

**APLIKASI SURAT KETERANGAN PENDAMPING IJAZAH
BERBASIS WEB**

(Studi Kasus: Fakultas Teknik Universitas Katolik De La Salle Manado)

LAPORAN KERJA PRAKTIK

Disusun oleh:

Micael Liem

17013009



PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS KATOLIK DE LA SALLE

MANADO

2020

**APLIKASI SURAT KETERANGAN PENDAMPING IJAZAH
BERBASIS WEB**

(Studi Kasus: Fakultas Teknik Universitas Katolik De La Salle Manado)

LAPORAN KERJA PRAKTIK

Ditulis untuk Memenuhi Persyaratan Mata Kuliah Kerja Praktik

(IF5161)

Disusun oleh:

Micael Liem

17013009



PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS KATOLIK DE LA SALLE

MANADO

2020

LEMBAR PENGESAHAN
LAPORAN KERJA PRAKTIK

Judul:
APLIKASI SURAT KETERANGAN PENDAMPING IJAZAH
BERBASIS WEB
(Studi Kasus: Fakultas Teknik Universitas Katolik De La Salle
Manado)

Telah disetujui dan disahkan pada tanggal:
Oleh:
Fakultas Teknik Universitas Katolik De La Salle Manado

Mengetahui,
Tanda Tangan & Cap Perusahaan

Ronald Albert Rachmadi, S.T., M.T.
Dekan Fakultas Teknik

LEMBAR PERNYATAAN

Yang bertandatangan di bawah ini:

Nama : Micael H. Liem
NIM : 17013009
Tempat, Tanggal Lahir : Bitung 27 Mei 1999
Fakultas / Jurusan : Teknik / Teknik Informatika

Menyatakan bahwa Laporan Kerja Praktik dan atau Aplikasi / Program berjudul **Aplikasi Surat Keterangan Pedamping Ijazah Berbasis Web (Studi Kasus: Fakultas Teknik Universitas Katolik De La Salle Manado)** yang saya buat adalah benar hasil karya saya dan bukan karya tulis orang lain, baik sebagian atau seluruhnya kecuali dalam bentuk kutipan yang telah disebutkan sumbernya.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya dan apabila pernyataan ini tidak benar maka saya bersedia menerima sanksi akademis sesuai dengan yang ditetapkan oleh Fakultas, berupa pembatalan Laporan Kerja Praktik dan hasilnya.

Manado, 18 Desember 2020

Menyatakan,

Micael H. Liem

Menyetujui,

Dosen Pembimbing I

Dosen Pembimbing II

Lanny Sitanayah, S.T., M.Sc., Ph.D.

Vivie. D. Kumenap, S.T., M.Cs.

Mengetahui,

Ketua Program Studi

Dekan Fakultas Teknik

Vivie. D. Kumenap, S.T., M.Cs.

Ronald Albert Rachmadi, S.T., M.T.



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KATOLIK DE LA SALLE
MANADO**

FORM KP - 003

FORMULIR DATA UMUM PERUSAHAAN

NAMA MAHASISWA : Micael H. Liem
NIM : 17013009
NAMA PERUSAHAAN : Fakultas Teknik Universitas Katolik De La
Salle Manado

ALAMAT PERUSAHAAN : Kairagi I Kombos Manado, Sulawesi Utara

DIDIRIKAN TAHUN : 2000
IJIN USAHA : Surat Keputusan Menteri Pendidikan Nomor
123/D/O/2000
JUMLAH KARYAWAN : ± 50
PIMPINAN : Ronald Albert Rachmadi, S.T., M.T.

WAKIL PERUSAHAAN
Tanggal : 18 Desember 2020
Nama : Vivie. D. Kumenap, S.T., M.Cs.
Jabatan : Ketua Program Studi Informatika

(Tanda tangan dan
cap perusahaan) : Ronald Albert Rachmadi, S.T., M.T.



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KATOLIK DE LA SALLE
MANADO**

FORM KP - 004

FORMULIR PENILAIAN KEMAJUAN KERJA PRAKTIK

A. UMUM

Nama Mahasiswa : Micael H. Liem
NIM Mahasiswa : 17013009
Program Studi : Teknik Informatika
Dosen Pembimbing Akademik : Lanny Sitanayah, S.T., M.Sc.,
Ph.D.
Topik/Rencana Bidang : Aplikasi Surat Keterangan
Pendamping Ijazah Berbasis Web
Pembimbing 1 : Lanny Sitanayah, S.T., M.Sc.,
Ph.D.
Pembimbing 2 : Vivie. D. Kumenap, S.T., M.Cs.
Terhitung Mulai : 14 Agustus 2020
Target Selesai : 12 Desember 2020

B. KEGIATAN PELAKSANAAN KERJA PRAKTIK

No.	Tanggal	Jenis Kegiatan	Paraf Pembimbing
1.	21-09-2020	Konsultasi laporan KP	
2.	05-10-2020	Konsultasi revisi laporan KP	
3.	27-10-2020	Konsultasi laporan KP	
4.	20-11-2020	Konsultasi revisi laporan KP	
5.	22-11-2020	Konsultasi revisi laporan KP	
6.	05-12-2020	Konsultasi laporan KP	
7.	07-12-2020	Konsultasi revisi laporan KP	

No.	Tanggal	Jenis Kegiatan	Paraf Pembimbing
8.	12-12-2020	Konsultasi revisi laporan KP	
9.	14-09-2020	Meminta data umum perusahaan	
10.	17-09-2020	Meminta data umum perusahaan	
11.	24-09-2020	Konsultasi laporan KP	
12.	28-09-2020	Konsultasi revisi laporan KP	
13.	12-11-2020	Konsultasi laporan KP	
14.	25-11-2020	Konsultasi revisi laporan KP	
15.	02-12-2020	Konsultasi laporan KP	
16.	12-12-2020	Konsultasi revisi laporan KP	

Manado, 12 Desember 2020

Dosen Pembimbing KP

Lanny Sitanayah, S.T., M.Sc., Ph.D.



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KATOLIK DE LA SALLE
MANADO**

FORM KP - 005

FORMULIR PENILAIAN PELAKSANAAN KERJA PRAKTIK

Mohon diisi dan dicek seperlunya,

NAMA MAHASISWA : Micael H. Liem
NIM : 17013009
NAMA PERUSAHAAN : Fakultas Teknik Universitas Katolik
De La Salle Manado
ALAMAT PERUSAHAAN : Kairagi I Kombos Manado, Sulawesi
Utara
TGL KERJA PRAKTIK : 14 Agustus 2020
TOPIK YANG DIBAHAS : Aplikasi Surat Keterangan
Pendamping Ijazah Berbasis Web

Nilai							
Sikap	=	50	60	70	80	90	100
Kerajinan	=	50	60	70	80	90	100
Prestasi	=	50	60	70	80	90	100

KOMENTAR/SARAN

NILAI RATA-RATA :
TANGGAL :
NAMA PENILAI
JABATAN :
(Tanda tangan
dan cap perusahaan) :

KATA PENGANTAR

Puji dan Syukur saya panjatkan kepada Tuhan yang Maha Esa karena kasih dan penyertaan-Nya, sehingga laporan Kerja Praktik (KP) mengenai “Aplikasi Surat Keterangan Pedamping Ijazah Berbasis Web” dapat terselesaikan dengan baik.

Dalam proses penyusunan laporan ini, penulis banyak mendapat banyak arahan serta saran dari berbagai pihak, baik itu secara langsung maupun tidak langsung. Karena itu pada kesempatan ini penulis secara khusus mengucapkan banyak terima kasih kepada pihak-pihak yang telah membantu dalam penyusunan laporan Kerja Praktik ini khususnya:

1. Prof. Dr. Johanis Ohoitumur selaku Rektor Universitas Katolik De La Salle Manado.
2. Bapak Ronald Albert Rachmadi, S.T., M.T. selaku Dekan Fakultas Teknik Universitas Katolik De La Salle Manado.
3. Ibu Vivie D. Kumenap, S.T., M.Cs. selaku Ketua Program Studi Teknik Informatika Fakultas Teknik Universitas Katolik De La Salle Manado dan Dosen Pembimbing II Kerja Praktik yang telah membantu dan memberikan arahan serta saran selama pembuatan laporan dan aplikasi.
4. Ibu Lanny Sitanayah, S.T M.Sc., Ph.D. selaku Dosen Pembimbing Akademik dan Dosen Pembimbing I Kerja Praktik yang telah membantu dan memberikan arahan serta saran selama pembuatan laporan dan aplikasi.
5. Keluarga, saudara, dan teman-teman yang telah senantiasa mendoakan dan memberikan dukungan kepada penulis selama pembuatan tugas Kerja Praktik.

Penulis menyadari bahwa dalam pembuatan laporan kerja praktik ini penulis tidak terlepas dari kekurangan, oleh karena itu kritik dan saran dari para pembaca sangat diharapkan.

Manado, 18 Desember 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
FORMULIR DATA UMUM PERUSAHAAN	iv
FORMULIR PENILAIAN KEMAJUAN KERJA PRAKTIK	v
FORMULIR PENILAIAN PELAKSANAAN KERJA PRAKTIK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Tujuan Kerja Praktik	2
1.4 Manfaat Kerja Praktik	2
1.5 Batasan Masalah.....	2
1.6 Sistematika Penulisan.....	3
BAB II DATA UMUM PERUSAHAAN	4
2.1 Sejarah Singkat Perusahaan.....	4
2.2 Lingkup Pekerjaan Perusahaan.....	5
2.2.1 Aktivitas Bisnis Perusahaan.....	5
2.2.2 Visi dan Misi Perusahaan	5
2.2.3 Logo Perusahaan.....	6
2.2.4 Struktur Organisasi	7
2.2.5 Tugas Pokok dan Fungsi Bagian	8
2.3 Lingkup Pekerjaan yang Dilakukan	8
BAB III LANDASAN TEORI.....	9
3.1 Teori Pendukung	9
3.1.1 Surat Keterangan Pendamping Ijazah.....	9
3.1.1 <i>Hypertext Markup Language</i>	9

3.1.2 PHP	10
3.1.3 <i>JavaScript</i>	11
3.1.4 CSS	12
3.1.5 <i>Bootstrap</i>	12
3.1.6 <i>Laravel</i>	13
3.1.7 <i>Database</i>	13
3.2 Metodologi Pengembangan Aplikasi	15
3.2.1 <i>Unified Modeling Language</i>	18
3.3 Prosedur Pengumpulan Data	22
BAB IV PEMBAHASAN.....	24
4.1 Perencanaan.....	24
4.2 Analisis	24
4.2.1 Wawancara.....	24
4.2.2 Pembahasan Hasil Wawancara	24
4.2.3 Mengidentifikasi dan Menganalisis Masalah	25
4.2.4 Ruang Lingkup Proyek	26
4.2.5 <i>System Requirements</i>	27
4.3 Desain	27
4.3.1 Pemodelan Sistem.....	28
4.3.2 Membuat Desain <i>Prototype</i> Antarmuka	37
4.3.3 Revisi Desain <i>Prototype</i> antarmuka	42
4.4. Implementasi	44
4.4.1 Implementasi Basis Data	44
4.4.2 Implementasi Pemrograman	46
4.4.3 Implementasi Aplikasi Serta Fungsinya	51
4.4.4 Pengujian Aplikasi	59
BAB V KESIMPULAN DAN SARAN.....	63
5.1 Kesimpulan.....	63
5.2 Saran.....	63
DAFTAR PUSTAKA	64
LAMPIRAN	

DAFTAR TABEL

Tabel 3.1 Contoh Tag HTML	10
Tabel 3.2 Use Case Diagram.....	19
Tabel 3.3 Activity Diagram.....	20
Tabel 3.4 Class Diagram	21
Tabel 4.1 Identifikasi Masalah.....	25
Tabel 4.2 Identifikasi Sumber Daya.....	27
Tabel 4.3 Use Case #1 Masuk.....	29
Tabel 4.4 Use Case #2 Menambahkan Data SKPI.....	29
Tabel 4.5 Use Case #3 Mengubah Data SKPI	30
Tabel 4.6 Use Case #4 Menambah Akun Mahasiswa.....	30
Tabel 4.7 Use Case #5 Mengubah Akun.....	31
Tabel 4.8 Use Case #6 Memeriksa Data SKPI Mahasiswa	31
Tabel 4.9 Use Case #7 Mengubah Data SKPI Mahasiswa	32
Tabel 4.10 Use Case #8 Menghapus Data SKPI Mahasiswa.....	32
Tabel 4.11 Use Case #9 Mencetak SKPI	33
Tabel 4.12 Use Case #10 Menyimpan SKPI.....	33
Tabel 4.13 Use Case #11 Keluar.....	34
Tabel 4.14 Hasil Pengujian Black Box	60
Tabel 4.15 Hasil Beta Testing.....	61

DAFTAR GAMBAR

Gambar 2.1 Logo Universitas Katolik De La Salle Manado [2].....	6
Gambar 2.2 Struktur Organisasi Fakultas Teknik.....	8
Gambar 3.1 Pengembangan Bertahap [15].....	16
Gambar 3.2 Prototyping [15]	17
Gambar 3.3 Throwaway Prototyping [15]	17
Gambar 4.1 Use Case Diagram Aplikasi SKPI.....	28
Gambar 4.2 Activity Diagram Mahasiswa.....	35
Gambar 4.3 Activity Diagram Admin.....	36
Gambar 4.4 Class Diagram Aplikasi SKPI	37
Gambar 4.5 Halaman Memasukkan Data	38
Gambar 4.6 Halaman Ubah Data	38
Gambar 4.7 Halaman Ganti Kata Sandi	39
Gambar 4.8 Halaman Dashboard	39
Gambar 4.9 Halaman Tambah Pengguna.....	40
Gambar 4.10 Halaman Detail Data SKPI Pengguna.....	41
Gambar 4.11 Halaman Ubah Data Pengguna	41
Gambar 4.12 Halaman Masuk.....	42
Gambar 4.13 Halaman Ganti Kata Sandi	43
Gambar 4.14 Halaman Dashboard	43
Gambar 4.15 Halaman Masuk.....	44
Gambar 4.16 Tabel Users.....	45
Gambar 4.17 Tabel Prestasi	45
Gambar 4.18 Tabel Pesan	45
Gambar 4.19 Tabel Informasi Tambahan	46
Gambar 4.20 Tabel Biodata	46
Gambar 4.21 Halaman Masuk.....	51
Gambar 4.22 Halaman Dashboard Admin	52
Gambar 4.23 Halaman Tambah Mahasiswa	52
Gambar 4.24 Halaman Ubah Pengguna	53
Gambar 4.25 Halaman Detail Data SKPI Pengguna.....	53
Gambar 4.26 Halaman Ubah Data SKPI.....	54
Gambar 4.27 Fungsi Tambah Pesan.....	54
Gambar 4.28 Fungsi Pencarian	55
Gambar 4.29 Fungsi Pengurutan Nama	55
Gambar 4.30 Fungsi pengurutan Nama Pengguna.....	56
Gambar 4.31 Fungsi Cetak.....	56
Gambar 4.32 Halaman Ganti Kata Sandi	57
Gambar 4.33 Halaman Utama.....	57
Gambar 4.34 Fungsi Tambah Data	58
Gambar 4.35 Fungsi Ubah Data.....	58
Gambar 4.36 Halaman Ganti Kata Sandi	59

DAFTAR LAMPIRAN

LAMPIRAN A WAWANCARA.....	A-1
LAMPIRAN B <i>USER ACCEPTANCE TEST</i>	B-1

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada tahun 2014, Menteri Pendidikan dan Kebudayaan Republik Indonesia telah mengeluarkan Peraturan Menteri Pendidikan dan Kebudayaan (Permendikbud) tentang ijazah, sertifikat kompetensi, dan sertifikat profesi pendidikan tinggi. Dalam Permendikbud tersebut juga secara khusus mengatur tentang pemberian Surat Keterangan Pendamping Ijazah yang mewajibkan setiap perguruan tinggi untuk menerbitkannya.

Surat Keterangan Pendamping Ijazah (SKPI) yang disebut juga sebagai *Diploma Supplement* merupakan sebuah surat pernyataan resmi yang dikeluarkan oleh perguruan tinggi untuk setiap lulusannya yang berisikan informasi tentang pencapaian akademik serta kualifikasi yang menyatakan capaian pembelajaran lulusan tersebut. SKPI dikeluarkan untuk mendampingi ijazah dan transkrip akademik.

Di dalam SKPI terdapat berbagai informasi tentang mahasiswa dan universitas, yaitu informasi tentang identitas diri pemegang SKPI, informasi tentang identitas penyelenggara program, dan informasi tentang kualifikasi dan hasil yang dicapai. Pada Fakultas Teknik Universitas Katolik De La Salle Manado, prosedur pembuatan SKPI masih dilakukan dengan cara mahasiswa harus datang langsung ke program studi, untuk mengumpulkan dokumen-dokumen terkait berupa piagam atau sertifikat di bidang akademik maupun non akademik. Kemudian staf program studi akan memasukkan data ke dalam format SKPI yang telah tersedia untuk dicetak. Terdapat cukup banyak data yang tercantum dalam sebuah SKPI meliputi nama lengkap, tempat tanggal lahir, nomor induk mahasiswa, nama perguruan tinggi, nama program studi, jenis pendidikan, jenjang pendidikan, capaian pembelajaran, dan sebagainya. Berdasarkan kondisi tersebut maka dibutuhkan sebuah aplikasi yang mampu untuk menampung dan mengelola data yang dibutuhkan sehingga menghasilkan informasi yang akan digunakan untuk pembuatan SKPI. Oleh karena itu, atas dasar permasalahan yang disebutkan, maka akan dibuat sebuah Aplikasi Pembuatan Surat Keterangan Pendamping Ijazah

Berbasis Web dengan tujuan untuk memudahkan dan mempercepat seluruh proses pembuatan SKPI mulai dari *input* data sampai menghasilkan *output* berupa Surat Keterangan Pendamping Ijazah.

1.2 Rumusan Masalah

Bagaimana membuat aplikasi Surat Keterangan Pendamping Ijazah berbasis web yang dapat membantu untuk menyimpan informasi diri pemegang SKPI, identitas penyelenggara program, kualifikasi dan hasil yang dicapai, dan juga dokumen-dokumen bukti prestasi, serta menghasilkan *output* berupa Surat Keterangan Pendamping Ijazah?

