

**SIPJABS : SISTEM PENGAWAKAN JABATAN STRUKTURAL
DI UNIVERSITAS TELKOM**

SIPJABS : CAREER MANAGEMENT SYSTEM IN TELKOM UNIVERSITY

PROYEK AKHIR

Disusun dalam rangka memenuhi salah satu persyaratan untuk menyelesaikan
Program D3 Rekayasa Perangkat Lunak Aplikasi

Disusun oleh:

Zakaria Wahyu Nur Utomo 6706170028
Elsa Jelista Sari 6706170016



**FAKULTAS ILMU TERAPAN
UNIVERSITAS TELKOM
BANDUNG
2020**

LEMBAR PERNYATAAN ORISINALITAS

Nama : Zakaria Wahyu Nur Utomo	Elsa Jelista Sari
NIM : 6706170028	6706170016
Alamat : Mlese, Ceper, Klaten	Candirejo, Ngawen, Klaten
Email : zakarianur6@gmail.com	firstjanuari@gmail.com

Menyatakan bahwa Proyek Akhir D3 RPLA ini merupakan karya orisinal kami, dengan judul:

**SIPJABS : SISTEM PENGAWAKAN JABATAN STRUKTURAL
DI UNIVERSITAS TELKOM**
SIPJABS : CAREER MANAGEMENT SYSTEM IN TELKOM UNIVERSITY

Atas pernyataan ini, kami siap menanggung resiko/sanksi yang dijatuhkan kepada kami apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini. Jika terbukti melanggar hal-hal di atas, kami bersedia dikenakan sanksi sesuai dengan **Peraturan Akademik dan Kemahasiswaan Universitas Telkom** bagian **Kode Etik Mahasiswa** untuk pelanggaran akademik.

Bandung, 8 Juni 2020

Zakaria Wahyu Nur Utomo Elsa Jelista Sari
6706170028 6706170016

LEMBAR PENGESAHAN

SIPJABS : SISTEM PENGAWAKAN JABATAN STRUKTURAL DI UNIVERSITAS TELKOM

SIPJABS : CAREER MANAGEMENT SYSTEM IN TELKOM UNIVERSITY

**Telah disetujui dan disahkan sebagai Proyek Akhir
Program D3 Rekayasa Perangkat Lunak Aplikasi
Fakultas Ilmu Terapan
Universitas Telkom
Bandung**

Disusun oleh:

Zakaria Wahyu Nur Utomo 6706170028
Elsa Jelista Sari 6706170016

Bandung, 8 Juni 2020

**Menyetujui,
Dosen Pembimbing**

Hetty Hidayati, S.Kom., M.T.
NIP : 06750056

ABSTRAK

Banyak model seleksi yang dilakukan guna untuk menilai seseorang terutama ketika perusahaan mencari posisi jabatan diantaranya dengan melakukan assessment center dan mengisi formulir penilaian untuk setiap kandidat yang akan dicalonkan sebagai pemimpin dan staf. Banyak prosedur serta ketentuan yang harus dimiliki oleh calon pemimpin dan staf, baik itu manajer atau kepala bagian dan staf. Setiap orang yang terpilih berarti telah memenuhi ketentuan yang sudah ditetapkan perusahaan. Ketentuan dibuat berdasarkan kompetensi setiap bagian yang disusun dalam kamus kompetensi perusahaan. Melalui kamus kompetensi tersebut juga dapat dijadikan sebagai pedoman untuk bagian Sumber Daya Manusia dalam mencari pegawai yang berpotensi tinggi demi keberlangsungan perusahaan. Terdapat permasalahan belum adanya proses mekanisme penentuan kandidat yang tepat, apabila terdapat posisi yang digantikan atau kosong. Maka tidak adanya data pegawai yang akan dijadikan kandidat untuk mengisi posisi yang digantikan atau kosong tersebut.

