencapai 2 (dua) miliar lebih.
5. Untuk halaman awal atau *login* hendaknya dipisahkan antara proses *login* siswa dengan asesor.

DAFTAR REFERENSI

- A.S, Rosa dan M. Shalahuddin. 2015, *Rekayasa Perangkat Lunak: Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Ariawan, 2009. *Buku Ajar Sistem Informasi Manajemen*. [e-book]. OSF. Tersedia melalui: OSF <<https://osf.io/p3cqg>> [Diakses 29 Agustus 2018]
- Badan Nasional Sertifikasi Profesi, 2018. *LSP P1 SMKN 4 MALANG*. [online] Badan Nasional Sertifikasi Profesi. Tersedia di: <<https://bnsf.go.id/lsp/373/LSP-P1-SMKN-4-MALANG>> [Diakses 29 Agustus 2018]
- Bertolino, Antonia, 2001. *Software Testing*. [online] Research Area of S. Cataldo. Tersedia di: <<https://pdfs.semanticscholar.org/7464/3ad2486e432b5ca55ee233913ec17a6269b.pdf>> [Diakses 08 Mei 2018]
- Davis, Gordon. 1991, *Kerangka Dasar Sistem Informasi Manajemen Bagian I Pengantar*. Jakarta: PT. Pustaka Binaman Pressindo.
- CodeIgniter, 2018. *Welcome to CodeIgniter*. [online] CodeIgniter. Tersedia di: <https://www.codeigniter.com/user_guide/general/welcome.html> [Diakses 04 September 2018]
- Evantio, Yhouga Beta. 2017. *Pengembangan Sistem Informasi E-Complaint Management (Studi Kasus: Batching Plant Produksi Beton P.T. Holcim Indonesia Regional Jawa Timur*. S1. Universitas Brawijaya.
- Fowler, Susan & Stanwick, Victor. 2004. *Web Application Design Handbook Best Practices for Web-Based Software*. [e-book]. Tersedia melalui: ScienceDirect <<https://www.sciencedirect.com/book/9781558607521/web-application-design-handbook>> [Diakses 02 Mei 2019]
- Galin, Daniel. 2004. *Software Quality Assurance (From theory to implementation)*. [e-book]. Pearson. Tersedia melalui: Politeknik Elektronika Negeri Surabaya <<http://desy.lecturer.pens.ac.id>> [Diakses 29 Agustus 2018]
- Glosarium Kementerian Sosial Republik Indonesia, 2018. *Asesmen*. [online] Kementerian Sosial Republik Indonesia. Tersedia di: <<https://www.kemsos.go.id/glosarium>> [Diakses 04 September 2018]
- Jogiyanto, Hartono. 2005, *Analisis dan Desain Sistem Infromasi (Edisi III)*. Yogyakarta: ANDI.
- Kendall, Kenneth E. dan Julie E. Kendall. 2014, *Analisis dan Perancangan Sistem*. Jakarta: PT. Indeks.
- Kementerian Pendidikan dan Kebudayaan, 2017. *327 SMK Sudah Jadi Lembaga Sertifikasi Profesi*. [online] Kementerian Pendidikan dan Kebudayaan. Tersedia di: <<https://www.kemdikbud.go.id/main/blog/2017/03/327-smk-sudah-jadi-lembaga-sertifikasi-profesi>> [Diakses 29 Agustus 2018]

- Mahmudi, M. 2010. Lembaga Sertifikasi Profesi (LSP) SMK Negeri 4 Malang. Wawancara langsung oleh Arif Rahmadani Vinanda 29 Agustus 2018, pukul 09.30.
- Nurmala, Alawiyah. 2017. *Pengembangan Aplikasi E-School Dengan Pendekatan Evolutionary Prototype Studi Kasus SMP Negeri 1 Cikarang Barat*. S1. Universitas Brawijaya.
- Pathak, Shreyas Vikas, 2016. *Acceptance Testing Technique: A Survey along with Its Operating Frameworks*. [online] International Journal on Recent and Innovation Trends in Computing and Communication. Tersedia di: <http://www.ijritcc.org/download/conferences/ICMTEST_2016/ICMTEST_2016_Track/1463810661_21-05-2016.pdf> [Diakses 29 Agustus 2018]
- Pratama, Fadly Yoga. 2017. *Pengembangan Sistem Informasi Manajemen Pariwisata Terpadu Sebagai Media Promosi Kabupaten Banyuwangi*. S1. Universitas Brawijaya.
- Pressman, R.S. 2015. *Rekayasa Perangkat Lunak: Pendekatan Praktisi Edisi 7 Buku 1*. Yogyakarta: Andi
- Sherman, Rick, 2015. *Business Intelligence Guidebook From Data Integration to Analytics*. [e-book]. Tersedia melalui: ScienceDirect <<https://www.sciencedirect.com/book/9780124114616/business-intelligence-guidebook>> [Diakses 02 Mei 2019]
- Software Testing Help, 2019. *What is User Acceptance Testing (UAT): A Complete Guide*. [online] Software Testing Help. Tersedia di: <<https://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/>> [Diakses 13 Mei 2019]
- Sommerville, Ian, 2011. *Software Engineering 9th Edition*. [e-book]. Moodle USP. Tersedia melalui: Universidade de São Paulo e-Disciplinas <<https://edisciplinas.usp.br>> [Diakses 29 Agustus 2018]
- Surat Keputusan Kementerian Pendidikan Dan Kebudayaan Republik Indonesia nomor 467/D5.6/KP tahun 2018 tentang Perubahan Atas Keputusan Nomor 003/D5.6/KP/2018 tentang Perubahan Atas Keputusan Nomor 1066/D.5.6/KP/2017 tentang Penetapan SMK Sebagai Jejaring Kerja Lembaga Sertifikasi Profesi Pihak Pertama (LSP P1) Di Lingkungan Sekolah Menengah Kejurusan (Tahap 1). Jakarta: Kementerian Pendidikan dan Kebudayaan Republik Indonesia.
- Tong, Qiang. *Construction of RDF(S) From UML Class Diagram*. [online] Journal of Computing and Information Technology. Tersedia di: <https://www.researchgate.net/profile/Jingwei_Cheng/publication/281690028_Construction_of_RDFS_from_UML_class_diagrams/links/56d5929a08ae4774211505d7/Construction-of-RDFS-from-UML-class-diagrams.pdf> [Diakses 13 Mei 2019]