1.3 Tujuan Kerja Praktik

Tujuan dari kerja praktik adalah untuk membuat aplikasi Surat Keterangan Pendamping Ijazah berbasis web yang dapat membantu untuk menyimpan informasi diri pemegang SKPI, identitas penyelenggara program, kualifikasi dan hasil yang dicapai, dan juga dokumen-dokumen bukti prestasi, serta menghasilkan *output* berupa Surat Keterangan Pendamping Ijazah.

1.4 Manfaat Kerja Praktik

1. Bagi pengguna (mahasiswa Fakultas Teknik)
Pengguna mendapatkan kemudahan dalam menginput data yang berkaitan dengan SKPI.
2. Bagi admin (staf Fakultas Teknik)
Admin dapat melihat, mengubah, menghapus, dan melakukan validasi terhadap data yang telah dimasukkan, kemudian mencetak Surat Keterangan Pendamping Ijazah.
3. Bagi mahasiswa
Mendapatkan pengalaman serta pengetahuan dalam kerja praktik ini.

1.5 Batasan Masalah

Adapun batasan masalah untuk kerja praktik ini adalah:

1. Aplikasi yang dibuat hanya untuk Fakultas Teknik.

2. Hasil cetak SKPI adalah dalam bentuk *portrait*.

1.6 Sistematika Penulisan

Laporan kerja praktik ini akan dituliskan dalam lima bab, antara lain:

BAB I PENDAHULUAN

Berisi penjelasan tentang latar belakang, rumusan masalah, tujuan kerja praktik, manfaat kerja praktik, batasan masalah, dan sistematika penulisan.

BAB II DATA UMUM PERUSAHAAN

Berisi tentang sejarah singkat perusahaan, lingkup pekerjaan perusahaan, dan lingkup pekerjaan yang dilakukan.

BAB III LANDASAN TEORI

Berisi penjelasan tentang teori pendukung, metodologi pengembangan aplikasi, dan prosedur pengumpulan data.

BAB IV PEMBAHASAN

Berisi pembahasan mengenai pengumpulan dan pengolahan data, analisis dan pemecahan masalah, serta pembahasan metodologi yang diikuti.

BAB V KESIMPULAN DAN SARAN

Berisi penjelasan tentang kesimpulan dari hasil kerja praktik dan pembahasan, serta saran yang merupakan usulan untuk kemungkinan pengembangan lanjutan.

BAB II

DATA UMUM PERUSAHAAN

2.1 Sejarah Singkat Perusahaan

Universitas Katolik De La Salle Manado didirikan pada tanggal 7 Agustus 2000 atas dasar kepedulian Keuskupan Manado terhadap krisis moneter yang melanda Indonesia pada tahun 1997. Sebagai pendiri, Uskup Manado Mgr. Joseph Suwatan, MSC menyadari bahwa krisis yang terjadi pada masa itu dapat diatasi dengan pendirian suatu lembaga pendidikan tinggi yang berkualitas. Pada awal sebelum berdirinya Universitas Katolik De La Salle Manado, Uskup Manado Mgr. Joseph Suwatan, MSC bersama dengan Frater Armin Luistro, FSC mendirikan Yayasan Perguruan Tinggi Universitas Katolik De La Salle Manado pada tanggal 7 Agustus 1999 [1].

Pada saat didirikan, Universitas Katolik De La Salle Manado berlokasi di pusat Kota Manado dengan alamat Jalan Walanda Maramis No. 145, hingga pada bulan Oktober 2002 Universitas Katolik De La Salle Manado berpindah ke lokasi baru yang terletak di perbukitan Kombos kompleks Wenang Permai II, Kairagi I. Universitas Katolik De La Salle Manado diresmikan pada tanggal 30 November 2002 oleh Uskup Manado, Mgr. Josef Suwatan, MSC dan Gubernur Sulawesi Utara yang menjabat pada saat itu, Drs. A. J. Sondakh. Adapun Rektor pertama yang ditunjuk saat itu adalah Prof. Dr. Johanis Ohoitumur yang bertugas hingga masa bakti tahun 2003 [1].

Pada awal berdirinya Universitas Katolik De La Salle Manado terdapat enam fakultas yaitu Fakultas Teknik, Fakultas Ekonomi, Fakultas Pertanian, Fakultas Hukum, Fakultas MIPA dan Fakultas Keperawatan. Kemudian pada tahun 2009 Fakultas MIPA Program Studi Ilmu Komputer digabung menjadi satu dengan Fakultas Teknik Program Studi Teknik Informatika. Dan sekarang ini Fakultas Teknik telah memiliki empat Program Studi antara lain Program Studi Teknik Informatika, Program Studi Teknik Industri, Program Studi Teknik Elektro, dan Program Studi Teknik Sipil [1].

2.2 Lingkup Pekerjaan Perusahaan

Fakultas Teknik Universitas Katolik De La Salle Manado merupakan salah satu fakultas yang telah ada sejak berdirinya Universitas Katolik De La Salle Manado. Saat ini Fakultas Teknik memiliki empat Program Studi yaitu Teknik Elektro, Teknik Industri, Teknik Informatika, dan Teknik Sipil.

2.2.1 Aktivitas Bisnis Perusahaan

Fakultas Teknik Universitas Katolik De La Salle Manado memiliki lingkup pekerjaan dalam bidang penyelenggaraan pendidikan Teknik yang sesuai dengan perkembangan teknologi di Indonesia untuk menghasilkan lulusan yang unggul, berjiwa Lasallian, dan dapat terserap di pasar tenaga kerja. Juga menghasilkan penelitian yang berkualitas yang dapat diakui pada tingkat nasional maupun internasional, menghasilkan layanan dan kegiatan pengabdian kepada masyarakat khususnya di bidang Teknik. Selain itu untuk membuat visi dan misi yang mendukung kegiatan pembelajaran di Universitas Katolik De La Salle Manado.

2.2.2 Visi dan Misi Perusahaan

Visi adalah suatu gambaran yang menjadi tujuan masa depan suatu organisasi yang berisi pandangan mengenai arah perkembangan suatu organisasi. Sedangkan misi adalah suatu proses atau tahapan yang akan dilalui oleh organisasi untuk mencapai tujuan yang dimaksudkan di dalam visi tersebut.

Visi dan misi merupakan hal yang penting dalam sebuah organisasi, karena visi dan misi akan menjadi landasan dasar bagi perkembangan sebuah organisasi. Berikut ini merupakan visi dan misi Fakultas Teknik Universitas Katolik De La Salle Manado.

a. Visi

Pada tahun 2025 Fakultas Teknik menjadi fakultas yang unggul dan inovatif dalam pelaksanaan tridarma di Indonesia.

b. Misi

1. Meningkatkan kualitas pendidikan Teknik sesuai etika, profesionalisme keilmuan dan perkembangan teknologi terkini.

2. Meningkatkan kualitas dan kuantitas penelitian bertaraf nasional dan internasional.
3. Melaksanakan pengabdian kepada masyarakat dengan menerapkan IPTEK untuk kesejahteraan masyarakat.
4. Melaksanakan diseminasi hasil penelitian dan pengabdian kepada masyarakat untuk pengembangan pengajaran dan peningkatan kesejahteraan masyarakat.
5. Meningkatkan jaringan kerja sama tingkat nasional dan internasional.

2.2.3 Logo Perusahaan

Logo pada perusahaan dapat didefinisikan sebagai tanda identitas yang memiliki peranan besar terhadap perusahaan tersebut dan sebagai ciri khas yang membedakan suatu perusahaan dengan perusahaan lainnya. Dan dengan adanya sebuah logo, setiap perusahaan tertentu dapat lebih mudah dikenali dan diingat oleh orang banyak. Adapun setiap logo yang dibuat dengan baik memiliki makna dan arti pada setiap atributnya. Pada Universitas Katolik De La Salle Manado terdapat logo yang disebut juga sebagai lambang dari Universitas Katolik De La Salle Manado. Pada logo Universitas Katolik De La Salle Manado terdapat beberapa atribut dengan arti dan maknanya sendiri. Berikut ini merupakan logo dari Universitas Katolik De La Salle Manado yang dapat dilihat pada Gambar 2.1 di bawah.



Gambar 2.1 Logo Universitas Katolik De La Salle Manado [2]

Adapun arti dan makna yang terkandung dalam logo Universitas Katolik De La Salle Manado adalah sebagai berikut:

a. Lambang dan Logo

1. Universitas Katolik De La Salle Manado memiliki lambang berupa perisai yang bagian atasnya dilingkari dengan pita warna kuning yang bertuliskan UNIKA DE LA SALLE MANADO, dan pada bagian bawahnya dilingkari dengan pita kuning yang bertuliskan INDONESIA. Pada bagian ujung pita atas dan pita bawah dibatasi dengan simbol salib di kiri dan kanan.
2. Di tengah perisai yang berwarna hijau terdapat tulisan *RELIGIO MORES CULTURA*, dan juga terdapat simbol bintang dengan lima sudut yang berwarna putih yang diselingi warna biru.
3. Pada bagian tengah simbol bintang terdapat kitab terbuka berwarna kuning.

b. Arti dari warna

1. Warna kuning berarti keagungan ilmu yang dicita-citakan.
2. Warna hijau yang melatarbelakangi lambang universitas berarti sikap pengharapan yang kokoh terhadap kesuksesan tanpa mengenal rasa putus asa.
3. Warna biru berarti sikap ketabahan yang harus dipupuk untuk mencapai cita-cita.
4. Warna putih yang berarti kesucian dan ketulusan.

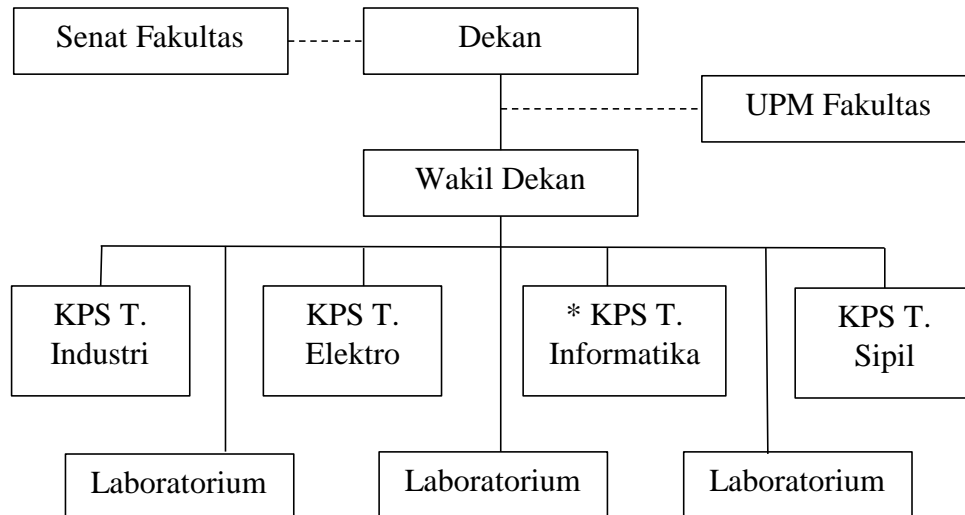
c. Arti dari simbol-simbol

1. Simbol bintang melambangkan iman akan Tuhan yang menjadi landasan ilmu yang dikejar.
2. Kitab terbuka sebagai lambang ilmu pengetahuan yang tidak kaku melainkan senantiasa terbuka dan dinamis.
3. Salib sebagai lambang pengorbanan Yesus Kristus yang hendaknya menjadi tauladan kita untuk berani berkorban.
4. Tulisan *Religio, Mores, Cultura* yang berarti Iman, Moral, dan Budaya.

2.2.4 Struktur Organisasi

Struktur organisasi dapat diartikan sebagai susunan berbagai komponen atau unit-unit kerja pada suatu organisasi. Dengan adanya struktur organisasi maka

kinerja suatu organisasi dapat lebih terarah sesuai dengan tugas dan tanggung jawab masing-masing pihak. Berikut ini adalah struktur organisasi Fakultas Teknik Universitas Katolik De La Salle Manado yang dapat dilihat pada Gambar 2.2 di bawah.



Gambar 2.2 Struktur Organisasi Fakultas Teknik

Ket. (*) adalah bagian dimana mahasiswa melakukan kerja praktik

2.2.5 Tugas Pokok dan Fungsi Bagian

Tugas pokok yang dilakukan bagian administrasi Fakultas Teknik meliputi pengarsipan surat masuk dan surat keluar, pembuatan surat pengantar fakultas, penyusunan jadwal sidang tugas akhir dan kerja praktik, penyusunan jadwal mata kuliah, serta pembuatan SKPI yang meliputi:

1. Pengumpulan dokumen-dokumen yang dibutuhkan.
2. Memasukkan data ke dalam format SKPI yang telah tersedia.
3. Mencetak SKPI.

2.3 Lingkup Pekerjaan yang Dilakukan

Lingkup pekerjaan yang penulis lakukan saat menjalani kerja praktik di Fakultas Teknik Universitas Katolik De La Salle Manado, yaitu:

1. Mengumpulkan data yang berkaitan dengan pembuatan SKPI.
2. Merancang dan membangun aplikasi SKPI berbasis web.

BAB III

LANDASAN TEORI

3.1 Teori Pendukung

3.1.1 Surat Keterangan Pendamping Ijazah

Surat Keterangan Pendamping Ijazah (SKPI) yang disebut juga sebagai *Diploma Supplement* merupakan sebuah surat pernyataan resmi yang dikeluarkan oleh perguruan tinggi untuk setiap lulusannya yang berisikan informasi tentang pencapaian akademik serta kualifikasi lulusan tersebut. SKPI dibuat dalam suatu format yang mudah dipahami, dan dalam format tersebut diuraikan mengenai capaian pembelajaran lulusan perguruan tinggi yang disusun ke dalam bentuk narasi deskriptif [3].

Dasar hukum pembuatan SKPI tertera pada Pasal 52 ayat (3) dan pada Pasal 54 ayat (1) huruf a dalam Undang-Undang Nomor 12 Tahun 2012 tentang pendidikan tinggi sebagai pemenuhan Standard Kompetensi Lulusan (SKL). Adapun manfaat-manfaatnya yaitu: sebagai dokumen tambahan yang menyatakan kemampuan kerja, dan dapat lebih mudah dimengerti oleh pihak pengguna dibandingkan dengan transkrip. Akan tetapi perlu diingat bahwa SKPI bukan media yang bisa memastikan pemegangnya mendapatkan pengakuan secara otomatis [3].

3.1.1 Hypertext Markup Language

HTML merupakan sebuah bahasa *markup* yang menjadi dasar pembuatan *website*. HTML dikategorikan sebagai bahasa *markup* yang berperan sebagai fondasi dengan menyediakan bahasa dasar untuk menandai konten teks dan atribut berbagai elemennya. Fungsi dari HTML adalah untuk menampilkan isi atau konten sebuah *website*. Secara singkat saat *browser* mengirimkan permintaan kepada web server ketika sedang mengunjungi sebuah *website*, sebagai balasannya web server merespon dengan mengembalikan isi atau konten yang diberi *markup* kepada *browser* untuk ditampilkan menjadi sebuah halaman *website* [4].

HTML ditulis dengan menggunakan *tag-tag* yang memiliki fungsinya masing-masing. Secara umum *tag-tag* dasar yang paling sering digunakan adalah:

Tabel 3.1 Contoh Tag HTML

Contoh tag	Penjelasan
<code><html> </html></code>	Digunakan sebagai <i>tag</i> pembuka dan penutup.
<code><head> </head></code>	Digunakan sebagai kepala untuk menyimpan <i>tag-tag</i> tertentu.
<code><title> </title></code>	Digunakan untuk memberikan judul pada sebuah halaman web.
<code><body> </body></code>	Digunakan sebagai tempat untuk menaruh isi dari halaman web.
<code><h1> </h1></code>	Judul tingkat satu.
<code><p> </p></code>	Sebuah paragraf.

Selain dari yang telah dituliskan di atas masih terdapat pula *tag-tag* lain dengan fungsi yang bervariasi. Adapun dalam proses pembuatan *website*, penulisan bahasa HTML harus selalu dikombinasikan dengan bahasa lainnya seperti *JavaScript* dan PHP agar mendapatkan hasil yang maksimal [5].

3.1.2 PHP

PHP atau disebut juga sebagai *Hypertext Preprocessor*, adalah sebuah bahasa pemrograman yang biasa digunakan untuk membuat atau mengembangkan halaman web. PHP berfungsi sebagai bahasa utama yang dipakai untuk membuat suatu *website* yang dinamis, karena PHP merupakan sebuah bahasa pemrograman yang terletak pada sisi server. Dalam penggunaannya PHP selalu digunakan bersamaan dengan HTML, kode-kode PHP biasanya disisipkan pada bagian dalam *tag* HTML.

Adapun beberapa fungsi dari PHP, antara lain: membuat sebuah halaman web menjadi dinamis dengan terhubung ke *database*, melakukan perintah dasar dalam pemrograman seperti *looping* dan lain-lain, melakukan perintah *input*, *update*, *delete* ke *database*, manajemen *cookie* dan *session* saat proses *login*, dan masih banyak lagi fungsi-fungsi dari PHP. Adapun contoh dari penulisan PHP adalah sebagai berikut[6] :


```
<? php
    echo "Hello World!";
?>
```

Contoh penulisan di atas menggunakan metode standar atau disebut juga XML *style*, metode penulisan inilah yang disarankan dan paling banyak dipakai. PHP juga merupakan bahasa pemrograman web yang cukup populer dikalangan *developer*, kerana menurut survey dari W3tech.com pada tahun 2019 tercatat bahwa 79% *website* di dunia ini menggunakan PHP sebagai bahasa pemrograman utama. Kepopuleran ini tentu saja didukung dengan keunggulan yang dimilikinya yaitu:

1. PHP dapat dijalankan pada sejumlah sistem operasi yang berbeda, misalnya *windows, Mac OS, Linux*, dan lain-lain.
2. PHP merupakan bahasa pemrograman yang fleksibel terutama dengan dukungannya terhadap berbagai format *file*.
3. PHP mendukung banyak jenis *database*.
4. PHP juga memiliki berbagai *library* serta *framework* yang membantu menangani banyak hal [6].