Untuk mengatasi permasalahan diatas, maka di rancang aplikasi "**SiP JabS : Sistem Pengawakan Jabatan Struktural di Universitas Telkom**" yang bertujuan untuk memperbaiki proses manajemen karir. Agar organisasi atau perusahaan memperoleh kandidat yang berkualitas. Dengan memberikan penerapan lebih kompetitif dan adil. Kemudian juga dapat menganalisis risiko, misalnya identifikasi pegawai yang berpotensi keluar. Maka dapat meningkatkan program pembelajaran dan pengembangan untuk kinerja dan mengembangkan kompetensi yang lebih baik di masa depan.

Kata Kunci: *Pegawai, Perekutan, Perusahaan*

ABSTRACT

Many selection models are carried out in order to assess someone, especially when a company seeks positions such as by conducting an assessment center and filling out an assessment form for each candidate to be nominated as a leader and staff. Many procedures and conditions must be had by prospective leaders and staff, be it managers or division heads and staff. Every person chosen means fulfilling the conditions set by the company. Provisions are made based on the competence of each section compiled in the company competency dictionary. Through the competency dictionary, it can also be used as a guideline for the Human Resources section in looking for high-potential employees for the sustainability of the company. There is a problem that there is no mechanism for determining the right candidate, if there is a position that is replaced or vacant. Then there is no employee data that will be used as a candidate to fill the replaced or vacant position.

To overcome the above problems, the application design "**SiPJabS : Career Management System in Telkom University**" aims to improve the career management process. So that the organization or company gets a high quality. By providing more competitive and fair application. Then it can also analyze risks, for example identification of employees who have the potential to leave. Then it can improve learning and development programs for performance and develop better competencies in the future.

Keywords: Employees, Recruitment, Companies

LEMBAR PERSEMBAHAN

Bismillahirrohmanirrohim

Segala puji dan syukur kami panjatkan kepada **Allah Subhanahu Wa Ta’ala**.
Alhamdulilahi robbil ’alamin, karena atas rahmat dan hidayah-Nya kami dapat
menyelesaikan Proyek Akhir yang sederhana ini dengan baik. Karya ini kami
persesembahkan untuk:

Ibu dan Bapak kami tersayang
Terima kasih atas segala pengorbanan, doa dan kasih sayangnya.

Kakak, adik dan saudara – saudara kami yang tidak bisa disebutkan satu persatu
Terima kasih atas segala kasih sayang dan dukungan yang diberikan.

Teman – teman yang ada disekitar kami yang tidak bisa disebutkan satu persatu
Terimakasih atas dukungannya, motivasi, semangat, persahabatan sekaligus
persaudaraannya yang selalu membangun.

Dosen Pembimbing kami ibu Hetti Hidayati, S.Kom., MT.
Terimakasih atas bimbingannya dan arahannya selama ini.

KATA PENGANTAR

Dengan mengucap puji syukur kami panjatkan kehadirat Allah Subhanahu Wa Ta'ala yang telah melimpahkan rahmat dan karunia-Nya sehingga kami dapat menyelesaikan Proyek Akhir ini dengan baik. Adapun judul Proyek Akhir yaitu **"SiPJabS : Sistem Pengawakan Jabatan Struktural di Universitas Telkom"**.

Tujuan penulisan Proyek Akhir ini dibuat sebagai salah satu syarat kelulusan Diploma Tiga (D.III) Rekayasa Perangkat Lunak Aplikasi Universitas Telkom. Sebagai bahan penulisan diambil berdasarkan penelitian, observasi dan beberapa sumber yang turut mendukung dalam penulisan ini. Penulis menyadari bahwa tanpa bimbingan dan dorongan dari semua pihak, maka Proyek Akhir ini tidak dapat terselesaikan dengan baik.

Penulis menyadari sepenuhnya bahwa Proyek Akhir ini jauh dari kata sempurna, oleh karena itu penulis mengharapkan saran dan kritik yang bersifat membangun yang lebih baik untuk generasi penerus kita.