West, Matthew, 2011. *Developing High Quality Data Models*. Tersedia melalui: ScienceDirect <<https://www.sciencedirect.com/book/9780123751065/developing-high-quality-data-models>> [Diakses 02 Mei 2019]



LAMPIRAN A CONTOH DOKUMEN ASESMEN LSP




FR-APL-01. FORMULIR PERMOHONAN SERTIFIKASI KOMPETENSI

Bagian 1 : Rincian Data Pemohon Sertifikasi

Pada bagian ini, cantumkan data pribadi, data pendidikan formal serta data pekerjaan anda pada saat ini.

a. Data Pribadi

Nama lengkap : _____

Tempat / tgl. lahir : _____

Jenis kelamin : Laki-laki / Wanita *

Kebangsaan : _____

Alamat rumah : _____

Kode pos : _____

No. Telepon/E-mail : Rumah : _____ Kantor : _____

HP : _____ E-mail : _____

Pendidikan Terakhir : _____

b. Data Pekerjaan Sekarang

Nama Lembaga/Perusahaan : _____

Jabatan : _____

Alamat : _____

Kode pos : _____

No. Telp/Fax/E-mail : Telp : _____ Fax : _____

E-mail : _____

c. Data permohonan sertifikasi

Tujuan asesmen : ☐ RPL ☐ Pencapaian proses pembelajaran ☐ RCC ☐ Sertifikasi ☐ Lainnya: _____

Skema sertifikasi : Unit/klaster/okupasi/KKNI*: _____

Pere kayasaan Perangkat Lunak II

*) Coret yang tidak sesuai

LSP SMK NEGERI 4 MALANG 2018

FORM APL-01

1



Bagian 2 : Daftar Unit Kompetensi

Pada bagian 2 ini berisikan Unit Kompetensi yang anda ajukan untuk dinilai/diuji kompetensi dalam rangka mendapatkan pengakuan sesuai dengan latar belakang pendidikan, pelatihan serta pengalaman kerja yang anda miliki. Unit kompetensi yang diajukan sesuai dengan Skema Sertifikasi

Judul

Skema Sertifikasi : PEREKAYASAAN PERANGKAT LUNAK II

Nomor :

No	Kode Unit	Judul Unit	Jenis Standar (Standar Khusus/Standar Internasional/SKKNI)
1	TIK.JK01.006.01	Menerapkan prosedur kesehatan, keselamatan, dan keamanan kerja.	SKKNI
2	TIK.PR01.014.01	Bekerja secara efektif dalam lingkungan Teknologi Informasi.	SKKNI
3	TIK.PR02.006.01	Menulis program lanjut.	SKKNI
4	TIK.PR02.008.01	Mengoperasikan bahasa pemrograman terstruktur.	SKKNI
5	TIK.PR02.009.01	Mengoperasikan bahasa pemrograman berorientasi objek (<i>object oriented</i>).	SKKNI
6	TIK.PR02.013.01	Menulis dan mengkompilasi kode program sesuai dengan kebutuhan.	SKKNI
7	TIK.PR02.019.01	Membuat dokumen kode program.	SKKNI
8	TIK.PR02.023.01	Membuat program untuk akses basis data.	SKKNI

Bagian 3 : Bukti Kelengkapan Pemohon

Unit/Elemen Kompetensi	Bukti (paling relevan) : Rincian Pendidikan/Pelatihan, Pengalaman Kerja, Pengalaman Hidup
1. Menerapkan prosedur kesehatan, keselamatan, dan keamanan kerja.	
2. Bekerja secara efektif dalam lingkungan Teknologi Informasi.	
3. Menulis program lanjut.	
4. Mengoperasikan bahasa pemrograman terstruktur.	
5. Mengoperasikan bahasa pemrograman berorientasi objek (<i>object oriented</i>).	
6. Menulis dan mengkompilasi kode program sesuai dengan kebutuhan.	
7. Membuat dokumen kode program.	
8. Membuat program untuk akses basis data.	