3.1.3 JavaScript

JavaScript adalah sebuah bahasa pemrograman web yang merupakan tipe bahasa pemrograman yang dapat dijalankan melalui komputer *client* dengan menggunakan aplikasi. Mayoritas situs web modern dibuat menggunakan *JavaScript* beserta dengan bahasa lainnya seperti HTML dan CSS. Selain itu *JavaScript* juga dijalankan oleh aplikasi yang disebut dengan istilah *web browser*. Adapun berbagai macam *web browser* yang sering digunakan saat ini antara lain *google chrome, mozilla firefox, internet explorer*, dan sebagainya [7].

JavaScript dibuat oleh Netscape pada masa awal perkembangan web dan pada awal pengembangannya *JavaScript* dikenal dengan nama *ECMAScript*. *Javascript* memiliki fungsi yang hampir sama dengan bahasa pemrograman lainnya yaitu fungsi dasar berupa *input* dan *output*, perhitungan aritmatika, perulangan, pengkondisian, dan lain-lain. Selain itu JavaScript juga memiliki berbagai fungsi lain, seperti membuat animasi dan visual efek, fungsi pengurutan dalam suatu tabel,

fungsi untuk menyembunyikan suatu konten dan menampilkannya kembali, dan fungsi-fungsi lainnya yang berhubungan dengan tampilan [7].

3.1.4 CSS

Cascading Style Sheet (CSS) merupakan bahasa yang digunakan untuk mengatur tampilan pada web atau disebut sebagai *style sheet language*. CSS berfungsi untuk memberikan *style* dan corak pada tampilan suatu elemen atau struktur suatu halaman web yang dibuat dengan HTML [8].

Contoh dari penggunaan CSS bisa berupa memberikan warna pada huruf, mengganti jenis huruf, mengganti ukuran huruf, memberikan *background* pada suatu halaman web, mengatur *style* paragraf, mengatur ukuran gambar, dan masih banyak lagi kegunaan lain dari CSS [8].

Dalam penggunaannya terdapat tiga cara untuk menggunakan CSS yaitu dengan cara *external*, *internal*, dan *inline*. Pada cara penggunaan *external*, *file* CSS harus disimpan secara terpisah dari file HTML yang digunakan. Nantinya dalam file HTML tersebut disisipkan *link* yang menuju pada *file* CSS yang akan digunakan. Untuk penggunaan secara *internal*, kode-kode CSS yang akan digunakan disisipkan diantara elemen `<style>` yang ditempatkan didalam tag `<head>` pada *file* HTML yang akan digunakan. Kemudian untuk cara penggunaan *inline* kode-kode CSS dituliskan secara langsung pada baris yang dimana terdapat tag HTML yang akan diberikan *style* [8].

3.1.5 Bootstrap

Bootstrap merupakan sebuah *framework* CSS yang dibuat untuk memudahkan proses desain dari suatu halaman web. Dengan adanya bootstrap penulisan kode CSS menjadi lebih mudah, karena *bootstrap* telah menyediakan berbagai komponen CSS mulai dari dasar-dasarnya sampai pada sistem *grid* yang sudah responsif sehingga desain suatu halaman web dapat diimplementasikan ke berbagai platform tanpa membatasi atau menguragi fitur-fitur pada web tersebut. Tidak hanya itu, *bootstrap* juga menyediakan banyak hal baru yang membuat desain suatu halaman web menjadi lebih menarik seperti *glyphicons*, *alerts*, *carousel*, dan lain-lain [9].

Bootstrap pada awalnya dirilis oleh Twitter dengan tujuan untuk mempertahankan konsistensi dalam desain web internal mereka. Kemudian sejak dirilis versi ketiga, *bootstrap* berubah menjadi berlisensi *open source* dan menjadi sangat populer karena telah digunakan pada banyak proyek yang ada di *Github* [9].

3.1.6 *Laravel*

Laravel merupakan sebuah *framework* dari bahasa pemrograman PHP yang dapat digunakan untuk membuat *website* ataupun aplikasi. Pada *Laravel* terdapat banyak *library* dan *functions* yang siap digunakan guna mempercepat proses pembuatan *website* maupun aplikasi. Inti dari penggunaan *Laravel* adalah untuk memudahkan *developer* dalam mengembangkan sebuah *website* maupun aplikasi. Pada *Laravel* terdapat sebuah konsep yang dikenal dengan istilah MVC (*Model, View, Controller*). MVC adalah suatu teknik atau konsep yang sudah banyak digunakan oleh *framework* PHP lainnya. Cara kerja dari konsep MVC adalah dengan memisahkan komponen utama menjadi tiga bagian dengan tujuan agar struktur *file* pada program menjadi lebih konsisten sehingga mempermudah proses pengembangan, terutama pengembangan yang dilakukan secara berkelompok dengan *developer* lain [10].

Model merupakan bagian yang berhubungan dengan *database* dalam hal mengelola dan manipulasi data, misalnya proses pengambilan data, *input* data, dan *update* data serta pengolahan *database* lainnya. *View* merupakan bagian antar muka yang ditampilkan kepada *user*. Seluruh tampilan untuk *user* dikumpulkan pada bagian ini agar terpisah dari *controller* dan *model* dengan tujuan untuk memudahkan pengembangan pada tampilan *website*. *Controller* merupakan sebuah *class* yang menghubungkan antara *model* dan *view*, dan dalam *Laravel* terdapat dua jenis *controller* yaitu *standard controller* dan *resource controller* [10].

3.1.7 *Database*

Database atau yang disebut juga basis data, merupakan suatu struktur yang digunakan untuk menyimpan data. Data yang disimpan dalam sebuah basis data dapat diproses atau dikelola menjadi informasi yang nantinya akan digunakan oleh sebuah sistem yang terhubung dengan basis data tersebut [11].

Pada basis data juga terdapat beberapa model yang diantaranya merupakan model relasional. Di dalam model ini, struktur dari basis data diubah ke dalam bentuk kumpulan dari tabel-tabel yang saling berelasi atau berhubungan. Tiap tabel ini mewakili masing-masing jenis data yang juga disebut sebagai entitas. Misalnya pada basis data sekolah jika kita membuat sebuah basis data relasional, maka kita perlu membuat tabel-tabel sesuai dengan jumlah entitas yang terlibat dan memberikan relasi diantara tabel-tabel tersebut [11].

3.1.7.1 Database Management System

Database Management System (DBMS) atau bisa disebut sistem manajemen basis data, merupakan kumpulan dari program yang bertujuan untuk menyimpan dan mengelolah basis data. Selain mengelolah, DBMS juga harus memastikan keamanan dari data yang disimpan agar terhindar dari kerusakan maupun akses yang tidak sah [12].

Di dalam DBMS terdapat dua jenis bahasa yang digunakan untuk pengelolannya. Bahasa-bahasa tersebut dikenal dengan istilah *data-definition language* (DDL) dan *data-manipulation language* (DML). Akan tetapi dalam penggunaannya kedua jenis bahasa tersebut disatukan menjadi sebuah bahasa *database*. Bahasa *database* memiliki fungsi untuk mengelolah atau memanipulasi data yang ada di dalam basis data dengan menggunakan *query*. *Query* yang digunakan terdiri dari kumpulan instruksi dengan fungsi yang bermacam-macam. Diantaranya bisa berupa fungsi untuk memasukkan data, mengubah data, menghapus data, dan juga menampilkan data [12].

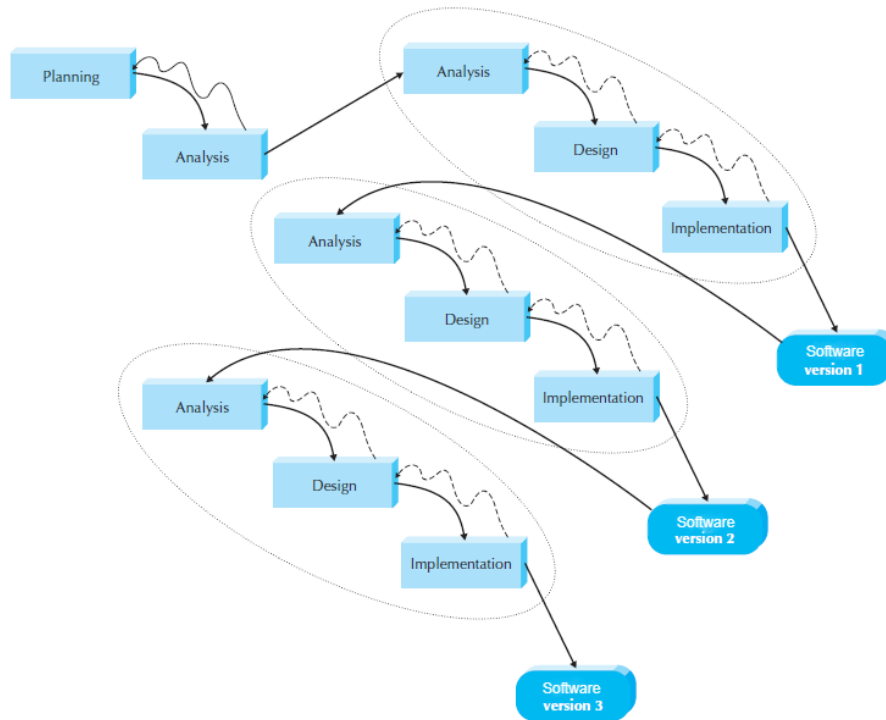
3.1.7.2 MariaDB

MariaDB adalah sebuah aplikasi pengelolaan basis data yang bisa digunakan secara gratis, hal ini menyebabkan MariaDB menjadi cukup populer di kalangan *programmer*. MariaDB juga memiliki kemampuan yang sangat baik dalam pengolahan data, bisa dijalankan pada berbagai macam perangkat keras dan juga dapat diinstal diberbagai macam sistem operasi seperti windows dan linux [13].

3.2 Metodologi Pengembangan Aplikasi

Rapid Application Development (RAD) adalah salah satu metode yang populer dalam proses pengembangan perangkat lunak. Manfaat utama dari metode RAD adalah penyelesaian proyek yang cepat, sehingga menjadikannya pilihan yang menarik bagi para *developer* yang harus bekerja dengan cepat untuk mengembangkan perangkat lunak. Kecepatan dalam metode RAD terjadi karena fokus dari metode ini dalam meminimalkan proses perencanaan dan memaksimalkan proses pengembangan prototipe. Dengan mengurangi waktu perencanaan dan memaksimalkan pengembangan prototipe, RAD memungkinkan manajer proyek dan *stakeholders* yang terlibat untuk mengukur kemajuan secara akurat dan berkomunikasi secara *real time* atas masalah atau perubahan yang terjadi. Hal ini menghasilkan efisiensi yang lebih besar, pengembangan yang lebih cepat, dan komunikasi yang efektif [14].

Metode RAD terbagi ke dalam tiga kategori yang pertama adalah metode yang berpusat pada proses, kedua berpusat pada data, dan yang ketiga berorientasi objek. Metode yang berpusat pada proses disebut juga dengan istilah pengembangan bertahap. Metode ini memecah keseluruhan proses menjadi serangkaian versi yang dikembangkan secara berurutan [15].

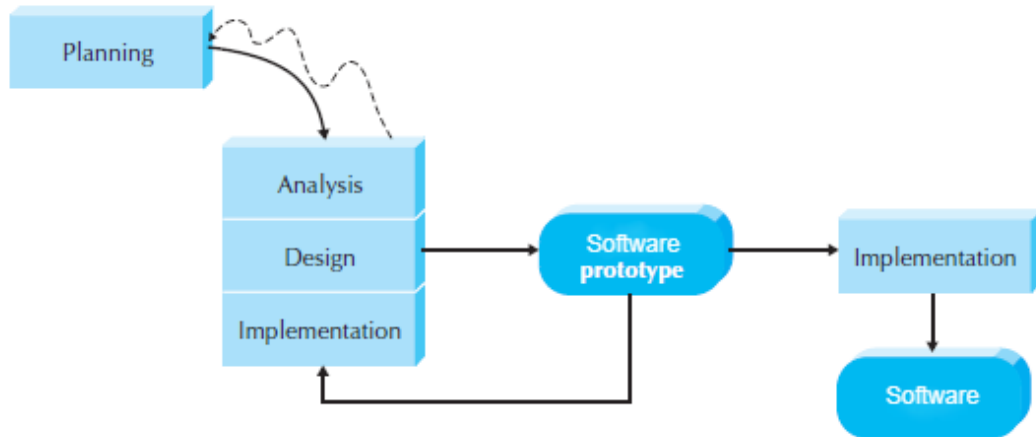


Gambar 3.1 Pengembangan Bertahap [15]

Seperti pada Gambar 3.1, metode ini dimulai dengan fase perencanaan kemudian fase analisis untuk mengidentifikasi konsep aplikasi secara keseluruhan, dan kemudian *developer* serta *stakeholders* yang terlibat mengategorikan persyaratan ke dalam serangkaian versi. Persyaratan yang paling penting dan mendasar digabungkan pada versi pertama yang terdiri atas analisis, desain, dan implementasi. Setelah itu versi pertama dari aplikasi diberikan kepada pengguna untuk diuji, dan pengguna kemudian memberikan *feedback*. Proses berlanjut ke versi kedua dengan analisis tambahan yang diperoleh dari *feedback* pengguna, setelah versi kedua selesai dirancang dan diimplementasikan diberikan kembali pada pengguna untuk diuji, proses ini berlanjut hingga semua kebutuhan pengguna terpenuhi [15].

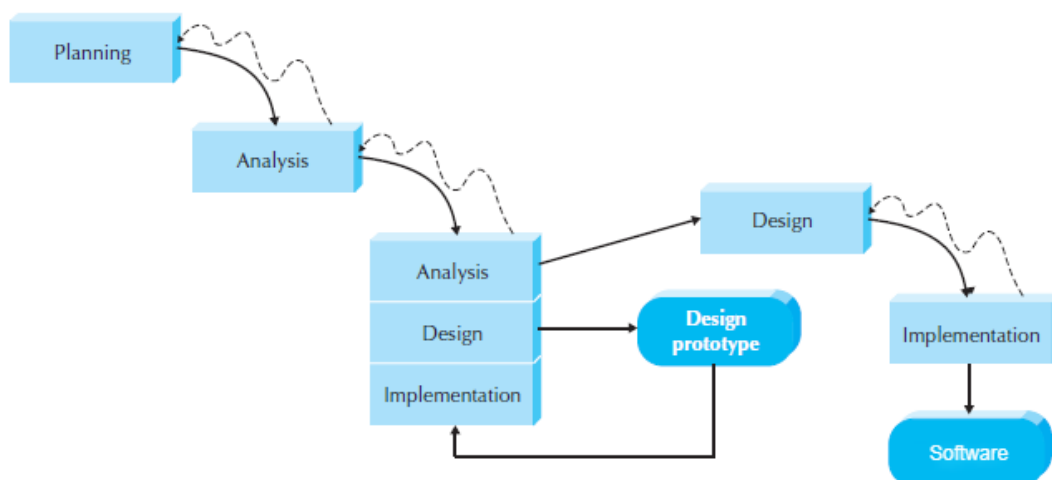
Metode kedua seperti yang ditunjukkan pada Gambar 3.2 berpusat pada data yang disebut dengan istilah *prototyping*. Pada metode ini fase analisis, desain, dan implementasi dilakukan secara bersamaan sampai menghasilkan sebuah prototipe dari aplikasi yang diinginkan. Prototipe yang sudah selesai ditunjukkan pada pengguna agar pengguna memberikan *feedback* yang akan digunakan untuk membuat ulang prototipe. Proses ini berlanjut sampai pengguna setuju dengan

prototipe yang dibuat, kemudian proses dilanjutkan dengan implementasi pada aplikasi [15].



Gambar 3.2 Prototyping [15]

Metode ketiga dapat dilihat pada Gambar 3.3 merupakan sebuah metode yang berorientasi objek dan disebut dengan istilah *Throwaway Prototyping*. Pada metode ini terdapat banyak kesamaan dengan metode sebelumnya. Hanya saja pada metode ini desain prototipe yang dirancang hanya sekali pakai, setelah itu *developer* membuat desain aplikasi yang baru dan diimplementasikan menjadi sebuah aplikasi yang sesuai dengan kebutuhan pengguna [15].



Gambar 3.3 Throwaway Prototyping [15]

3.2.1 Unified Modeling Language




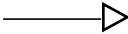


Unified Modeling Language (UML) adalah konsolidasi dari berbagai praktik terbaik yang telah ditetapkan selama bertahun-tahun dalam penggunaan bahasa pemodelan. UML memungkinkan kita untuk menyajikan berbagai aspek pada sistem perangkat lunak misalnya, persyaratan, struktur data, aliran data, dan arus informasi dalam satu kerangka kerja yang menggunakan konsep berorientasi objek [16].

UML dapat digunakan secara konsisten di seluruh proses pengembangan perangkat lunak. Pada semua tahap perkembangan, konsep bahasa yang sama dapat digunakan dalam notasi yang sama. Dengan demikian, suatu model dapat disempurnakan secara bertahap. Model tidak perlu diterjemahkan ke dalam bahasa pemodelan lain. Hal ini memungkinkan proses pengembangan perangkat lunak yang berulang dan bertahap. UML cocok untuk berbagai area aplikasi dengan persyaratan berbeda terkait kompleksitas, volume data, sistem waktu nyata, dan lain-lain. Terdapat berbagai macam diagram pada UML, namun dalam pembuatan aplikasi ini akan digunakan tiga diagram yaitu *use case diagram*, *activity diagram*, dan *class diagram* [16].

3.2.1.1 Use Case Diagram

Use case diagram memungkinkan kita untuk mendeskripsikan kemungkinan dari skenario penggunaan yang dikembangkan oleh suatu sistem. Diagram ini mengungkapkan apa yang harus dilakukan sistem tetapi tidak membahas detail realisasi apa pun seperti struktur data, algoritma, dan lain-lain. Detail ini dicakup oleh diagram lainnya seperti diagram kelas. *Use case diagram* juga memodelkan pengguna sistem sesuai dengan fungsionalitas yang digunakan, dan juga menyatakan siapa yang sebenarnya akan bekerja dengan sistem yang akan dibangun [16].