Bandung, 8 Juni 2020

Zakaria Wahyu Nur Utomo Elsa Jelista Sari
6706170028 6706170016

DAFTAR ISI

LEMBAR PERNYATAAN ORISINALITAS

LEMBAR PENGESAHAN

ABSTRAK	iv
ABSTRACT	v
LEMBAR PERSEMPAHAN	vi
KATA PENGANTAR	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Permasalahan	2
1.4 Tujuan	3
1.5 Metode Penyelesaian Masalah	3
1.6 Pembagian Tugas Anggota	4
II TINJAUAN PUSTAKA	5
2.1 Definisi Pengawakan	5
2.2 Proses Pengawakan	5
2.3 Manfaat Pengawakan	7
2.4 Tujuan Pengawakan	8
2.5 Hasil dan Manfaat Manajemen Pengawakan	8
2.6 MySQL dan Basis Data	9
2.7 <i>Framework</i> Laravel	10
2.8 <i>Domain</i>	12
2.9 <i>Hosting</i>	13

III ANALISIS KEBUTUHAN DAN PERANCANGAN	14
3.1 Sistem Arsitektur	14
3.1.1 Gambaran Umum Sistem	14
3.1.2 Target Pengguna Aplikasi	15
3.1.3 Spesifikasi Target Perangkat	15
3.1.4 Diagram Alir Aplikasi	16
3.2 Kebutuhan Pengembangan Sistem	17
3.2.1 Kubutuhan Perangkat Keras (<i>Hardware</i>)	17
3.2.2 Kebutuhkan Perangkat Lunak (<i>Software</i>)	18
3.2.3 Kebutuhan <i>Hosting</i>	18
3.3 Perancangan Model Program	19
3.3.1 Use Case Diagram	19
3.3.2 Use Case Skenario	20
3.3.3 Class Diagram	48
3.3.4 Entity Relationship Diagram	49
3.4 Perancangan Aplikasi	49
3.4.1 Perancangan Antar Muka	49
3.4.1.1 Perancangan Antar Muka Admin	50
3.4.1.2 Perancangan Antar Muka User	62
3.4.2 Perancangan Level Tinggi	66
IV IMPLEMENTASI DAN PENGUJIAN	67
4.1 Implementasi Aplikasi	67
4.1.1 Implementasi Antarmuka Aplikasi	67
4.1.1.1 Implementasi Admin	67
4.1.1.2 Implementasi <i>User</i>	85
4.1.2 Struktur Kode	91
4.2 Pengujian Aplikasi	93
4.2.1 Pengujian Alpha	93
4.2.1.1 Pengujian Fungsionalitas	94
4.2.1.2 Pengujian Kesesuaian	101
4.2.2 Pengujian Beta	102
4.3 Diskusi Hasil Pengujian	103
V KESIMPULAN DAN SARAN	104
5.1 Kesimpulan	104
5.2 Saran	104

DAFTAR GAMBAR

2.1	<i>MySQL Database</i>	9
2.2	<i>Framework Laravel</i>	10
2.3	Illustrasi MVC	11
3.1	<i>Low Level Design</i>	14
3.2	<i>Flowchart</i>	16
3.3	Use Case Diagram	19
3.4	<i>Class Diagram</i>	48
3.5	<i>ERD</i>	49
3.6	<i>High Level Design</i>	66
4.1	Halaman <i>Login Admin</i>	67
4.2	Halaman <i>Dashboard Admin</i>	68
4.3	Halaman <i>Profile Admin</i>	69
4.4	Halaman <i>Edit Profile Admin</i>	69
4.5	Halaman <i>Reset Password Admin</i>	70
4.6	Halaman <i>Help Admin</i>	70
4.7	Halaman Data <i>Users</i> - Admin	71
4.8	Halaman <i>Edit Data Users</i> - Admin	71
4.9	Halaman Tambah <i>Users</i> - Admin	72
4.10	Halaman Data Pegawai - Admin	72
4.11	Halaman <i>View Detail Pegawai</i> - Admin	73
4.12	Halaman <i>Edit Pegawai</i> - Admin	73
4.13	Halaman Data Kandidat - Admin	74
4.14	Halaman <i>View Detail Kandidat</i> - Admin	74
4.15	Halaman Data Unit Kerja - Admin	75
4.16	Halaman <i>Edit Unit Kerja</i> - Admin	75
4.17	Halaman Tambah Unit Kerja - Admin	76
4.18	Halaman Data Jabatan - Admin	76
4.19	Halaman <i>Edit Jabatan</i> - Admin	77
4.20	Halaman Tambah Jabatan - Admin	77
4.21	Halaman Data Unit Bagian - Admin	78
4.22	Halaman <i>Edit Unit Bagian</i> - Admin	78