Rekomendasi :	Pemohon :	
	Nama	
Catatan :	Tanda tangan/ Tanggal	
	Asesor :	
	Nama	
	No. Reg.	
	Tanda tangan/ Tanggal	





FR-APL-02 ASESMEN MANDIRI

Nama Peserta : _____ Tanggal/Waktu : _____

Nama Asesor : 1. _____ TUK : Sewaktu/Tempat Kerja/Mandiri*)
2. _____ *) coret yang tidak sesuai

Pada bagian ini, anda diminta untuk menilai diri sendiri terhadap unit (unit-unit) kompetensi yang akan di-ases.

1. Pelajari seluruh standar Kriteria Unjuk Kerja (KUK), batasan variabel, panduan penilaian dan aspek kritis serta yakinkan bahwa anda sudah benar-benar memahami seluruh isinya.
2. Laksanakan penilaian mandiri dengan mempelajari dan menilai kemampuan yang anda miliki secara obyektif terhadap seluruh daftar pertanyaan yang ada, serta tentukan apakah sudah kompeten (K) atau belum kompeten (BK) dengan mencantumkan tanda ✓ dan tuliskan bukti-bukti pendukung yang anda anggap relevan terhadap setiap elemen/KUK unit kompetensi.
3. Asesor dan Peserta menandatangani form Asesmen Mandiri

Kode Unit Kompetensi	TIK.JK01.006.01									
Judul Unit Kompetensi	Menerapkan prosedur kesehatan, keselamatan, dan keamanan kerja									
Elemen Kompetensi	01 Menetapkan hal-hal yang berhubungan dengan kesehatan dan keselamatan kerja di lingkungan kerja									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :	Penilaian		Bukti-bukti Pendukung	Diisi Asesor						
	K	BK		V	A	T	M			
1.1 Apakah anda dapat menetapkan pengawasan kesehatan dan keselamatan kerja sesuai undang-undang kesehatan dan keselamatan kerja.										
1.2 Apakah anda dapat meninjau dan melaporkan prosedur kesehatan dan keselamatan kerja di lingkungan kerja untuk memperbaiki masalah yang ada pada pengawas.										
1.3 Apakah anda dapat melaksanakan prosedur kesehatan dan keselamatan kerja untuk menjamin keamanan di lingkungan kerja.										
Elemen Kompetensi	02 Mendokumentasikan dan menyebarkan syarat- syarat kesehatan dan keselamatan kerja									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :	Penilaian		Bukti-bukti Pendukung	Diisi Asesor						
	K	BK		V	A	T	M			
2.1 Apakah anda dapat mengumpulkan informasi yang berhubungan dengan peraturan kesehatan dan keselamatan kerja dan syarat-syaratnya										

Kode Unit Kompetensi	TIK.PR01.014.01							
Judul Unit Kompetensi	Bekerja Secara Efektif Dalam Lingkungan Teknologi Informasi							
Elemen Kompetensi	01 Memenuhi prosedur dan kebijakan teknologi informasi							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor			
		K	BK		V	A	T	M
1.1 Apakah anda dapat menetapkan dan menjelaskan peranan pelaku utama dari organisasi teknologi informasi								
1.2 Apakah anda dapat menetapkan pilihan karir dan opsi-opsi sesuai struktur organisasi								
1.3 Apakah anda dapat memenuhi kebijakan dan prosedur-prosedur seperti yang diarahkan oleh pengawas								



Elemen Kompetensi	02 Meningkatkan organisasi dan departemen teknologi informasi dengan cara yang konsisten dengan misi organisasi									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor					
		K	BK		V	A	T	M		
2.1 Apakah anda dapat menjelaskan peranan fungsi teknologi informasi dalam organisasi										
2.2 Apakah anda dapat mempromosikan organisasi dengan cara positif										
Elemen Kompetensi	03 Mengidentifikasi peralatan/ software teknologi informasi dan sistim operasi yang didukung oleh organisasi									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor					
		K	BK		V	A	T	M		
3.1 Apakah anda dapat mengidentifikasi peralatan/ software teknologi informasi dan sistim operasi yang tepat sesuai kebutuhan organisasi										
3.2 Apakah anda dapat mengidentifikasi persyaratan peralatan, lokasi dan pelayanan sesuai kebutuhan organisasi										

Kode Unit Kompetensi	TIK.PR.02.006.01									
Judul Unit Kompetensi	Menulis Program Lanjut									
Elemen Kompetensi	01 Menerapkan kualitas program									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor					
		K	BK		V	A	T	M		
1.1 Apakah anda dapat memakai parameter kualitas program sesuai standar kualitas										
1.2 Apakah anda dapat memakai kompleksitas dari program sesuai dengan standar kualitas										
Elemen Kompetensi	02 Menulis program									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor					
		K	BK		V	A	T	M		
2.1 Apakah anda dapat menulis program sesuai dengan desain modul yang sudah dibuat dan memenuhi aspek kualitas										
2.1 Apakah anda dapat menulis program mengikuti standar bahasa pemrograman yang telah ditentukan										