Tabel 3.2 *Use Case Diagram*



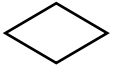
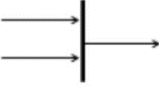

No	Gambar	Nama	Keterangan
1		<i>Actor</i> (Aktor)	Dalam <i>use case diagram</i> , aktor selalu berinteraksi dengan sistem dalam konteks <i>use case</i> mereka, yaitu <i>use case</i> yang terkait dengannya. Aktor dalam <i>use case</i> dapat berupa seseorang atau sesuatu.
2		<i>Include</i>	Merupakan sebuah relasi antara <i>use case</i> dimana <i>use case</i> tujuan harus memenuhi proses dari <i>use case</i> asal agar dapat menjalankan fungsinya.
3		<i>Extend</i>	Merupakan sebuah relasi antara <i>use case</i> dimana <i>use case</i> tujuan dapat berdiri sendiri tanpa harus memenuhi proses dari <i>use case</i> sebelumnya.
4		<i>Generalization</i> (Generalisasi)	Jika <i>use case</i> A menggeneralisasi <i>use case</i> B, B mewarisi perilaku A, yang dapat diperluas atau ditimpa oleh B. Kemudian, B juga mewarisi semua hubungan dari A. Oleh karena itu, B mengadopsi fungsionalitas dasar dari A tetapi memutuskan sendiri bagian mana dari A yang dieksekusi atau diubah.
5		<i>Association</i> (Asosiasi)	Merupakan penghubung antara aktor dan <i>use case</i> yang menandakan adanya interaksi.
6		<i>Use Case</i>	<i>Use case</i> berfungsi untuk menjelaskan fungsionalitas yang diharapkan dari sistem yang akan dikembangkan. Hal


No	Gambar	Nama	Keterangan
			ini mencakup sejumlah fungsi yang dijalankan saat menggunakan sistem.

3.2.1.2 Activity Diagram

Activity diagram berfokus pada pemodelan aspek pemrosesan prosedural dari suatu sistem. Diagram ini menentukan aliran kontrol dan aliran data antara berbagai langkah dan tindakan yang diperlukan untuk mengimplementasikan suatu aktivitas. Salah satu fitur khusus pada *activity diagram* adalah dukungannya untuk pemodelan sistem berorientasi objek dan sistem yang tidak berorientasi objek [16].

Tabel 3.3 Activity Diagram

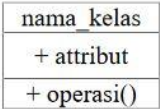

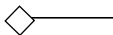
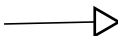
No	Gambar	Nama	Keterangan
1		<i>Initial Node</i> (Titik Mulai)	<i>Initial node</i> merupakan sebuah tempat awal dari suatu aktivitas yang akan dimulai.
2		<i>Activity</i> (Aktivitas)	<i>Activity</i> merupakan suatu aktivitas yang dilakukan sistem dan dalam penggunaannya biasanya diawali dengan kata kerja.
3		<i>Decision</i> (Keputusan /Percabangan)	<i>Decision</i> atau yang disebut juga percabangan merupakan sebuah titik atau poin dimana terdapat pilihan lebih dari satu aktivitas.
5		<i>Join</i> (Penggabungan)	<i>Join</i> atau penggabungan adalah sebuah keadaan dimana terdapat lebih dari satu aktivitas yang digabungkan menjadi satu.
7		<i>Swimlanes</i>	<i>Swimlanes</i> berfungsi memisahkan <i>activity diagram</i> menjadi baris atau kolom untuk membagi objek-objek

No	Gambar	Nama	Keterangan
			yang melakukan aktivitas sesuai dengan tanggung jawab terhadap aktivitas yang terjadi.
8		<i>Final State</i> <i>/End Point</i> (Titik Akhir)	<i>Final State</i> merupakan titik akhir yang menandakan berakhirnya sebuah aktivitas.

3.2.1.3 Class Diagram

Class diagram digunakan untuk memodelkan struktur statis dari suatu sistem, serta menggambarkan elemen-elemen pada sistem dan hubungan di antara mereka. Elemen-elemen beserta hubungan di antara mereka tidak berubah seiring waktu. Dalam konteks pemrograman berorientasi objek, diagram kelas memvisualisasikan kelas-kelas yang terdiri dari sistem perangkat lunak dan hubungan antara kelas-kelas ini [16].

Tabel 3.4 Class Diagram

No	Gambar	Nama	Keterangan
1		<i>Class</i>	Kelas adalah rencana konstruksi untuk sekumpulan objek serupa yang ada pada sebuah sistem secara spesifik.
2		<i>Asociation</i> (Asosiasi)	Asosiasi merupakan hubungan atau relasi umum antar kelas.
3		<i>Aggregation</i> (Agregasi)	Agregasi adalah bentuk asosiasi khusus yang digunakan untuk menyatakan bahwa <i>instance</i> dari satu kelas adalah bagian dari <i>instance</i> kelas lain.
5		<i>Generalization</i> (Generalisasi)	Generalisasi merupakan relasi antar kelas dari umum ke khusus.

3.3 Prosedur Pengumpulan Data

Proses pengumpulan data merupakan sebuah kegiatan untuk memperoleh informasi-informasi yang dibutuhkan dalam kerja praktik ini. Adapun prosedur pengumpulan data dilakukan dengan beberapa metode, yaitu metode observasi, metode wawancara, dan studi pustaka.

1. Metode observasi

Metode observasi adalah suatu teknik pengumpulan data dimana pihak pengamat melakukan pengamatan secara langsung terhadap hal-hal yang terjadi di lapangan, dan sekaligus juga memberikan penilaian terhadap hal-hal tersebut. Metode observasi juga membantu pihak pengamat untuk memahami gambaran dan konteks dari hal-hal yang diamati. Pada kerja praktik ini metode observasi digunakan untuk memperoleh pengetahuan terkait proses pembuatan SKPI.

2. Metode wawancara

Metode wawancara adalah suatu metode pengumpulan data yang dilakukan dengan cara mengajukan pertanyaan-pertanyaan terkait untuk dijawab oleh responden. Proses wawancara dapat dilakukan secara tatap muka langsung ataupun melalui media lainnya seperti media telepon dan media daring. pertanyaan yang diajukan biasanya merupakan pertanyaan tertulis yang bisa dijawab secara bebas oleh para responden ataupun telah diberikan opsi-opsi yang dapat dipilih oleh reponden. Pertanyaan yang diajukan merupakan pertanyaan yang telah diuji terlebih dahulu agar bisa menjadi alat ukur yang valid dan reliabel. Pada kerja praktik ini metode wawancara digunakan untuk memperoleh informasi tentang kebutuhan pengguna yang meliputi prosedur pembuatan SKPI, dan pengumpulan data yang diperlukan, serta proses *input* data ke dalam format yang telah tersedia.

3. Studi pustaka

Studi pustaka merupakan suatu kegiatan yang dilakukan dengan tujuan untuk mengumpulkan teori-teori dasar yang berkaitan dengan bidang yang sedang diteliti. Pada kerja praktik ini kegiatan studi pustaka dilakukan untuk mencari dasar ilmu serta informasi yang dibutuhkan untuk pembuatan aplikasi.

BAB IV

PEMBAHASAN

4.1 Perencanaan

Fase pertama pada metodologi RAD *prototyping* adalah perencanaan, dimana pada fase tersebut dilakukan kegiatan perencanaan bersama *client* yang membahas mengenai gambaran pada fase-fase selanjutnya yaitu analisis, desain, dan bagaimana proses implementasinya.

4.2 Analisis

Fase kedua ini merupakan fase analisis yang merupakan lanjutan dari perencanaan awal. Pada fase ini dilakukan kegiatan pengumpulan data untuk mengetahui kebutuhan pengguna, yang nantinya akan diterapkan pada prototipe dari aplikasi yang akan dibuat.

4.2.1 Wawancara

Berikut ini adalah daftar pertanyaan pada saat melakukan wawancara di Fakultas Teknik Universitas Katolik De La Salle Manado.

1. Apa fungsi dari SKPI untuk para lulusan Fakultas Teknik?
2. Bagaimana prosedur pembuatan SKPI?
3. Data apa saja yang diperlukan dalam pembuatan SKPI?

4.2.2 Pembahasan Hasil Wawancara

Berdasarkan wawancara sebelumnya diketahui bahwa SKPI berfungsi sebagai pelengkap yang dikeluarkan untuk mendampingi ijazah dan transkrip akademik. Pembuatan SKPI juga didasari oleh Permendikbud yang mewajibkan setiap perguruan tinggi untuk menerbitkan SKPI kepada setiap lulusannya.

Prosedur pembuatan SKPI dilakukan dengan cara mahasiswa harus datang terlebih dahulu ke program studi, untuk mengajukan pembuatan SKPI dengan membawa serta dokumen-dokumen terkait jika mahasiswa tersebut memilikinya. Kemudian data dari mahasiswa tersebut akan dimasukkan ke dalam format SKPI yang telah tersedia untuk kemudian dicetak.

Data yang diperlukan dalam proses pembuatan SKPI yang pertama adalah data identitas diri mahasiswa, yang kedua adalah data Universitas yang menjadi penyelenggara program pendidikan, dan yang ketiga adalah data mengenai capaian pembelajaran mahasiswa dimana juga termasuk informasi tambahan berupa prestasi akademik maupun non akademik serta pengalaman mengikuti organisasi.

4.2.3 Mengidentifikasi dan Menganalisis Masalah

Prosedur pembuatan SKPI pada Fakultas Teknik Universitas Katolik De La Salle Manado masih dilakukan dengan cara manual. Mahasiswa harus datang langsung ke program studi untuk mengumpulkan dokumen-dokumen terkait, yang nantinya akan digunakan untuk pengisian data dalam SKPI. Dokumen-dokumen terkait dapat berupa piagam atau sertifikat di bidang akademik maupun non akademik, yang kemudian akan dimasukkan datanya oleh staf program studi ke dalam format SKPI yang telah tersedia. Terdapat juga data yang lain yaitu identitas diri pemegang SKPI, dimana data tersebut berbeda-beda untuk tiap mahasiswa dan harus dimasukkan satu per satu ke dalam format SKPI untuk dicetak. Atas dasar permasalahan tersebut maka diperlukan sebuah aplikasi yang mampu untuk memudahkan dan mempersingkat prosedur pembuatan SKPI, mulai dari proses *input* data sampai menghasilkan *output* berupa Surat Keterangan Pendamping Ijazah yang telah dicetak.

Pada Tabel 4.1 berikut ini, akan dituliskan proses identifikasi dan analisis terhadap permasalahan yang telah disebutkan.

Tabel 4.1 Identifikasi Masalah

Masalah	Solusi
Pengumpulan dokumen-dokumen terkait yang masih dilakukan secara manual.	Pada aplikasi yang akan dibuat terdapat fitur dimana mahasiswa dapat memasukkan dokumen-dokumen yang terkait.
Proses <i>input</i> data identitas diri pemegang SKPI masih dilakukan secara manual untuk setiap mahasiswa.	Pada aplikasi juga terdapat fitur dimana mahasiswa dapat memasukkan data diri secara langsung yang kemudian

	akan dimunculkan secara otomatis saat SKPI dicetak.
--	---

4.2.4 Ruang Lingkup Proyek

Pada bagian ini akan dijelaskan mengenai ruang lingkup dari proyek yang akan dibuat, antara lain:

1. Melakukan penelitian terhadap masalah yang terjadi dalam proses pembuatan SKPI pada Fakultas Teknik Universitas Katolik De La Salle Manado.
2. Melakukan analisis terhadap masalah yang terjadi agar dapat menentukan solusi terhadap permasalahan tersebut.
3. Mengumpulkan data yang diperlukan dalam membangun aplikasi.
4. Membuat rancangan berupa basis data, dan desain antarmuka aplikasi yang akan dibangun.
5. Melakukan implementasi pengkodean serta pengujian terhadap aplikasi yang dibangun.

Fitur-fitur yang akan dibuat dalam aplikasi adalah sebagai berikut:

1. *Login*
Fitur ini merupakan proses awal yang dilakukan untuk masuk ke dalam aplikasi.
2. *Keluar*
Fitur ini merupakan proses untuk keluar dari aplikasi.
3. *Tambah data*
Fitur ini berfungsi untuk menambahkan data ke dalam aplikasi yaitu data mengenai identitas diri serta pencapaian prestasi yang tambahkan oleh mahasiswa, dan data akun yang ditambahkan oleh admin.
4. *Ubah data*
Fitur ini berfungsi untuk mengubah data yang telah tersimpan sebelumnya.
5. *Hapus data*
Fitur ini berfungsi untuk menghapus data.

6. Cetak SKPI

Fitur ini berfungsi untuk mencetak SKPI sesuai dengan data yang telah dimasukkan oleh mahasiswa.

7. *Import PDF*

Fitur ini berfungsi untuk membuat file PDF dari SKPI yang telah berisikan data mahasiswa.

4.2.5 System Requirements

Berikut ini merupakan spesifikasi perangkat lunak, maupun perangkat keras yang digunakan dalam membangun Aplikasi Surat Keterangan Pendamping Ijazah untuk Fakultas Teknik Universitas Katolik De La Salle Manado.

Tabel 4.2 Identifikasi Sumber Daya

No.	Sumber Daya	Spesifikasi
1.	Perangkat lunak: a. Bahasa pemrograman b. <i>Database</i> c. <i>Text editor</i> d. Pemodelan objek e. Sistem operasi	PHP 7.4.12 MariaDB 10.4.16 Visual Studio Code 1.51.1 Lucid Chart Windows 10
2.	Perangkat keras: a. <i>Processor</i> b. RAM c. <i>Hardisk</i> d. Lain-lain	AMD A4 4 GB 500 GB <i>Mouse, Keyboard, Modem, Printer</i>

4.3 Desain

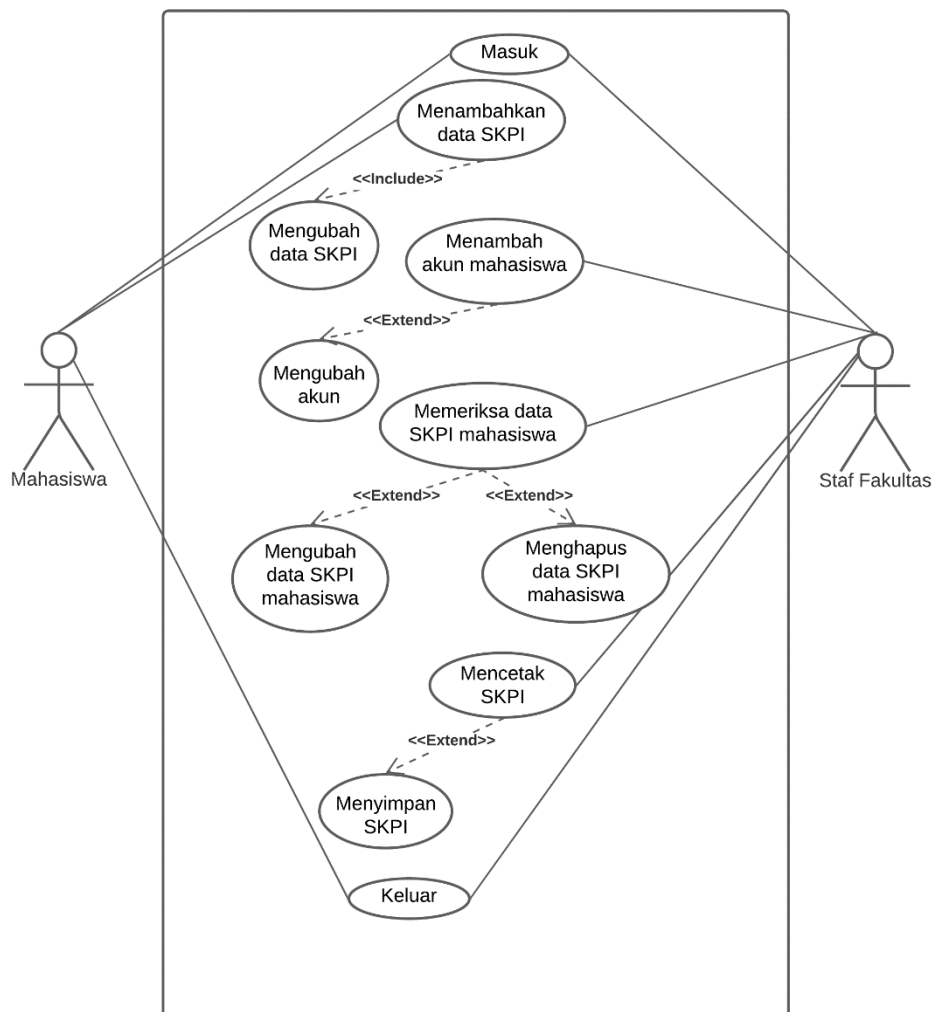
Bagian ini merupakan fase ketiga dari metode RAD *prototyping* yaitu fase desain. Dalam fase ini akan dibuat kaskas pemodelan dengan menggunakan *Unified Modeling Language* (UML) untuk memodelkan sistem dari aplikasi yang akan dibangun. Pada fase ini juga akan dibuat desain tampilan antarmuka dari aplikasi yang dibangun.

4.3.1 Pemodelan Sistem

Pada fase ini akan dibuatkan pemodelan sistem dari aplikasi yang dibangun. Pemodelan dibuat menggunakan UML yang terdiri dari *use case diagram*, *activity diagram*, dan *class diagram*.

4.3.1.1 Use Case Diagram

Berikut ini merupakan *use case diagram* dari aplikasi yang dibangun. Pada Gambar 4.1 di bawah terdapat kanvas pemodelan yang dibuat menggunakan UML dan juga terdapat tabel-tabel yang berisi penjelasan dari setiap *use case* yang digunakan.