4.23 Halaman Tambah Unit Bagian - Admin	79
4.24 Halaman Data Jabatan Struktural - Admin	79
4.25 Halaman <i>Edit</i> Jabatan Struktural - Admin	80
4.26 Halaman Tambah Jabatan Struktural - Admin	80
4.27 Halaman Data Pendidikan - Admin	81
4.28 Halaman <i>Edit</i> Pendidikan - Admin	81
4.29 Halaman Tambah Pendidikan - Admin	82
4.30 Halaman Data <i>Skill</i> - Admin	82
4.31 Halaman <i>Edit Skill</i> - Admin	83
4.32 Halaman Tambah Skill - Admin	83
4.33 Halaman Data <i>Personal Quality</i> - Admin	84
4.34 Halaman <i>Edit Personal Wuality</i> - Admin	84
4.35 Halaman Tambah Personal Quality - Admin	85
4.36 Halaman <i>Login User</i>	85
4.37 Halaman <i>Dashboard User</i>	86
4.38 Halaman <i>Profile User</i>	86
4.39 Halaman <i>Edit Profile User</i>	87
4.40 Halaman <i>Reset Password User</i>	87
4.41 Halaman <i>Help User</i>	88
4.42 Halaman Cari Kandidat - User	88
4.43 Halaman <i>Filtering</i> - User	89
4.44 Halaman <i>View Detail Pegawai</i> - User	89
4.45 Halaman <i>Cart</i> - User	90
4.46 Halaman <i>View Kandidat Sementara</i>	90
4.47 Halaman Data Kandidat	91

DAFTAR TABEL

3.1	Tabel Kebutuhan <i>Hardware</i>	17
3.2	Tabel Kebutuhan <i>Software</i>	18
3.3	Tabel Kebutuhan <i>Hosting</i>	18
3.4	Skenario <i>Login</i>	20
3.5	Skenario Edit <i>Profile</i>	21
3.6	Skenario Reset <i>Password</i>	21
3.7	Skenario Tambah <i>Users</i>	22
3.8	Skenario Edit <i>Users</i>	23
3.9	Skenario Delete <i>Users</i>	24
3.10	Skenario View Pegawai	25
3.11	Skenario Tambah Pegawai	25
3.12	Skenario Edit Pegawai	26
3.13	Skenario View Kandidat	27
3.14	Skenario Print Form Kandidat	28
3.15	Skenario Tambah Unit Kerja	28
3.16	Skenario Edit Unit Kerja	29
3.17	Skenario View Unit Kerja	30
3.18	Skenario Tambah Jabatan	31
3.19	Skenario Edit Jabatan	32
3.20	Skenario View Jabatan	32
3.21	Skenario Tambah Unit Bagian	33
3.22	Skenario Edit Unit Bagian	34
3.23	Skenario View Unit Bagian	35
3.24	Skenario Tambah Jabatan Struktural	36
3.25	Skenario Edit Jabatan Struktural	37
3.26	Skenario View Jabatan Struktural	37
3.27	Skenario Tambah Pendidikan	38
3.28	Skenario Edit Pendidikan	39
3.29	Skenario View Pendidikan	40
3.30	Skenario Tambah Skill	40
3.31	Skenario Edit Skill	41
3.32	Skenario View Skill	42