2.2	Apakah anda dapat memasukkan penanganan kesalahan program dalam penulisan kode program dan menulis kode program yang dapat menampilkan kesalahan kode program tanpa menyebabkan program menjadi tak jalan								
Elemen Kompetensi		03 Mengkompilasi program							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
3.1	Apakah anda dapat mengkompilasi program dikompilasi dengan benar, menunjukkan kesalahan kompilasi oleh compiler dan membetulkan kesalahan yang timbul dibetulkan sesuai dengan jenis kesalahannya								
3.2	Apakah anda dapat me- running program bebas dari kesalahan								

Kode Unit Kompetensi		TIK.PR.02.008.01							
Judul Unit Kompetensi		Mengoperasikan Bahasa Pemrograman Terstruktur							
Elemen Kompetensi		01 Menggunakan tipe data dan kontrol program							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
1.1	Apakah anda dapat mengetahui dan mempelajari Tipe data berupa tipe data dasar dan tipe data bentukan								
1.2	Apakah anda dapat mengetahui dan mempelajari aturan program berupa pemakaian nama dan sintak program								
1.3	Apakah anda dapat mengaplikasikan Kontrol program. Kontrol program merupakan urutan atau sekuensial program, harus terkendali dalam kode program								
Elemen Kompetensi		02 Membuat program sederhana							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
2.1	Apakah anda dapat membuat program baca tulis untuk masukan data dari keyboard dan menampilkan ke layar monitor termasuk variasinya sesuai tandar input/output								





2.2	Apakah anda dapat menggunakan analisis kasus percabangan, pengulangan dalam membuat program								
Elemen Kompetensi		03 Membuat program menggunakan prosedur dan fungsi							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
3.1	Apakah anda dapat membuat program dengan menggunakan prosedur sesuai aturan penulisan program								
3.2	Apakah anda dapat membuat program dengan menggunakan fungsi sesuai aturan penulisan program								
3.3	Apakah anda dapat membuat program dengan menggunakan prosedur dan fungsi secara bersamaan sesuai aturan penulisan program								
Elemen Kompetensi		04 Membuat program menggunakan array							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
4.1	Apakah anda dapat membuat program menggunakan array satu dimensi dan dua dimensi dibuat								
4.2	Apakah anda dapat mengaplikasikan pencarian data dan mengurutkan data dalam array								
Elemen Kompetensi		05 Membuat program untuk akses file							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
5.1	Apakah anda dapat membuat program untuk menulis data kedalam media penyimpanan data dan membaca data dari media penyimpan data								
Elemen Kompetensi		06 Membuat program menggunakan pointer							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
6.1	Apakah anda dapat membuat Program untuk menulis data dan membaca data dengan menggunakan pointer								





6.2 Apakah anda dapat mengoperasikan data pointer sesuai dengan kebutuhan seperti untuk pencarian data, pengurutan data, menambah data, menghapus data dsb.									
Elemen Kompetensi	07 Mengkompilasikan program								
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
7.1 Apakah anda dapat mengkompilasi program dengan mensetting pada editor program dan kode program yang dilibatkan									
7.2 Apakah anda dapat mengkoreksi kesalahan sintak dan logika agar bebas dari error dan dapat di jalankan									

Kode Unit Kompetensi	TIK.PR02.009.01								
Judul Unit Kompetensi	Mengoperasikan Bahasa Pemrograman Berorientasi Objek (<i>Object Oriented</i>)								
Elemen Kompetensi	01 Menggunakan tipe data dan kontrol program								
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
1.1 Apakah anda dapat menjelaskan tipe data dasar dan tipe data bentukan									
1.2 Apakah anda dapat memahami dan mengaplikasikan tata cara penulisan program OO									
1.3 Apakah anda dapat membuat dan mengendalikan kontrol program berupa urutan atau sekuensial program dalam kode program									
Elemen Kompetensi	02 Membuat program object oriented dengan class								
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
2.1 Apakah anda dapat membuat Program dengan menggunakan objek atau class									
2.2 Apakah anda dapat membuat properti class yang akan di realisasikan dalam bentuk prosedur/fungsi									

2.3	Apakah anda dapat membuat data di dalam class mandiri. Artinya data di dalam class tidak dapat di akses secara langsung oleh class lain. Data hanya dapat di akses melalui property class (berupa fungsi atau prosedur)								
Elemen Kompetensi		03 Membuat program object oriented dengan inheritace, polimorphy, overloading dan friends							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung		Diisi Asesor			
		K	BK			V	A	T	M
3.1	Apakah anda dapat mengaplikasikan inheritance pada class								
3.2	Apakah anda dapat mengaplikasikan Polimorphy pada class								
3.3	Apakah anda dapat mengaplikasikan Overloading pada class								
3.4	Apakah anda dapat mengaplikasikan Friends pada class								
Elemen Kompetensi		04 Membuat program object oriented dengan interface dan paket							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung		Diisi Asesor			
		K	BK			V	A	T	M
1.	Apakah anda dapat membuat interface class program								
4.2	Apakah anda dapat membuat paket dengan menggunakan class program								
Elemen Kompetensi		05 Mengkompilasi dengan object oriented							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung		Diisi Asesor			
		K	BK			V	A	T	M
5.1	Apakah anda dapat mengkompilasi program dengan mensetting pada editor program dan kode program								
5.2	Apakah anda dapat mengkoreksi dan membetulkan kesalahan sintak dan logika serta membuat dan menjalankan program yang bebas dari error								