Gambar 4.1 Use Case Diagram Aplikasi SKPI

Tabel 4.3 Use Case #1 Masuk

Nama Use Case:	#1 Masuk	
Pengguna:	Mahasiswa dan Staf Fakultas	
Deskripsi:	Melakukan <i>login</i> untuk masuk ke dalam aplikasi.	
Normal Course:	Aktor 1. Mengakses aplikasi. 3. Memasukkan nama pengguna dan kata sandi.	Aplikasi 2. Aplikasi menampilkan halaman <i>login</i> . 4. Menampilkan halaman utama.
Alternate Course:	Memasukkan nama pengguna atau kata sandi yang salah.	Menampilkan kembali halaman <i>login</i> beserta pemberitahuan kesalahan pada nama pengguna atau kata sandi.
Pre-Condition:	Mengakses halaman situs.	
Post-Condition:	Masuk ke halaman utama.	
Asumsi:	-	

Tabel 4.4 Use Case #2 Menambahkan Data SKPI

Nama Use Case:	#2 Menambahkan Data SKPI	
Pengguna:	Mahasiswa	
Deskripsi:	Menambahkan data yang diperlukan dalam pembuatan SKPI.	
Normal Course:	Aktor 1. Memasukkan data yang diperlukan pada tempat yang disediakan.	Aplikasi 2. Mengirimkan data dan menampilkan pemberitahuan data telah terkirim.
Alternate Course:	Tidak memasukkan data dengan lengkap.	Menampilkan pemberitahuan untuk

		memasukkan kembali data yang diminta.
Pre-Condition:	Telah masuk ke dalam aplikasi.	
Post-Condition:	Dapat melakukan perubahan pada data SKPI dan mengirimkannya kembali.	
Asumsi:	-	

Tabel 4.5 Use Case #3 Mengubah Data SKPI

Nama Use Case:	#3 Mengubah Data SKPI	
Pengguna:	Mahasiswa	
Deskripsi:	Mengubah data yang telah dimasukkan.	
Normal Course:	Aktor 1. Mengganti data yang ingin diubah kemudian menekan tombol <i>update</i> .	Aplikasi 2. Mengirimkan dan menampilkan perubahan pada data.
Alternate Course:	Tidak memasukkan data dengan lengkap.	Menampilkan pemberitahuan untuk memasukkan kembali data yang diminta.
Pre-Condition:	Telah memasukkan data terlebih dahulu.	
Post-Condition:	Data SKPI telah berhasil diubah.	
Asumsi:	Terdapat kesalahan pada data yang dimasukkan.	

Tabel 4.6 Use Case #4 Menambah Akun Mahasiswa

Nama Use Case:	#4 Menambah Akun Mahasiswa	
Pengguna:	Staf Fakultas	
Deskripsi:	Menambahkan akun pengguna untuk mahasiswa.	
Normal Course:	Aktor 1. Menekan tombol tambah data.	Aplikasi 2. Menampilkan halaman untuk

	3. Menekan tombol simpan untuk menyimpan data akun.	memasukkan data akun mahasiswa.
Alternate Course:	Tidak memasukkan data dengan lengkap.	Menampilkan pemberitahuan untuk memasukkan kembali data yang diminta.
Pre-Condition:	Telah masuk ke dalam aplikasi.	
Post-Condition:	Dapat melakukan perubahan pada akun.	
Asumsi:	-	

Tabel 4.7 Use Case #5 Mengubah Akun

Nama Use Case:	#5 Mengubah Akun	
Pengguna:	Staf Fakultas	
Deskripsi:	Mengubah akun mahasiswa yang telah ditambahkan sebelumnya.	
Normal Course:	Aktor 1. Menekan tombol ubah untuk mengubah data.	Aplikasi 2. Menampilkan halaman untuk mengubah data akun pengguna.
Alternate Course:	Tidak memasukkan data dengan lengkap.	Menampilkan pemberitahuan untuk memasukkan kembali data yang diminta.
Pre-Condition:	Telah terdapat akun sebelumnya.	
Post-Condition:	Data pada akun berhasil diubah.	
Asumsi:	Terdapat kekeliruan pada data akun sebelumnya	

Tabel 4.8 Use Case #6 Memeriksa Data SKPI Mahasiswa

Nama Use Case:	#6 Memeriksa Data SKPI Mahasiswa
Pengguna:	Staf Fakultas

Deskripsi:	Memeriksa data yang diperlukan dalam pembuatan SKPI.	
Normal Course:	Aktor 1. Menekan tombol detail untuk melihat data SKPI mahasiswa.	Aplikasi 2. Menampilkan halaman yang berisikan data SKPI mahasiswa.
Alternate Course:	-	
Pre-Condition:	Telah terdapat data SKPI sebelumnya.	
Post-Condition:	Dapat melakukan perubahan pada data SKPI dan juga menghapusnya.	
Asumsi:	-	

Tabel 4.9 Use Case #7 Mengubah Data SKPI Mahasiswa

Nama Use Case:	#7 Mengubah Data SKPI Mahasiswa	
Pengguna:	Staf Fakultas	
Deskripsi:	Mengubah data yang diperlukan dalam pembuatan SKPI	
Normal Course:	Aktor 1. Menekan tombol <i>edit</i> untuk mengubah data.	Aplikasi 2. Menampilkan halaman untuk mengubah data yang berkaitan dengan SKPI.
Alternate Course:	-	
Pre-Condition:	Telah terdapat data yang dimasukkan.	
Post-Condition:	Data pengguna berhasil diubah.	
Asumsi:	Terdapat kesalahan pada data SKPI yang dimasukkan.	

Tabel 4.10 Use Case #8 Menghapus Data SKPI Mahasiswa

Nama Use Case:	#8 Menghapus Data SKPI Mahasiswa
Pengguna:	Staf Fakultas

Deskripsi:	Menghapus data yang diperlukan dalam pembuatan SKPI	
Normal Course:	Aktor 1. Menekan tombol hapus.	Aplikasi 2. Menghapus data SKPI mahasiswa.
Alternate Course:	-	
Pre-Condition:	Telah terdapat data yang dimasukkan.	
Post-Condition:	Data pengguna berhasil dihapus.	
Asumsi:	-	

Tabel 4.11 Use Case #9 Mencetak SKPI

Nama Use Case:	#9 Mencetak SKPI	
Pengguna:	Staf Fakultas	
Deskripsi:	Mencetak SKPI sesuai dengan data yang telah dimasukkan oleh pengguna.	
Normal Course:	Aktor 1. Menekan tombol cetak.	Aplikasi 2. Menampilkan menu <i>print preview</i> dan tombol untuk mencetak.
Alternate Course:	-	
Pre-Condition:	Data serta dokumen yang dibutuhkan telah lengkap.	
Post-Condition:	Berhasil mencetak SKPI,	
Asumsi:	Tidak terdapat kesalahan pada data yang dimasukkan.	

Tabel 4.12 Use Case #10 Menyimpan SKPI

Nama Use Case:	#10 Menyimpan SKPI	
Pengguna:	Staf Fakultas	
Deskripsi:	Menyimpan SKPI yang akan dicetak dengan format PDF.	
Normal Course:	Aktor	Aplikasi

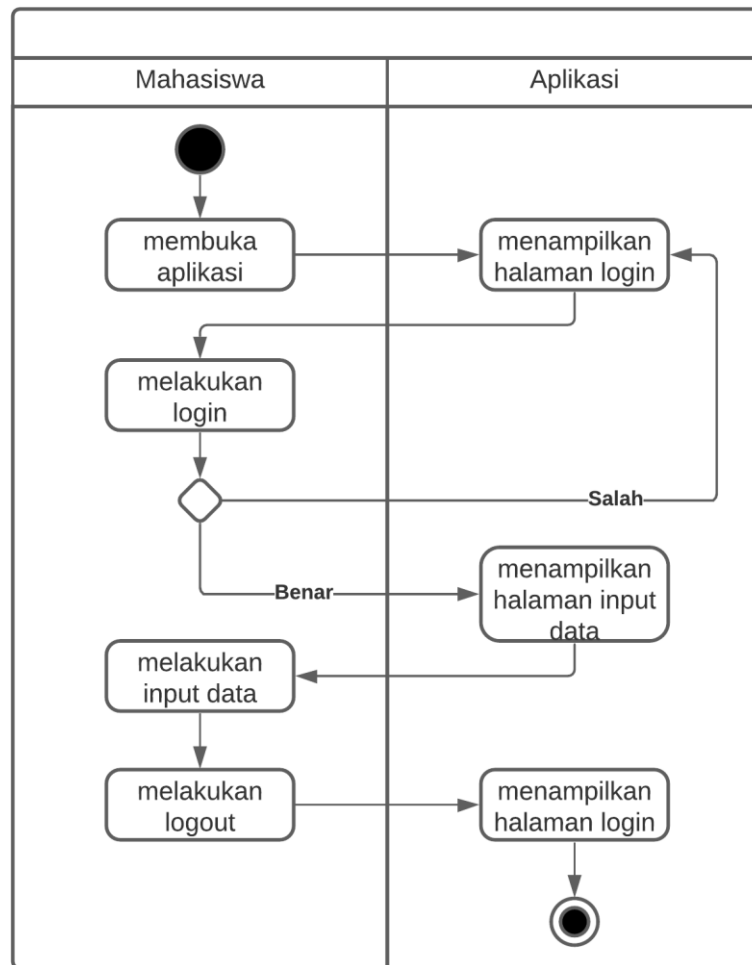
	1. Memilih pilihan <i>save as</i> PDF pada saat berada di menu <i>print preview</i> .	2. Menyimpan SKPI dengan format PDF.
Alternate Course:	-	
Pre-Condition:	Data serta dokumen yang dibutuhkan telah lengkap.	
Post-Condition:	Berhasil meyimpan SKPI kedalam format PDF.	
Asumsi:	Tidak terdapat kesalahan pada data yang dimasukkan.	

Tabel 4.13 Use Case #11 Keluar

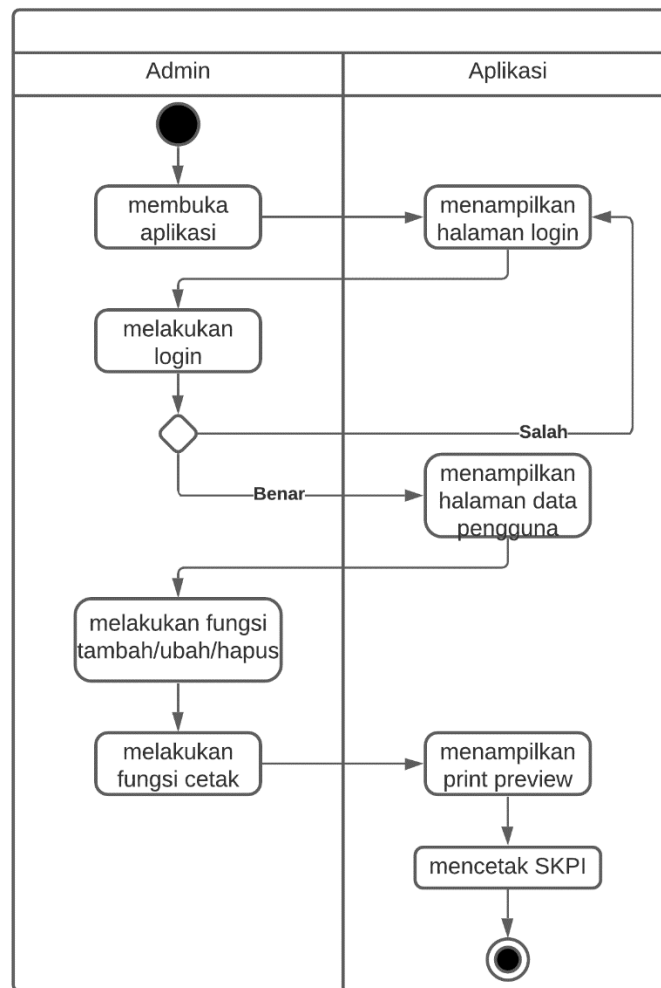
Nama Use Case:	#11 Keluar	
Pengguna:	Mahasiswa dan Staf Fakultas	
Deskripsi:	Menekan pilihan keluar untuk keluar dari aplikasi.	
Normal Course:	Aktor 1. Menekan pilihan keluar.	Aplikasi 2. Keluar dari aplikasi dan kembali ke halaman <i>login</i> .
Alternate Course:	-	
Pre-Condition:	Telah masuk ke dalam aplikasi.	
Post-Condition:	Kembali ke halaman <i>login</i> .	
Asumsi:	Telah menyelesaikan aktivitas di dalam aplikasi.	

4.3.1.2 Activity Diagram

Berikut ini merupakan *activity diagram* dari aplikasi yang akan dibuat. Pada Gambar 4.2 dan Gambar 4.3 di bawah terdapat *activity diagram* yang berisi informasi mengenai alur aktivitas pengguna saat menggunakan aplikasi.



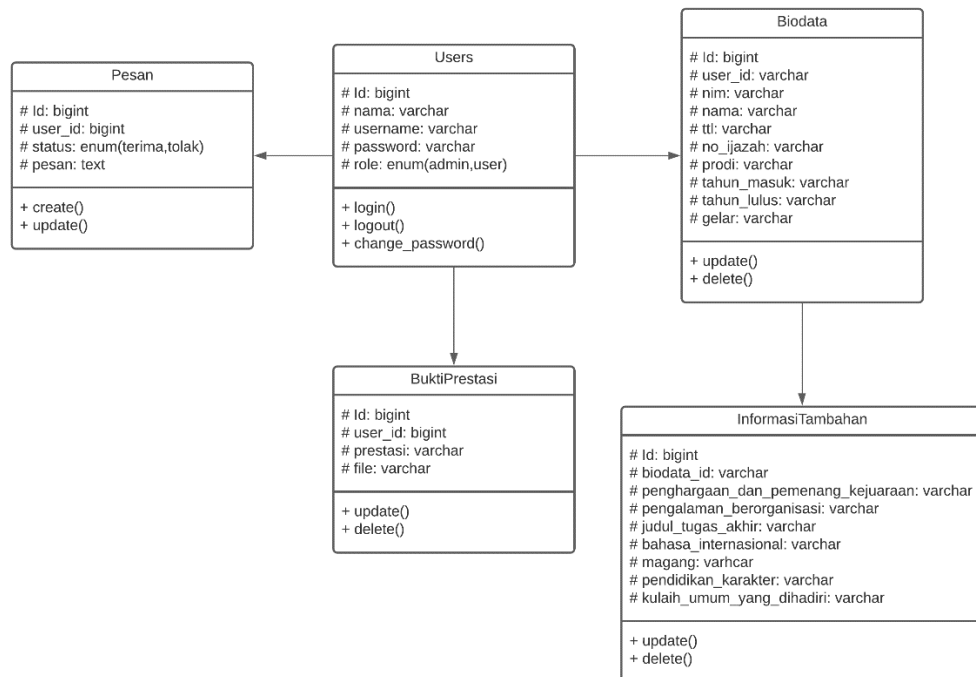
Gambar 4.2 Activity Diagram Mahasiswa



Gambar 4.3 Activity Diagram Admin

4.3.1.3 Class Diagram

Berikut ini merupakan *class diagram* dari aplikasi yang dibuat. Pada Gambar 4.4 terdapat *class diagram* yang berisi informasi visual dari struktur kelas yang ada pada aplikasi.



Gambar 4.4 Class Diagram Aplikasi SKPI

4.3.2 Membuat Desain *Prototype* Antarmuka

Pada tahap ini akan dibuat desain *prototype* untuk menggambarkan tampilan dari aplikasi. Desain *prototype* yang akan dibuat dibagi menjadi beberapa tampilan yaitu untuk mahasiswa, admin dan halaman login.

4.3.2.1 Tampilan Mahasiswa

Berikut ini merupakan tampilan yang bisa dilihat ketika *role* yang aktif adalah *user*.

1. Halaman Memasukkan Data

Gambar 4.5 Halaman Memasukkan Data

Keterangan:

1. Tempat untuk mengisi biodata.
2. Tempat untuk mengisi informasi tambahan.
3. Tempat untuk mengunggah berkas-berkas prestasi.
4. *Button* untuk mengirim seluruh data ke *database*.

2. Halaman Ubah Data

Gambar 4.6 Halaman Ubah Data

Keterangan:

1. Tempat untuk mengisi biodata.
2. Tempat untuk mengisi informasi tambahan.
3. Tempat untuk mengunggah berkas-berkas prestasi.
4. *Button* untuk memperbarui seluruh data di dalam *database*.

3. Halaman Ganti Kata sandi

Gambar 4.7 Halaman Ganti Kata Sandi

Keterangan:

1. *Text field* untuk mengisi kata sandi baru.
2. *Text field* untuk mengisi kembali kata sandi baru.
3. *Text field* untuk mengisi kata sandi lama.
4. *Button* untuk kembali ke halaman sebelumnya.
5. *Button* untuk menyimpan perubahan kata sandi.

4.3.2.2 Tampilan Admin

Berikut ini merupakan tampilan yang bisa dilihat ketika *role* yang aktif adalah admin.

1. Halaman *Dashboard*

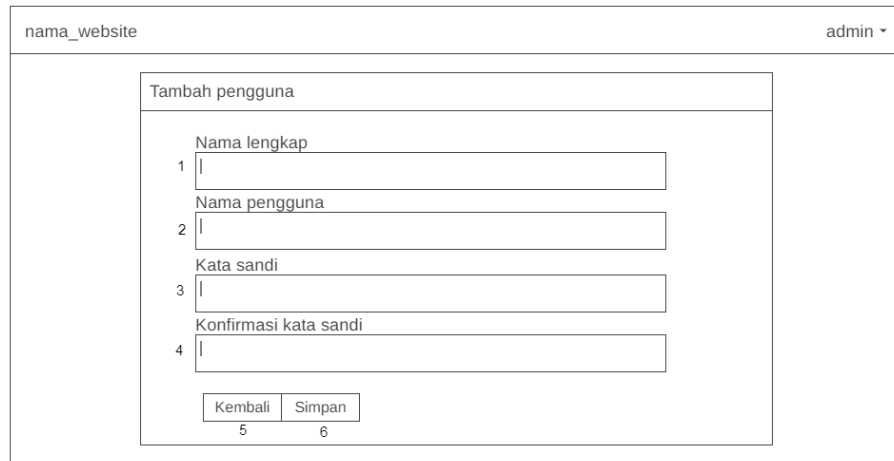
Nama	Nama pengguna	Status	Terdaftar pada	Aksi						
				<table border="1"> <tr> <td>Detail</td> <td>Cetak</td> <td>Ubah</td> </tr> <tr> <td>1</td> <td>2</td> <td>3</td> </tr> </table>	Detail	Cetak	Ubah	1	2	3
Detail	Cetak	Ubah								
1	2	3								

Gambar 4.8 Halaman *Dashboard*

Keterangan:

1. *Button* untuk masuk ke menu detail yang memuat data SKPI mahasiswa.
2. *Button* untuk mencetak SKPI.
3. *Button* untuk mengubah data akun mahasiswa.