Kode Unit Kompetensi	TIK.PR02.013.01									
Judul Unit Kompetensi	Menulis dan Mengkompilasi Kode Program Sesuai dengan Kebutuhan									
Elemen Kompetensi	01 Mendeklarasi-kan dan menetapkan variable-variable									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :	Penilaian		Bukti-bukti Pendukung	Diisi Asesor						
	K	BK		V	A	T	M			
1.1 Apakah anda dapat mendeklarasikan penamaan variable sesuai kaidah dan benar										
1.2 Apakah anda dapat mendeklarasikan variabel-variabel menurut persyaratan										
1.3 Apakah anda dapat mengumpulkan variabel-variabel dinamis yang <i>garbage</i> setelah menggunakan										
Elemen Kompetensi	02 Membuat kode modul									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :	Penilaian		Bukti-bukti Pendukung	Diisi Asesor						
	K	BK		V	A	T	M			
2.1 Apakah anda dapat menetapkan dan mengembangkan kelas instansi atau kode modul-modul										
2.2 Apakah anda dapat memenuhi kepaduan dan standar perangkat/ <i>coupling</i> modul-modul										
2.3 Apakah anda dapat mengembangkan array dinamis, table, dan struktur memori										
Elemen Kompetensi	03 Menguji setiap modul									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :	Penilaian		Bukti-bukti Pendukung	Diisi Asesor						
	K	BK		V	A	T	M			
3.1 Apakah anda dapat mengembangkan pengujian rutin untuk memverifikasi bahwa kode yang dihasilkan memenuhi persyaratan										
3.2 Apakah anda dapat menguji struktur memori yaitu <i>array</i> untuk pelanggaran batas										
3.3 Apakah anda dapat mengakhiri struktur kontrol atau <i>loop</i>										



Elemen Kompetensi	04 Menetapkan prosedur-prosedur <i>exception handling</i>									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :	Penilaian		Bukti-bukti Pendukung	Diisi Asesor						
	K	BK		V	A	T	M			
4.1 Apakah anda dapat menentukan pengkodean area dimana <i>exception</i> dapat terjadi										
4.2 Apakah anda dapat mengimplementasikan rutin-rutin <i>exception handling</i>										
4.3 Apakah anda dapat menetapkan semua <i>exception</i> sudah tercakup										
Elemen Kompetensi	05 Menggunakan teknik-teknik <i>debugging</i> dan penanganan error									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :	Penilaian		Bukti-bukti Pendukung	Diisi Asesor						
	K	BK		V	A	T	M			
5.1 Apakah anda dapat menggunakan teknik <i>debugging</i> dan fasilitas penanganan error										
5.2 Apakah anda dapat menggunakan metoda-metoda error handling eksternal (penggunaan data base) menyisakan kepaduan sangat tinggi dan <i>loosly coupled</i>										
5.3 Apakah anda dapat mengkompilasi dan menguji kode dan di uji jika di perlukan di terasi										
Elemen Kompetensi	06 Mendokumentasikan kode program									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :	Penilaian		Bukti-bukti Pendukung	Diisi Asesor						
	K	BK		V	A	T	M			
6.1 Apakah anda dapat menggabungkan dokumentasi internal ke dalam program										
6.2 Apakah anda dapat membuat dokumentasi eksternal sesuai kebutuhan user dan pemelihara										
6.3 Apakah anda dapat memahami dan memelihara dokumentasi eksternal oleh user										





Kode Unit Kompetensi	TIK.PR02.019.01									
Judul Unit Kompetensi	Membuat Dokumen Kode Program									
Elemen Kompetensi	01 Mengidentifikasi kode program									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :					Penilaian K BK		Bukti-bukti Pendukung		Diisi Asesor V A T M	
1.1 Apakah anda dapat mengidentifikasi modul program berdasarkan fungsi, prosedur dan data										
1.2 Apakah anda dapat mengetahui parameter data yang menjadi masukan dan keluaran pada fungsi atau prosedur										
1.3 Apakah anda dapat menjelaskan data dan algoritman pada deskripsi fungsi dan prosedur										
Elemen Kompetensi	02 Membuat Dokumentasi Modul Program									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :					Penilaian K BK		Bukti-bukti Pendukung		Diisi Asesor V A T M	
2.1 Apakah anda dapat membuat dokumentasi modul sesuai dengan identitas untuk memudahkan pelacakan										
2.2 Apakah anda dapat menjelaskan nama fungsi dan prosedur yang ada di dalam modul pada dokumentasi modul										
Elemen Kompetensi	03 Membuat dokumentasi fungsi atau prosedur program									
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :					Penilaian K BK		Bukti-bukti Pendukung		Diisi Asesor V A T M	
3.1. Apakah anda dapat membuat dokumentasi fungsi dan prosedur sesuai dengan identitas fungsi atau prosedur										
3.2 Apakah anda dapat menjelaskan parameter data masukan dan keluaran di jelaskan pada dokumen										
3.3 Apakah anda dapat menjelaskan data dan tipe data yang ada fungsi atau prosedur di jelaskan pada dokumen										