2. Halaman Tambah Pengguna



The screenshot shows a web application interface for adding a user. The page header contains 'nama_website' on the left and 'admin' with a dropdown arrow on the right. The main content area is titled 'Tambah pengguna' and contains a form with the following fields and labels:

- 1. Nama lengkap
- 2. Nama pengguna
- 3. Kata sandi
- 4. Konfirmasi kata sandi

At the bottom of the form, there are two buttons: 'Kembali' (labeled 5) and 'Simpan' (labeled 6).

Gambar 4.9 Halaman Tambah Pengguna

Keterangan:

1. *Text field* untuk mengisi nama lengkap.
2. *Text field* untuk mengisi nama pengguna yaitu NIM.
3. *Text field* untuk mengisi kata sandi.
4. *Text field* untuk mengisi ulang kata sandi.
5. *Button* untuk kembali ke halaman sebelumnya.
6. *Button* untuk menyimpan akun pengguna.

3. Halaman Detail Data SKPI Pengguna

Gambar 4.10 Halaman Detail Data SKPI Pengguna

Keterangan:

1. Tempat untuk menampilkan biodata mahasiswa.
2. Tempat untuk menampilkan informasi tambahan.
3. Tempat untuk menampilkan bukti prestasi mahasiswa.
4. *Button* untuk kembali ke halaman sebelumnya.
5. *Button* untuk mengubah data SKPI mahasiswa.
6. *Button* untuk menghapus data SKPI mahasiswa.

4. Halaman Ubah Data Pengguna

Gambar 4.11 Halaman Ubah Data Pengguna

Keterangan:

1. *Text field* untuk mengisi nama lengkap.
2. *Text field* untuk mengisi nama pengguna yaitu NIM.

3. *Button* untuk kembali ke halaman sebelumnya.
4. *Button* untuk menyimpan perubahan pada data akun pengguna.

4.3.2.3 Tampilan *Login*

Berikut ini merupakan tampilan yang bisa dilihat ketika akan masuk ke dalam aplikasi, seperti yang ditunjukkan pada Gambar 4.12.

The image shows a login form within a window titled 'nama_website'. The form has a header 'Masuk ke aplikasi'. Below the header, there are three elements: a text input field for 'Username' (labeled 1), a text input field for 'Password' (labeled 2), and a button labeled 'Masuk' (labeled 3).

Gambar 4.12 Halaman Masuk

Keterangan:

1. *Text field* untuk mengisi nama pengguna yaitu NIM.
2. *Text field* untuk mengisi kata sandi.
3. *Button* untuk masuk ke aplikasi.

4.3.3 Revisi Desain *Prototype* antarmuka

Pada tahap ini akan dibuat kembali desain *prototype* sesuai dengan revisi dari klien. Desain *prototype* yang dibuat terbagi menjadi beberapa tampilan yaitu untuk mahasiswa, admin dan halaman login.

4.3.3.1 Tampilan Mahasiswa

Berikut ini merupakan tampilan yang bisa dilihat ketika *role* yang aktif adalah *user*.

1. Halaman Ganti Kata sandi

Gambar 4.13 Halaman Ganti Kata Sandi

Keterangan:

1. *Text field* untuk mengisi kata sandi lama.
2. *Text field* untuk mengisi kata sandi baru.
3. *Text field* untuk mengisi kembali kata sandi baru.
4. *Button* untuk kembali ke halaman sebelumnya.
5. *Button* untuk menyimpan perubahan kata sandi.

4.3.3.2 Tampilan Admin

Berikut ini merupakan tampilan yang bisa dilihat ketika *role* yang aktif adalah admin.

1. Halaman *Dashboard*

Nama	NIM	Status Biodata	Status Konfirmasi	Aksi
				Detail 1
				Cetak 2
				Ubah 3

Gambar 4.14 Halaman *Dashboard*

Keterangan:

1. *Button* untuk masuk ke menu detail yang memuat data SKPI mahasiswa.
2. *Button* untuk mencetak SKPI.
3. *Button* untuk mengubah data akun mahasiswa.

4.3.3.3 Tampilan *Login*

Berikut ini merupakan tampilan yang bisa dilihat ketika akan masuk ke dalam aplikasi, seperti yang ditunjukkan pada Gambar 4.12.

The image shows a login form within a browser window. The window title is 'nama_website'. The form is titled 'Masuk ke aplikasi'. It contains two text input fields: 'Nama pengguna' with a small vertical bar on the left and a '1' to its right; and 'Kata sandi' with a small vertical bar on the left and a '2' to its right. Below these fields is a button labeled 'Masuk' with a '3' centered below it.

Gambar 4.15 Halaman Masuk

Keterangan:

1. *Text field* untuk mengisi nama pengguna yaitu NIM.
2. *Text field* untuk mengisi kata sandi.
3. *Button* untuk masuk ke aplikasi.

4.4. Implementasi

Pada fase ini akan dibahas mengenai implementasi dari Aplikasi Surat Keterangan Pendamping Ijazah yang telah dibangun. Pembahasan dalam fase ini dibagi dalam beberapa bagian yaitu implementasi basis data, implementasi pemrograman, implementasi aplikasi serta fungsinya, dan pengujian aplikasi.

4.4.1 Implementasi Basis Data

Implementasi basis data dari aplikasi yang dibuat menggunakan *Database Management System* (DBMS) MariaDB. Adapun kode-kode yang digunakan dalam

implementasi basis data Aplikasi Surat Keterangan Pendamping Ijazah adalah sebagai berikut:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id 🔑	bigint(20)	UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	nama	varchar(255)		No	None		
<input type="checkbox"/>	3	username	varchar(255)		No	None		
<input type="checkbox"/>	4	password	varchar(255)		No	None		
<input type="checkbox"/>	5	role	enum('admin', 'user')		No	None		
<input type="checkbox"/>	6	remember_token	varchar(100)		Yes	NULL		
<input type="checkbox"/>	7	created_at	timestamp		Yes	NULL		
<input type="checkbox"/>	8	updated_at	timestamp		Yes	NULL		

Gambar 4.16 Tabel Users

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id 🔑	bigint(20)	UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	users_id 🔗	bigint(20)	UNSIGNED	No	None		
<input type="checkbox"/>	3	prestasi	varchar(255)		No	None		
<input type="checkbox"/>	4	file	varchar(255)		No	None		
<input type="checkbox"/>	5	created_at	timestamp		Yes	NULL		
<input type="checkbox"/>	6	updated_at	timestamp		Yes	NULL		

Gambar 4.17 Tabel Prestasi

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id 🔑	bigint(20)	UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	user_id 🔗	bigint(20)	UNSIGNED	No	None		
<input type="checkbox"/>	3	status	enum('terima', 'tolak')		No	None		
<input type="checkbox"/>	4	pesan	text		Yes	NULL		
<input type="checkbox"/>	5	created_at	timestamp		Yes	NULL		
<input type="checkbox"/>	6	updated_at	timestamp		Yes	NULL		

Gambar 4.18 Tabel Pesan

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 🗝️	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 biodata_id 🗝️	bigint(20)		UNSIGNED	No	None		
<input type="checkbox"/>	3 penghargaan_dan_pemenang_kejuaraan	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/>	4 pengalaman_berorganisasi	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/>	5 judul_tugas_akhir	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/>	6 bahasa_internasional	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/>	7 magang	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/>	8 pendidikan_karakter	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/>	9 kuliah_umum_yang_dihadiri	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/>	10 created_at	timestamp			Yes	NULL		
<input type="checkbox"/>	11 updated_at	timestamp			Yes	NULL		

Gambar 4.19 Tabel Informasi Tambahan

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 🗝️	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 user_id 🗝️	bigint(20)		UNSIGNED	No	None		
<input type="checkbox"/>	3 no_dokumen	varchar(255)	utf8mb4_unicode_ci		Yes	NULL		
<input type="checkbox"/>	4 tgl_dokumen	date			Yes	NULL		
<input type="checkbox"/>	5 nim	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	6 nama	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	7 ttl	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	8 no_ijazah	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	9 prodi	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	10 tahun_masuk	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	11 tahun_lulus	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	12 gelar	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	13 created_at	timestamp			Yes	NULL		
<input type="checkbox"/>	14 updated_at	timestamp			Yes	NULL		

Gambar 4.20 Tabel Biodata

4.4.2 Implementasi Pemrograman

Implementasi pemrograman dilakukan dengan menggunakan bahasa pemrograman PHP dan *framework Laravel*. Adapun kode-kode yang digunakan untuk implementasi pemrograman Aplikasi Surat Keterangan Pendamping Ijazah adalah sebagai berikut:

1. APP_NAME=Laravel
2. APP_ENV=local
3. APP_KEY=base64:YK8aZjRRT1f6NlwiJ0ZdkN1NetkH1QbUSW003PyXU+I=
4. APP_DEBUG=true
5. APP_URL=http://skpi_fatek.test
- 6.

```

7. LOG_CHANNEL=stack
8. LOG_LEVEL=debug
9.
10. DB_CONNECTION=mysql
11. DB_HOST=127.0.0.1
12. DB_PORT=3306
13. DB_DATABASE=fatek_skpi
14. DB_USERNAME=root
15. DB_PASSWORD=
16.
17. BROADCAST_DRIVER=log
18. CACHE_DRIVER=file
19. QUEUE_CONNECTION=sync
20. SESSION_DRIVER=file
21. SESSION_LIFETIME=120
22.
23. REDIS_HOST=127.0.0.1
24. REDIS_PASSWORD=null
25. REDIS_PORT=6379
26.
27. MAIL_MAILER=smtp
28. MAIL_HOST=smtp.mailtrap.io
29. MAIL_PORT=2525
30. MAIL_USERNAME=null
31. MAIL_PASSWORD=null
32. MAIL_ENCRYPTION=null
33. MAIL_FROM_ADDRESS=null
34. MAIL_FROM_NAME="${APP_NAME}"
35.
36. AWS_ACCESS_KEY_ID=
37. AWS_SECRET_ACCESS_KEY=
38. AWS_DEFAULT_REGION=us-east-1
39. AWS_BUCKET=
40.
41. PUSHER_APP_ID=
42. PUSHER_APP_KEY=
43. PUSHER_APP_SECRET=
44. PUSHER_APP_CLUSTER=mt1
45.
46. MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
47. MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"

```

Kode program di atas merupakan kode untuk mengatur koneksi ke *database* serta konfigurasi lainnya.

```

1. <?php
2.
3. namespace App\Models;
4.
5. use App\Models\Biodata;
6. use App\Models\BuktiPrestasi;
7. use Illuminate\Notifications\Notifiable;
8. use Illuminate\Contracts\Auth\MustVerifyEmail;
9. use Illuminate\Database\Eloquent\Factories\HasFactory;
10. use Illuminate\Foundation\Auth\User as Authenticatable;
11.
12. class User extends Authenticatable
13. {
14.     use HasFactory, Notifiable;

```

```

15.
16.  /**
17.   * The attributes that are mass assignable.
18.   *
19.   * @var array
20.   */
21.  protected $fillable = [
22.      'nama',
23.      'username',
24.      'password',
25.      'role'
26.  ];
27.
28.  protected $dates = [
29.      'created_at'
30.  ];
31.
32.  /**
33.   * The attributes that should be hidden for arrays.
34.   *
35.   * @var array
36.   */
37.  protected $hidden = [
38.      'password',
39.      'remember_token',
40.  ];
41.
42.  public function biodata()
43.  {
44.      return $this->hasOne(Biodata::class);
45.  }
46.
47.  public function buktiPrestasi()
48.  {
49.      return $this->hasMany(BuktiPrestasi::class, 'users_id');
50.  }
51. }

```

Kode program di atas merupakan model dari *user* yang berhubungan langsung dengan *database*, dimana dalam *framework Laravel* digunakan konsep MVC sebagai dasar arsitektur dari *framework* tersebut.

```

1. <?php
2.
3. namespace App\Models;
4.
5. use App\Models\User;
6. use App\Models\BuktiPrestasi;
7. use App\Models\InformasiTambah;
8. use Illuminate\Database\Eloquent\Model;
9. use Illuminate\Database\Eloquent\Factories\HasFactory;
10.
11. class Biodata extends Model
12. {
13.     use HasFactory;
14.
15.     protected $table = 'biodata';
16.     protected $fillable = [

```

```

17.     'user_id',
18.     'nim',
19.     'nama',
20.     'ttl',
21.     'no_ijazah',
22.     'prodi',
23.     'tahun_masuk',
24.     'tahun_lulus',
25.     'gelar'
26. ];
27.
28. public function user()
29. {
30.     return $this->belongsTo(User::class, 'user_id');
31. }
32.
33. public function informasiTambah()
34. {
35.     return $this->hasOne(InformasiTambah::class);
36. }
37.
38. public function buktiPrestasi()
39. {
40.     return $this->hasMany(BuktiPrestasi::class);
41. }
42. }

```

Kode program di atas merupakan model dari biodata yang berhubungan langsung dengan *database*.

```

1. <?php
2.
3. namespace App\Models;
4.
5. use App\Models\Biodata;
6. use Illuminate\Database\Eloquent\Model;
7. use Illuminate\Database\Eloquent\Factories\HasFactory;
8.
9. class InformasiTambah extends Model
10. {
11.     use HasFactory;
12.
13.     protected $table = 'informasi_tambahan';
14.     protected $fillable = [
15.         'biodata_id',
16.         'penghargaan_dan_pemenang_kejuaraan',
17.         'pengalaman_berorganisasi',
18.         'judul_tugas_akhir',
19.         'bahasa_internasional',
20.         'magang',
21.         'pendidikan_karakter',
22.         'kuliah_umum_yang_dihadiri'
23.     ];
24.
25.     public function biodata()
26.     {
27.         return $this->belongsTo(Biodata::class, 'biodata_id');
28.     }
29. }

```

Kode program di atas merupakan model dari informasi tambahan yang berhubungan langsung dengan *database*.

```

1. <?php
2.
3. namespace App\Models;
4.
5. use App\Models\User;
6. use Illuminate\Database\Eloquent\Model;
7. use Illuminate\Database\Eloquent\Factories\HasFactory;
8.
9. class BuktiPrestasi extends Model
10. {
11.     use HasFactory;
12.
13.     protected $table = 'bukti_prestasi';
14.     protected $fillable = [
15.         'users_id',
16.         'prestasi',
17.         'file'
18.     ];
19.
20.     public function users()
21.     {
22.         return $this->belongsTo(User::class, 'users_id');
23.     }
24. }

```

Kode program di atas merupakan model dari bukti prestasi yang berhubungan langsung dengan *database*.

```

1. <?php
2.
3. namespace App\Models;
4.
5. use App\Models\User;
6. use Illuminate\Database\Eloquent\Model;
7. use Illuminate\Database\Eloquent\Factories\HasFactory;
8.
9. class Pesan extends Model
10. {
11.     use HasFactory;
12.
13.     protected $table = 'pesan';
14.     protected $fillable = [
15.         'user_id',
16.         'status',
17.         'pesan'
18.     ];
19.
20.     public function user()
21.     {
22.         return $this->belongsTo(User::class, 'user_id');
23.     }

```

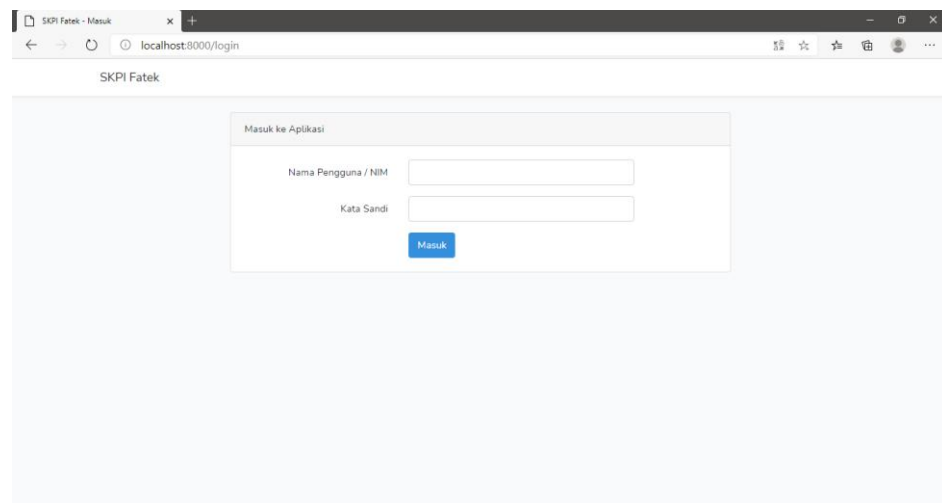

24. }

Kode program di atas merupakan model dari pesan yang berhubungan langsung dengan *database*.

4.4.3 Implementasi Aplikasi Serta Fungsinya

Setelah melakukan implementasi pemrograman, tahapan berikutnya adalah merealisasikan kode-kode bahasa pemrograman agar dapat menghasilkan suatu aplikasi yang nyata dan dapat digunakan. Berikut ini merupakan hasil implementasi dari pemrograman sehingga menghasilkan aplikasi beserta fungsi-fungsinya yang sudah dapat dijalankan.