Elemen Kompetensi	04 Mengevaluasi Dokumen kode						
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :	Penilaian		Bukti-bukti Pendukung	Diisi Asesor			
	K	BK		V	A	T	M
4.1 Apakah anda dapat merevisi dokumen berdasarkan standar dokumen							
4.2 Apakah anda dapat mengoreksi Dokumen untuk menjamin kesempurnaan teknis dan tata bahasa							
4.3 Apakah anda dapat melengkapi dokumen untuk memenuhi kebutuhan dokumen teknis							

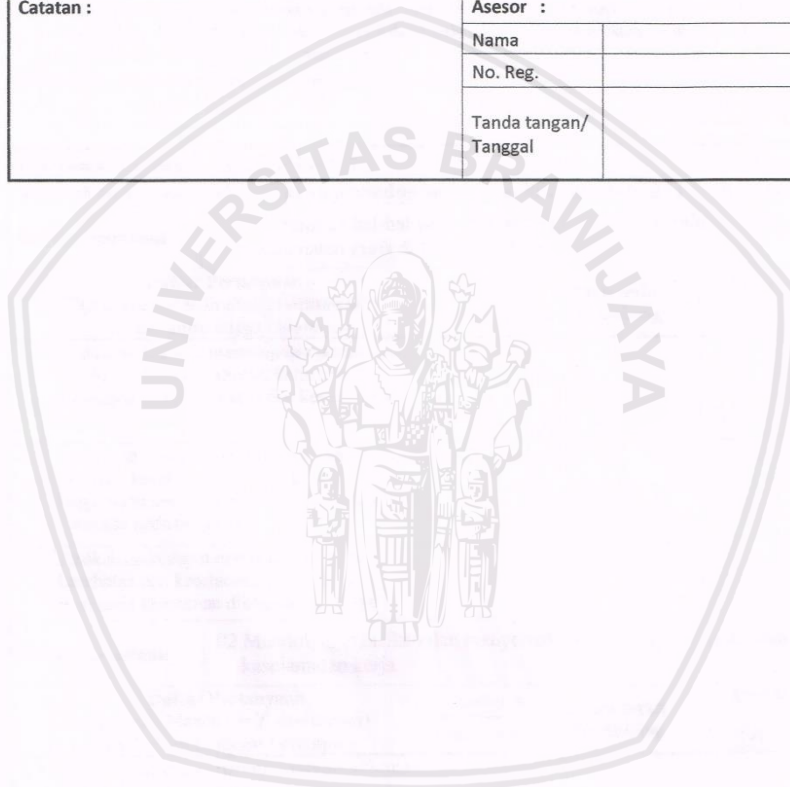
Kode Unit Kompetensi	TIK.PR02.023.01						
Judul Unit Kompetensi	Membuat Program Untuk Akses Basis Data						
Elemen Kompetensi	01 Mengidentifikasi-kasi basis data						
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :	Penilaian		Bukti-bukti Pendukung	Diisi Asesor			
	K	BK		V	A	T	M
1.1 Apakah anda dapat memenuhi kebutuhan sistim sesuai skope dan fungsi dari basis data							
1.2 Apakah anda dapat membuat bentuk struktur data, bentuk querie, bentuk tampilan, bentuk laporan (cetak) disesuaikan dengan table							
1.3 Apakah anda dapat menggunakan sekuriti dan metode akses pada basis data sesuai perancangan							
Elemen Kompetensi	02 Membuat program basis data						
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :	Penilaian		Bukti-bukti Pendukung	Diisi Asesor			
	K	BK		V	A	T	M
2.1 Apakah anda dapat membangun struktur data pada bahasa pemrograman sesuai dengan disain yang telah di buat. Apakah anda dapat mengimplementasikan struktur data dalam bentuk table dan view, mengikuti disain yang telah di buat							
2.2 Apakah anda dapat menormalisasikan Basis data agar menyederhanakan relasi table dan memperkokoh relasi table							