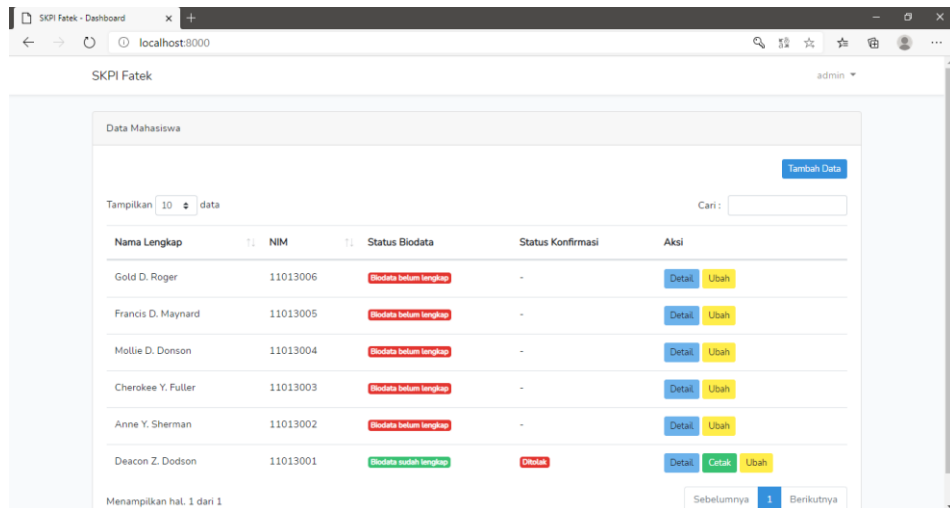
1. Halaman Masuk



Gambar 4.21 Halaman Masuk

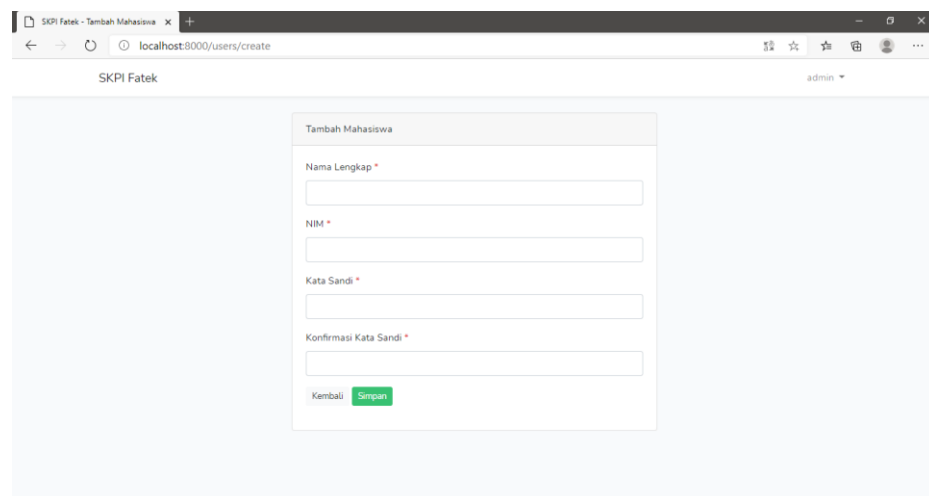
Halaman ini merupakan halaman yang pertama kali terlihat ketika admin atau pengguna mengakses aplikasi. Fungsi yang dijalankan pada halaman adalah fungsi *login* dan fungsi untuk mengecek *role* antara admin dan pengguna.

2. Halaman Admin



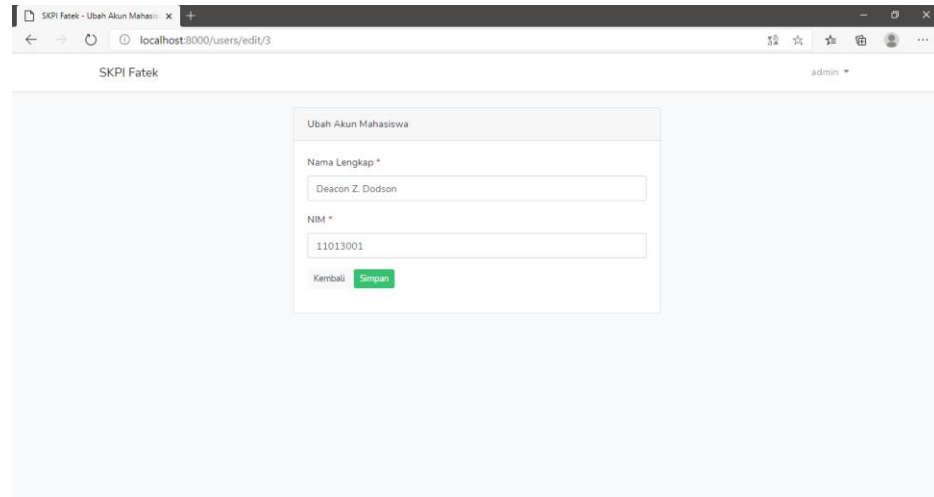
Gambar 4.22 Halaman *Dashboard* Admin

Halaman ini merupakan halaman utama admin yang tampil setelah melakukan proses *login*.



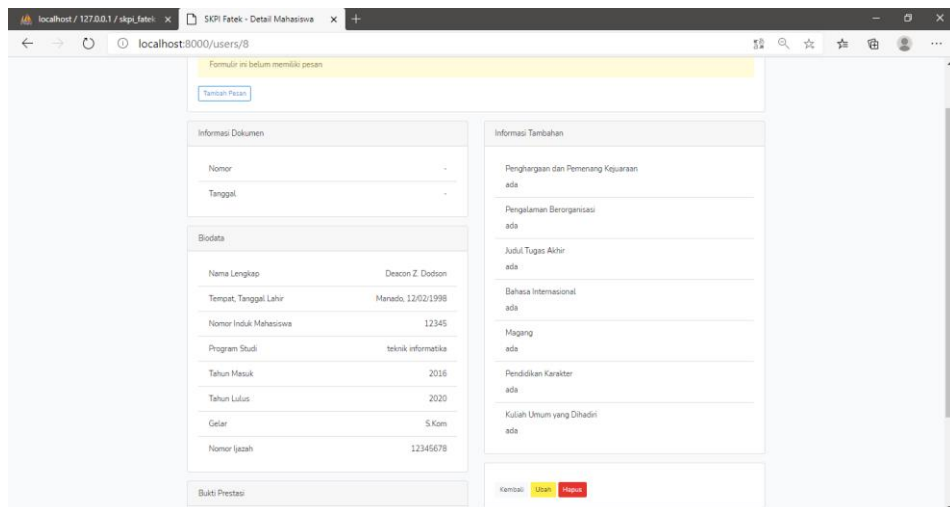
Gambar 4.23 Halaman Tambah Mahasiswa

Halaman ini merupakan halaman untuk menambahkan akun mahasiswa, yang akan muncul saat admin menekan tombol tambah data yang ada pada halaman *dashboard*.



Gambar 4.24 Halaman Ubah Pengguna

Halaman ini menampilkan data akun mahasiswa yang akan diubah oleh admin. Halaman ini muncul saat admin menekan tombol ubah yang terdapat pada halaman *dashboard*.



Gambar 4.25 Halaman Detail Data SKPI Pengguna

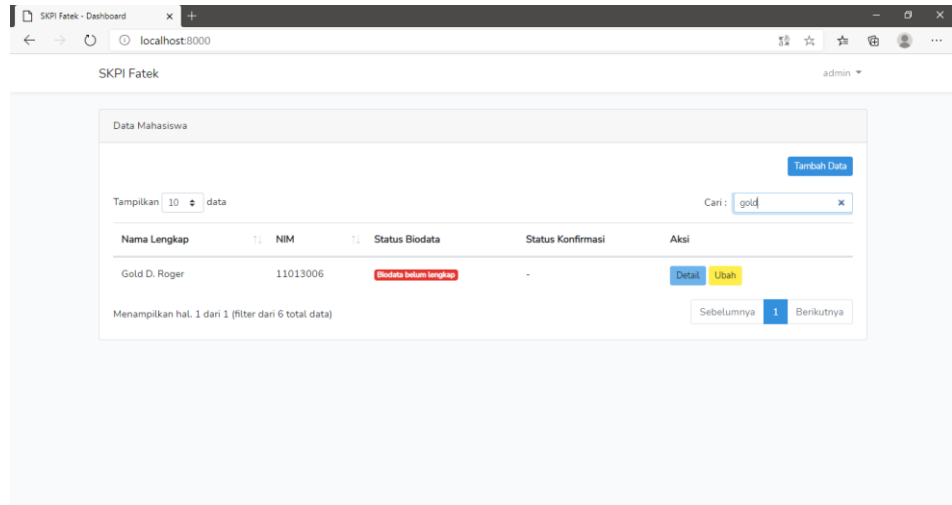
Pada halaman ini, menampilkan data SKPI yang telah dimasukkan oleh mahasiswa. Halaman ini muncul ketika admin menekan tombol detail yang berada pada halaman *dashboard*, yang berisi data biodata mahasiswa, informasi tambahan yang berkaitan dengan pencapaian prestasi, dan dokumen bukti prestasi yang telah diunggah oleh mahasiswa. Pada halaman ini juga terdapat tombol ubah, dan tombol tambah pesan.

Gambar 4.26 Halaman Ubah Data SKPI

Halaman ini menampilkan data SKPI mahasiswa yang akan diubah oleh admin. Halaman ini muncul saat admin menekan tombol ubah yang terdapat pada halaman detail data SKPI pengguna.

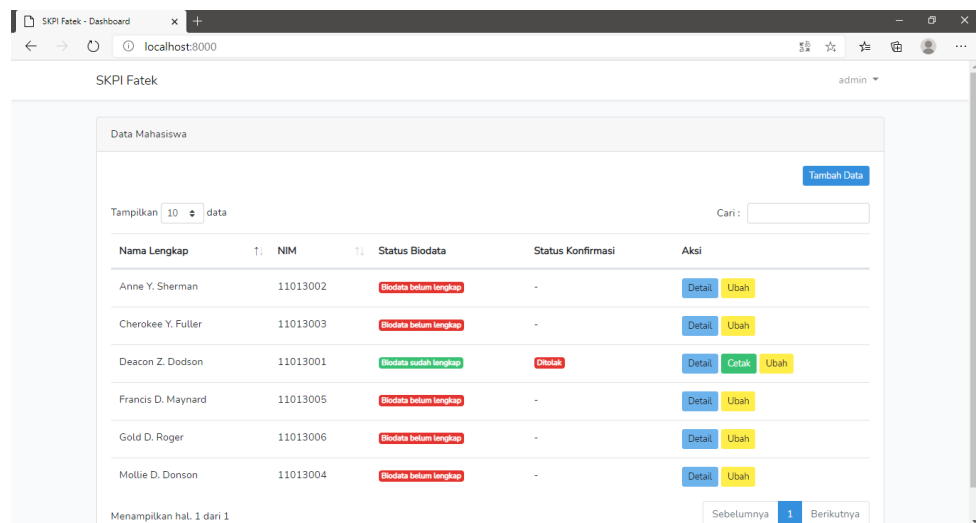
Gambar 4.27 Fungsi Tambah Pesan

Pada Gambar 4.27 di atas menunjukkan *pop up* menu, yang muncul saat admin menekan tombol tambah pesan. Setelah itu admin dapat memilih status ditolak atau diterima, dan dapat memasukkan pesan untuk dikirimkan kepada mahasiswa terkait. Saat status diterima yang dipilih maka mahasiswa sudah tidak dapat memperbarui data yang telah dimasukkan.



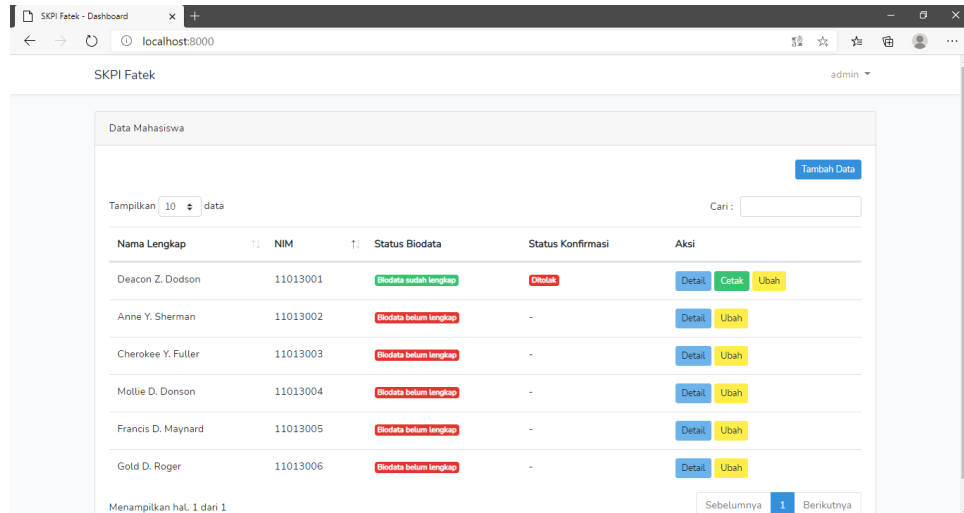
Gambar 4.28 Fungsi Pencarian

Pada Gambar 4.28 di atas menampilkan halaman *dashboard* ketika admin menggunakan fungsi pencarian berdasarkan nama.



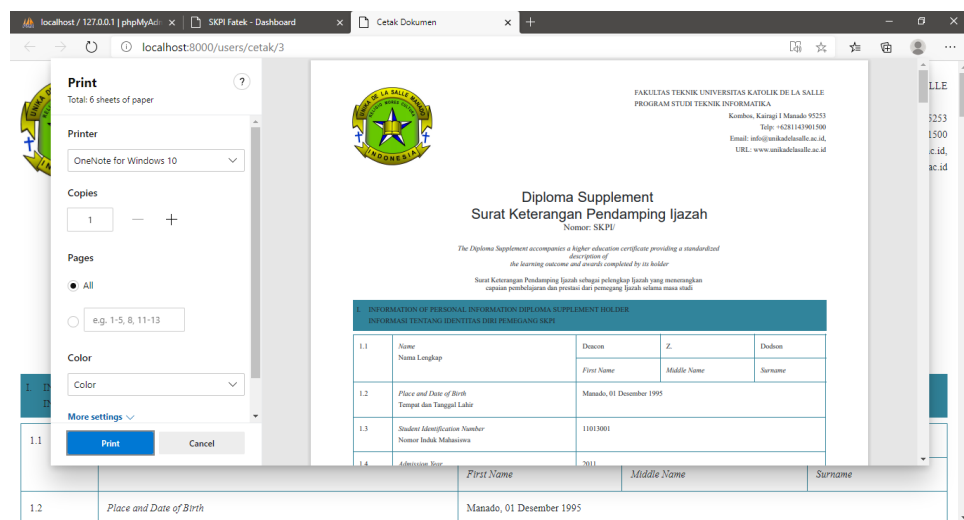
Gambar 4.29 Fungsi Pengurutan Nama

Pada Gambar 4.29 di atas menunjukkan fungsi pengurutan berdasarkan nama yang dilakukan oleh admin pada halaman *dashboard*.



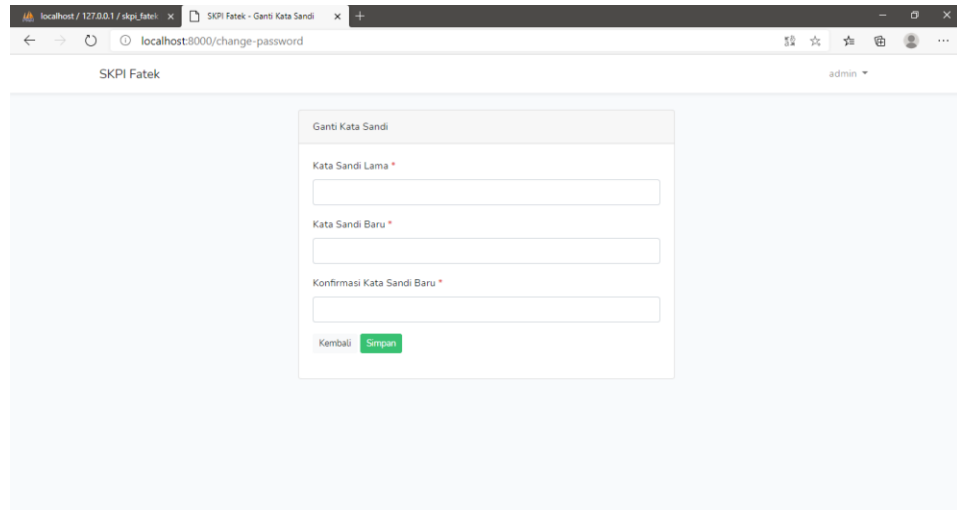
Gambar 4.30 Fungsi pengurutan Nama Pengguna

Pada Gambar 4.30 di atas menampilkan fungsi pengurutan berdasarkan nama pengguna dengan asumsi mahasiswa menggunakan nomor induk mahasiswa sebagai nama pengguna.



Gambar 4.31 Fungsi Cetak

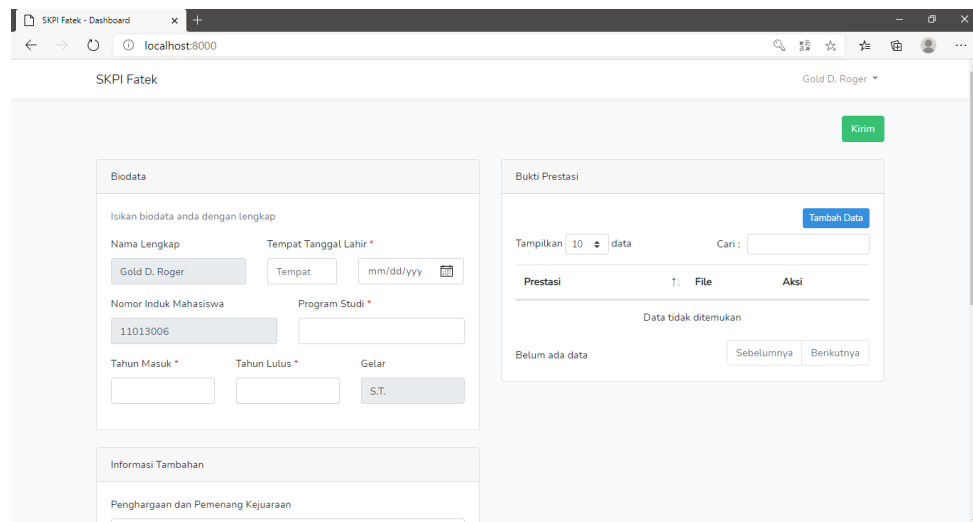
Gambar 4.31 di atas menunjukkan fungsi cetak yang terdapat pada halaman *dashboard*. Setelah mahasiswa melengkapi data SKPI beserta bukti prestasi, maka akan muncul tombol cetak pada halaman *dashboard* dan admin dapat menekan tombol tersebut untuk menjalankan fungsi cetak. Dengan asumsi admin telah memeriksa data beserta bukti prestasi yang dimasukkan oleh mahasiswa.