2.3 Apakah anda dapat membangun query, tampilan dan laporan (cetak) menurut bahasa pemrograman yang sesuai dengan disain. Mengimplementasikan Query dengan prosedur, fungsi maupun trigger. Tampilan dan laporan akan berisikan data untuk menampilkan table atau view									
Elemen Kompetensi		03 Membuat koneksi dan sekuriti basis data							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
3.1 Apakah anda dapat membangun pengguna dan password mengikuti disain yang di buat. Apakah anda dapat mengimplementasikan pengguna dan password dalam mengakses basis data pada DBMS atau dalam bahasa pemrograman									
3.2 Apakah anda dapat membangun privilege setiap pengguna terhadap basis data mengikuti disain yang di buat. Apakah anda dapat mengimplementasikan privilege pengguna terhadap basis data dapat pada DBMS. Sedangkan privilege untuk mengakses tampilan dan laporan di implementasikan pada bahasa pemrograman									
Elemen Kompetensi		04 Menguji program basis data							
Daftar Pertanyaan (Asesmen Mandiri/Self Assessment) Apakah anda dapat / mampu :		Penilaian		Bukti-bukti Pendukung	Diisi Asesor				
		K	BK		V	A	T	M	
4.1 Apakah anda dapat menyiapkan data uji dan skenario uji. Data uji merupakan data yang akan di masukkan kedalam sistim. Sedangkan skenario uji merupakan urutan/langkah pengujian									
4.2 Apakah anda dapat memeriksa, mengkoreksi logika, mengkompilasi dan menjalankan program basis data. Jika ditemukan kesalahan, kesalahan tersebut di betulkan. Program basis data dapat di kompilasi dan di jalankan									
4.3 Apakah anda dapat menyimpan dan menampilkan data uji sesuai kebutuhan. Apakah anda dapat mencetak laporan sesuai kebutuhan. Apakah anda dapat menemukan, memeriksa dan mengkoreksi kesalahan pada tampilan dan laporan									



4.4 Apakah anda dapat menguji program basis data memenuhi kebutuhan yang telah di tentukan. Hasil pengujian harus menunjukkan pemenuhan terhadap kebutuhan/spesifikasi program									
---	--	--	--	--	--	--	--	--	--

Rekomendasi Asesor : ☐ Asesmen dilanjutkan ☐ Mengikuti Pelatihan ☐ On The Job Training	Asesi :	
	Nama	
	Tanda tangan/ Tanggal	
Catatan :	Asesor :	
	Nama	
	No. Reg.	
	Tanda tangan/ Tanggal	



LAMPIRAN B KERANGKA WAWANCARA

LEMBAR WAWANCARA

Tempat :
Tanggal :
Lokasi :
Waktu :
Narasumber :

1. Apakah SMKN 4 Malang sudah mempunyai lisensi LSP dari BNSP?

2. Apakah dalam proses asesmen terdapat banyak sekali dokumen yang harus diisi?

3. Berapa jumlah dokumen (dalam lembar kertas) yang harus dimiliki oleh satu siswa?

4. Apakah dokumen tersebut akan disimpan untuk arsip setelah dilakukan asesmen?

5. Apakah sekolah telah menyediakan komputer untuk menunjang proses asesmen?

6. Apakah tersedia jaringan internet pada komputer yang disediakan?

7. Apakah dibutuhkan suatu sistem informasi asesmen untuk LSP di SMK Negeri 4 Malang?

8. Apakah sistem informasi perlu untuk siswa mendaftar atau telah disediakan daftar siswa?

CATATAN

Malang,

2019

LAMPIRAN C HASIL WAWANCARA

LEMBAR WAWANCARA

Tempat : Laboratorium RPL
 Tanggal : 20 Februari 2018
 Lokasi : SMK Negeri 4 Malang
 Waktu : 09.00
 Narasumber : Bpk. M. Mahmudi

1. Apakah SMKN 4 Malang sudah mempunyai lisensi LSP dari BNSP?

Sudah, mulai tahun 2016

2. Apakah dalam proses asesmen terdapat banyak sekali dokumen yang harus diisi?

Ya, sesuai jumlah siswa yang mendaftar.

3. Berapa jumlah dokumen (dalam lembar kertas) yang harus dimiliki oleh satu siswa?

Sekitar 50 lembar

4. Apakah dokumen tersebut akan disimpan untuk arsip setelah dilakukan asesmen?

Ya, untuk bukti bagi BNSP

5. Apakah sekolah telah menyediakan komputer untuk menunjang proses asesmen?

Ya, untuk menghindari kecurangan dengan laptop mereka sendiri

6. Apakah tersedia jaringan internet pada komputer yang disediakan?

Ya, lancar

7. Apakah dibutuhkan suatu sistem informasi asesmen untuk LSP di SMK Negeri 4 Malang?

Ya, untuk menghemat penyimpanan serta penggunaan kertas

8. Apakah sistem informasi perlu untuk siswa mendaftar atau telah disediakan daftar siswa?

Siswa tetap mendaftar secara individu, karena sistem dalam sekolah belum terintegrasi

CATATAN

Malang, 20 FEBRUARI 2019

M. MAHMUDI

LAMPIRAN D USER ACCEPTANCE TESTING SISWA

USER ACCEPTANCE TEST

User acceptance test berisikan pertanyaan-pertanyaan yang diajukan kepada *stakeholder* dalam hal ini responden berperan sebagai siswa pemohon sertifikasi untuk Lembaga Sertifikasi Profesi (LSP) SMK Negeri 4 Malang. Pertanyaan ini bertujuan untuk mengetahui apakah sistem yang dibuat telah membantu menyelesaikan masalah yang dilihat dari sudut pandang pemakai aplikasi. Selain itu, lembar ini juga dijadikan sebagai tanda persetujuan pengguna bahwa aplikasi yang dibuat dapat diterima oleh pengguna.

Daftar pertanyaan yang harus dijawab oleh siswa selaku pengguna yang dapat dilihat pada Tabel 1, dimana pertanyaan dapat dijawab dengan memberikan tanda *checklist* (V) pada jawaban YA atau TIDAK. Selain itu, sebagai tambahan, pengguna dapat memberikan masukan atau komentar pada kolom yang disediakan.

Tabel 1 Tabel Pertanyaan Penerimaan Pengguna

No.	Pertanyaan	Jawaban		Komentar
		Ya	Tidak	
1	Sistem Informasi Asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang telah berjalan sesuai dengan kebutuhan yang didefinisikan	✓		
2	Proses pendaftaran sertifikasi profesi dapat berjalan dengan baik	✓		
3	Proses pengisian asesmen mandiri siswa dapat dilakukan dengan baik	✓		Sudah bagus. Dapat dikembangkan kedepannya.
4	Pengguna menerima e-mail apabila terjadi perubahan status pada proses asesmen	✓		
5	Informasi maupun tampilan yang diberikan sistem dapat dengan mudah dimengerti	✓		Informasi tersampaikan dengan jelas, tampilan kurang menarik.

Malang, 8 - 5 - 2019

Mengetahui


MUBANDHA ARIAN YUDIANTO.

USER ACCEPTANCE TEST

User acceptance test berisikan pertanyaan-pertanyaan yang diajukan kepada *stakeholder* dalam hal ini responden berperan sebagai siswa pemohon sertifikasi untuk Lembaga Sertifikasi Profesi (LSP) SMK Negeri 4 Malang. Pertanyaan ini bertujuan untuk mengetahui apakah sistem yang dibuat telah membantu menyelesaikan masalah yang dilihat dari sudut pandang pemakai aplikasi. Selain itu, lembar ini juga dijadikan sebagai tanda persetujuan pengguna bahwa aplikasi yang dibuat dapat diterima oleh pengguna.

Daftar pertanyaan yang harus dijawab oleh siswa selaku pengguna yang dapat dilihat pada Tabel 1, dimana pertanyaan dapat dijawab dengan memberikan tanda *checklist* (V) pada jawaban YA atau TIDAK. Selain itu, sebagai tambahan, pengguna dapat memberikan masukan atau komentar pada kolom yang disediakan.

Tabel 1 Tabel Pertanyaan Penerimaan Pengguna

No.	Pertanyaan	Jawaban		Komentar
		Ya	Tidak	
1	Sistem Informasi Asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang telah berjalan sesuai dengan kebutuhan yang didefinisikan	✓		
2	Proses pendaftaran sertifikasi profesi dapat berjalan dengan baik	✓		
3	Proses pengisian asesmen mandiri siswa dapat dilakukan dengan baik	✓		
4	Pengguna menerima e-mail apabila terjadi perubahan status pada proses asesmen	✓		
5	Informasi maupun tampilan yang diberikan sistem dapat dengan mudah dimengerti	✓		

Malang, 8 Mei 2019
Mengetahui


Anisatus Soladah

LAMPIRAN E USER ACCEPTANCE TESTING ASESOR

USER ACCEPTANCE TEST

User acceptance test berisikan pertanyaan-pertanyaan yang diajukan kepada *stakeholder* dalam hal ini responden berperan sebagai asesor Lembaga Sertifikasi Profesi (LSP) SMK Negeri 4 Malang. Pertanyaan ini bertujuan untuk mengetahui apakah sistem yang dibuat telah membantu menyelesaikan masalah yang dilihat dari sudut pandang pemakai aplikasi. Selain itu, lembar ini juga dijadikan sebagai tanda persetujuan pengguna bahwa aplikasi yang dibuat dapat diterima oleh pengguna.

Daftar pertanyaan yang harus dijawab oleh asesor selaku pengguna yang dapat dilihat pada Tabel 1, dimana pertanyaan dapat dijawab dengan memberikan tanda *checklist* (V) pada jawaban YA atau TIDAK. Selain itu, sebagai tambahan, pengguna dapat memberikan masukan atau komentar pada kolom yang disediakan.

Tabel 1 Tabel Pertanyaan Penerimaan Pengguna

No.	Pertanyaan	Jawaban		Komentar
		Ya	Tidak	
1	Sistem Informasi Asesmen Lembaga Sertifikasi Profesi SMK Negeri 4 Malang telah berjalan sesuai dengan kebutuhan yang didefinisikan	✓		
2	Proses penilaian asesmen mandiri siswa oleh asesor telah berjalan dengan baik	✓		
3	Proses pencarian siswa dan pencetakan dokumen asesmen mandiri dapat dilayani oleh sistem dengan baik	✓		
4	Proses manajemen asesmen yang meliputi manajemen skema, unit, elemen, dan pertanyaan dapat ditangani sistem dengan baik	✓		
5	Informasi maupun tampilan yang diberikan sistem dapat dengan mudah dimengerti	✓		Belum Ada Debetoran Form yang Aktif yg yg belum Aktif

Malang, 9 - 6 - 2019
Mengetahui

M. MAHMUDI
NIP. 19830514 201001 1008