Gambar 4.32 Halaman Ganti Kata Sandi

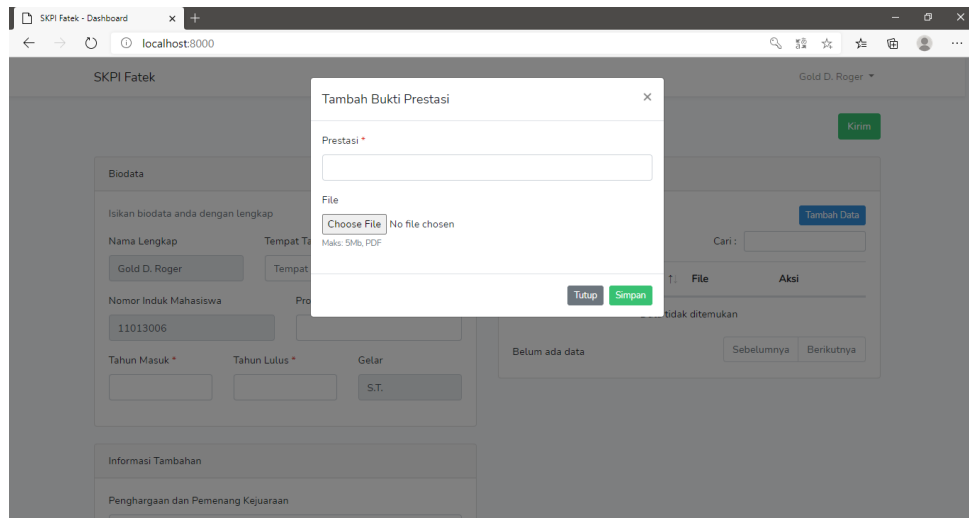
Halaman ini menampilkan menu ganti kata sandi, yang akan muncul saat admin menekan tombol *dropdown* yang terletak di sudut kanan atas.

3. Halaman Mahasiswa



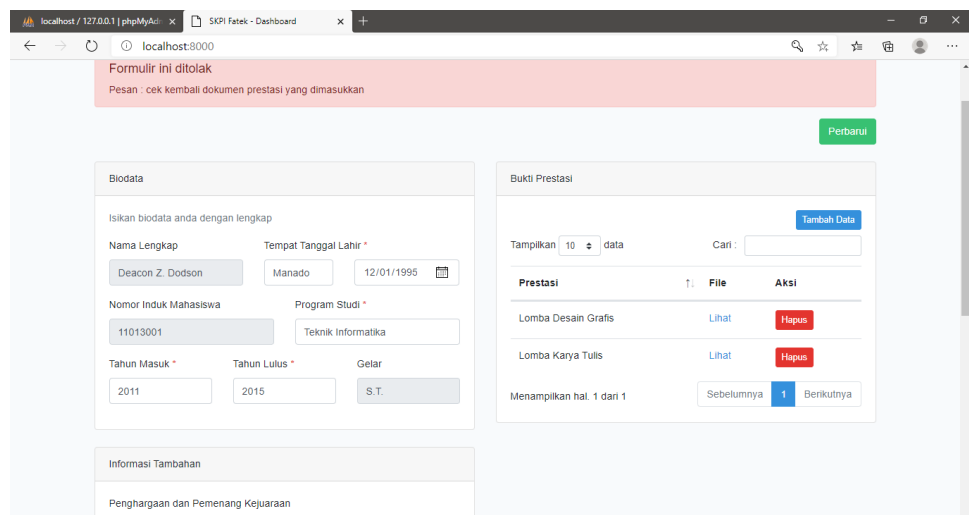
Gambar 4.33 Halaman Utama

Pada Gambar 4.33 di atas menunjukkan halaman utama saat mahasiswa mengakses aplikasi setelah melewati halaman *login*. Terdapat tombol kirim yang berfungsi untuk mengirimkan data yang telah dimasukkan oleh mahasiswa ke dalam *database*. Terdapat juga tombol tambah data pada bagian bukti prestasi yang berfungsi untuk mengirimkan *file* dokumen prestasi yang telah dipilih oleh mahasiswa.



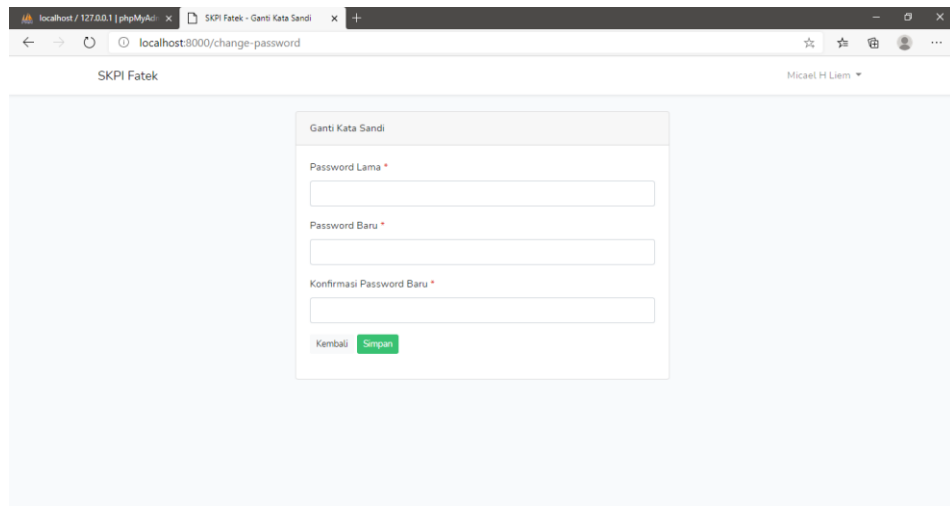
Gambar 4.34 Fungsi Tambah Data

Pada Gambar 4.34 di atas menunjukkan *pop up* menu, yang muncul saat mahasiswa menekan tombol tambah data untuk menambahkan data prestasi dan *file* dokumen PDF.



Gambar 4.35 Fungsi Ubah Data

Gambar 4.35 di atas menampilkan menu utama yang telah berisikan data. Mahasiswa bisa langsung mengubah data tersebut dan menekan tombol perbarui untuk memperbaharui data yang telah tersimpan.



Gambar 4.36 Halaman Ganti Kata Sandi

Halaman ini menampilkan menu ganti kata sandi, yang akan muncul saat admin menekan tombol *dropdown* yang terletak di sudut kanan atas.

4.4.4 Pengujian Aplikasi

Pada tahap ini akan dilakukan pengujian terhadap aplikasi, dengan tujuan untuk mengecek fitur-fitur serta kekurangan yang terdapat dalam aplikasi. Pengujian ini bertujuan untuk mengetahui apakah aplikasi yang dibuat telah memenuhi kriteria yang sesuai dengan kebutuhan pengguna, dan dapat memenuhi tujuan dari pembuatan aplikasi tersebut. Pada tahapan ini pengujian akan dibagi menjadi dua fase, yang pertama merupakan fase *alpha test* dan yang kedua merupakan fase *beta test*.

4.4.4.1 Alpha Testing

Pada fase pertama dari pengujian ini akan dilakukan dengan menggunakan metode *black box*. Pengujian dengan metode *black box* merupakan metode pengujian yang berfokus pada fungsionalitas dari aplikasi. Berikut ini merupakan hasil dari *alpha testing*:

Tabel 4.14 Hasil Pengujian *Black Box*

No.	Role	Fungsionalitas	Keterangan	
			Sukses	Error
1.	Admin (staf fakultas)	Mengakses aplikasi menggunakan beberapa <i>web browser</i> yaitu <i>Google Chrome, Mozilla Firefox, Microsoft Edge</i> .	√	
		Melakukan proses <i>login</i> ke dalam aplikasi.	√	
		Aplikasi menampilkan pemberitahuan kesalahan dan mencegah proses <i>login</i> saat admin salah dalam memasukkan nama pengguna atau kata sandi.	√	
		Melakukan tambah data akun mahasiswa.	√	
		Melakukan perubahan data pada akun mahasiswa.	√	
		Menggunakan fitur pencarian.	√	
		Melakukan proses pengecekan data SKPI.	√	
		Mengirimkan pesan status validasi kepada mahasiswa.	√	
		Menggunakan fitur cetak untuk mencetak SKPI.	√	
		Menghapus akun mahasiswa.	√	
		Melakukan proses <i>logout</i> dari aplikasi.	√	
		2.	Mahasiswa	Mengakses aplikasi menggunakan beberapa <i>web browser</i> yaitu <i>Google Chrome, Mozilla Firefox, Microsoft Edge</i> .

	Melakukan proses <i>login</i> ke dalam aplikasi.	√	
	Aplikasi menampilkan pemberitahuan kesalahan dan mencegah proses <i>login</i> saat mahasiswa salah dalam memasukkan nama pengguna atau kata sandi.	√	
	Melakukan pengisian data SKPI.	√	
	Menambahkan dokumen bukti prestasi.	√	
	Menghapus dokumen bukti prestasi.	√	
	Melakukan perubahan data SKPI.	√	
	Mengganti kata sandi akun.	√	
	Melakukan proses <i>logout</i> dari aplikasi.	√	

Fase pertama pengujian atau *alpha testing* yang dilakukan pada Aplikasi Surat Keterangan Pendamping Ijazah, dengan menggunakan metode *black box* memperoleh hasil bahwa aplikasi telah berjalan dan dapat digunakan dengan baik sesuai dengan hak akses masing-masing pengguna. Fungsionalitas atau fitur-fitur yang ada pada aplikasi juga berfungsi dengan baik.

4.4.4.2 Beta Testing

Fase kedua dari pengujian ini yang disebut dengan fase *beta testing*. Melalui pengujian ini, produk berupa aplikasi langsung diujikan kepada klien atau pengguna. Berikut ini merupakan hasil dari *beta testing*:

Tabel 4.15 Hasil Beta Testing

No.	Karakteristik	Pertanyaan	Penilaian			
			A	B	C	D
1.	<i>Functionality</i>	Apakah aplikasi tersebut dapat melakukan fungsi yang diperlukan?		√		

		Apakah hasil dari pengolahan data pada aplikasi tersebut sesuai dengan yang diharapkan?	√			
		Dapatkah aplikasi tersebut mencegah akses yang tidak sah?	√			
2.	<i>Usability</i>	Apakah aplikasi tersebut dapat dengan mudah untuk dimengerti dan digunakan?	√			
		Apakah langkah-langkah pengoperasian aplikasi dapat dipelajari dengan mudah?	√			
		Apakah tampilan dari aplikasi tersebut menarik?		√		
3.	<i>Portability</i>	Dapatkah aplikasi tersebut di- <i>install</i> dengan mudah?	√			

Keterangan:

A = Sangat baik

B = Baik

C = Kurang

D = Sangat kurang

Pada fase kedua dalam pengujian ini, Aplikasi Surat Keterangan Pendamping Ijazah telah diujikan kepada klien dengan beberapa kriteria yaitu *functionality*, *usability*, dan *portability*. Seperti yang tertera pada Tabel 4.15 hasil dari penilaian menunjukkan bahwa aplikasi telah memenuhi kriteria pengujian dengan baik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan isi dan pembahasan dari bab-bab sebelumnya yang sudah meliputi latar belakang, teori, dan analisis serta implementasi, dan pengujian. Maka penulis dapat mengambil kesimpulan sebagai berikut:

1. Aplikasi dapat melakukan fungsionalitas utama yaitu dalam mencetak Surat Keterangan Pendamping Ijazah.
2. Aplikasi dapat menyimpan dan menampilkan data serta dokumen-dokumen terkait pembuatan Surat Keterangan Pendamping Ijazah.
3. Aplikasi yang dibangun dapat berjalan dengan baik pada platform web yang diujicobakan yaitu *Google Chrome, Mozilla Firefox, Microsoft Edge*.

5.2 Saran

Dalam pembuatan serta pengujian yang dilakukan pada Aplikasi Surat Keterangan Pendamping Ijazah, masih terdapat beberapa hal yang bisa untuk dikembangkan lebih lanjut, seperti:

1. Aplikasi dapat diintegrasikan dengan sistem informasi akademik agar dapat mengambil data mahasiswa dengan lebih mudah.

DAFTAR PUSTAKA

- [1] Admin. *Sejarah Universitas Katolik De La Salle Manado* [Online]. Available: <https://unikadelasalle.ac.id/sejarah-ukdlsm/>
- [2] Admin. *Download* [Online]. Available: <https://unikadelasalle.ac.id/download/>
- [3] Tim KKNi (2015). *Surat Keterangan Pendamping Ijazah Dokumen 008* [Online]. Available: http://kkn.kemdikbud.go.id/asset/pdf/008-dokumen_skpi.pdf
- [4] Larry Alonson, "HTML and the Web," in *HTML manual of style : a clear, concise reference for hypertext markup language (including HTML5)*, 4th ed. Upper Saddle River, NJ: Addison-Wesley, 2011.
- [5] Larry Alonson, "The HTML Language," in *HTML manual of style : a clear, concise reference for hypertext markup language (including HTML5)*, 4th ed. Upper Saddle River, NJ: Addison-Wesley, 2011.
- [6] K. Tatroe, and P. MacIntyre, "Introduction to PHP," in *Programming PHP Creating Dynamic Web Pages*, 4th ed. 1005 Gravenstein Highway North, Sebastopol: O'Reilly Media, 2020.
- [7] David Flanagan, "Introduction to JavaScript," in *JavaScript: The Definitive Guide*, 6th ed. 1005 Gravenstein Highway North, Sebastopol: O'Reilly Media, 2011.
- [8] Rian Ariona (2013). *Belajar HTML dan CSS* [Online]. Available: <http://www.ariona.net/ebook-belajar-html-dan-css/>
- [9] A. Shenoy, and U. Sossou, "Getting Started With Bootstrap," in *Learning Bootstrap*, Birmingham – Mumbai: Packt Publishing, 2014.
- [10] Raphaël Saunier, "Meeting Laravel," in *Getting Started with Laravel 4*, Birmingham – Mumbai: Packt Publishing, 2014.
- [11] D. M. Kronke, D. J. Auer, S. L. Vandenberg, and R. C. Yoder, "DATABASE FUNDAMENTAL," in *Database Concepts*, 8th ed. Hoboken, New Jersey: Pearson, 2017.
- [12] A. Silberschatz, H. Korth, and S. Sudarshan, "Introduction," in *Database System Concepts*, 7th ed. New York: McGraw-Hill, 2020.
- [13] Robin Nixon, "Introduction to MySQL," in *Learning PHP, MySQL & JavaScript*, 1st ed. 1005 Gravenstein Highway North, Sebastopol: O'Reilly Media, 2018.
- [14] Lucidchart Content Team. *4 Phases of Rapid Application Development Methodology* [Online]. Available: <https://www.lucidchart.com/blog/rapid-application-development-methodology>
- [15] A. Dennis, B. H. Wixom, and D. Tegarden, "Rapid Application Development (RAD)," in *System Analyst and Design An Object-Oriented Approach with UML*, 5th ed. New York: Wiley, 2015.
- [16] M. Seidl, M. Scholz, C. Huemer, and G. Kappel, in *UML @ Classroom An Introduction to Object-Oriented Modeling*, 1st ed. Switzerland: Springer International Publishing, 2014.

LAMPIRAN A

WAWANCARA

Berikut ini adalah wawancara yang dilakukan dengan Ketua Program Studi Teknik Informatika Fakultas Teknik Universitas Katolik De La Salle Manado, dengan topik seputar Surat Keterangan Pendamping Ijazah.

Penulis	Apa fungsi dari Surat Keterangan Pendamping Ijazah untuk para lulusan Fakultas Teknik?
Narasumber	Surat Keterangan Pendamping Ijazah berfungsi sebagai pelengkap yang dikeluarkan untuk mendampingi ijazah dan transkrip akademik. Pembuatan SKPI juga didasari oleh Permendikbud yang mewajibkan setiap perguruan tinggi untuk menerbitkan SKPI kepada setiap lulusannya.
Penulis	Bagaimana prosedur pembuatan Surat Keterangan Pendamping Ijazah?
Narasumber	Prosedur pembuatan SKPI dilakukan dengan cara mahasiswa harus datang terlebih dahulu ke program studi, untuk mengajukan pembuatan SKPI dengan membawa serta dokumen-dokumen terkait jika mahasiswa tersebut memilikinya. Kemudian data dari mahasiswa tersebut akan dimasukkan ke dalam format SKPI yang telah tersedia untuk kemudian dicetak.
Penulis	Data apa saja yang diperlukan dalam pembuatan Surat Keterangan Pendamping Ijazah?
Narasumber	Data yang diperlukan dalam proses pembuatan SKPI yang pertama adalah data identitas diri mahasiswa, yang kedua adalah data Universitas yang menjadi penyelenggara program pendidikan, dan yang ketiga adalah data mengenai capaian pembelajaran mahasiswa dimana juga termasuk informasi tambahan berupa prestasi akademik maupun non akademik serta pengalaman mengikuti organisasi.

LAMPIRAN B

USER ACCEPTANCE TEST

No.	Karakteristik	Pertanyaan	Penilaian			
			A	B	C	D
1.	<i>Functionality</i>	Apakah aplikasi tersebut dapat melakukan fungsi yang diperlukan?		√		
		Apakah hasil dari pengolahan data pada aplikasi tersebut sesuai dengan yang diharapkan?		√		
		Dapatkah aplikasi tersebut mencegah akses yang tidak sah?	√			
2.	<i>Usability</i>	Apakah aplikasi tersebut dapat dengan mudah untuk dimengerti dan digunakan?	√			
		Apakah langkah-langkah pengoperasian aplikasi dapat dipelajari dengan mudah?	√			
		Apakah tampilan dari aplikasi tersebut menarik?		√		
3.	<i>Portability</i>	Dapatkah aplikasi tersebut di- <i>install</i> dengan mudah?	√			

Keterangan:

A = Sangat baik

B = Baik

C = Kurang

D = Sangat kurang

Manado, 13 Desember 2020

(Vivie D. Kumenap, S.T., M.Cs)