3.33 Skenario Tambah <i>Personal Quality</i>	43
3.34 Skenario Edit <i>Personal Quality</i>	44
3.35 Skenario View <i>Personal Quality</i>	44
3.36 Skenario Cari Kandidat	45
3.37 Skenario Memilih Kandidat	46
3.38 Skenario View Kandidat Sementara	47
3.39 Skenario Proses Kandidat Sementara	47
3.40 Tabel Perancangan Antar Muka Admin	50
3.41 Tabel Perancangan Antar Muka Admin (1)	51
3.42 Tabel Perancangan Antar Muka Admin (2)	52
3.43 Tabel Perancangan Antar Muka Admin (3)	53
3.44 Tabel Perancangan Antar Muka Admin (4)	54
3.45 Tabel Perancangan Antar Muka Admin (5)	55
3.46 Tabel Perancangan Antar Muka Admin (6)	56
3.47 Tabel Perancangan Antar Muka Admin (7)	57
3.48 Tabel Perancangan Antar Muka Admin (8)	58
3.49 Tabel Perancangan Antar Muka Admin (9)	59
3.50 Tabel Perancangan Antar Muka Admin (10)	60
3.51 Tabel Perancangan Antar Muka Admin (11)	61
3.52 Tabel Perancangan Antar Muka User	62
3.53 Tabel Perancangan Antar Muka User (1)	63
3.54 Tabel Perancangan Antar Muka User (2)	64
3.55 Tabel Perancangan Antar Muka User (3)	65
4.1 Tabel Struktur Kode	91
4.2 Tabel Struktur Kode (1)	92
4.3 Tabel Struktur Kode (2)	93
4.4 Tabel Pengujian Login Admin	94
4.5 Tabel Pengujian Login User	95
4.6 Tabel Pengujian Cari Kandidat	96
4.7 Tabel Pengujian Filtering	97
4.8 Tabel Pengujian Pilih Kandidat	98
4.9 Tabel Pengujian Proses Kandidat	99
4.10 Tabel Pengujian Cetak PDF	100
4.11 Tabel Pengujian Kesesuaian	101
4.12 Tabel Usability Testing	102

BAB I

PENDAHULUAN

1.1 Latar Belakang

Keberlangsungan perguruan tinggi Universitas Telkom tak akan lepas dari peran orang-orang yang bekerja didalamnya. Dengan struktur organisasi yang kompleks, menjadikan Universitas Telkom terdepan dibidangnya. Setiap pemegang jabatan memiliki peran yang penting dalam menunjang visi dan misi Universitas Telkom untuk mencapai tujuannya. Oleh karena itu, setiap orang yang terpilih untuk memegang jabatan penting di perguruan tinggi ini pasti memiliki kompetensi yang sesuai dengan apa yang diharapkan. Orang-orang tersebut dapat terpilih melalui tahap seleksi yang panjang, agar perguruan tinggi Universitas Telkom mendapatkan orang-orang terbaik untuk menjalankan tugasnya.

Banyak model seleksi yang dilakukan untuk menilai seseorang terutama ketika perusahaan mencari seorang pemimpin dan staf, diantaranya dengan melakukan *assessment center* dan mengisi formulir penilaian untuk setiap kandidat yang akan dicalonkan sebagai pemimpin dan staf. Banyak prosedur serta ketentuan yang harus dimiliki oleh calon pemimpin dan staf, baik itu manajer, kepala bagian, kepala urusan, sekretaris atau staf. Setiap orang yang terpilih berarti telah memenuhi ketentuan yang sudah ditetapkan perusahaan. Ketentuan dibuat berdasarkan kompetensi setiap bagian yang disusun dalam kamus kompetensi perusahaan. Melalui kamus kompetensi tersebut juga dapat dijadikan sebagai pedoman untuk bagian sumber daya manusia dalam mencari pegawai yang berpotensi tinggi demi keberlangsungan perusahaan [1].

Masalah yang paling banyak dijumpai pada suatu perusahaan yaitu berkaitan dengan pencarian kandidat yang sesuai dengan *job description*. Banyak kandidat memiliki *skill* yang sama dengan kandidat lainnya, namun perusahaan mencari kandidat sesuai dengan *requirement* yang telah ditetapkan oleh perusahaan. Dengan begitu, proses *filtering* akan membutuhkan waktu yang lama, jika kandidat tidak cepat ditemukan sesuai *requirement* yang ada. Untuk mengatasi permasalahan tersebut, proses *filtering* akan dipindahkan dengan aplikasi “**SiP JabS : Sistem Pengawakan Jabatan Struktural**”, yang diharapkan dapat membantu penemuan kandidat yang sesuai dengan *requirement* yang sudah ditetapkan oleh perusahaan.

Terdapat permasalahan belum adanya proses mekanisme penentuan kandidat yang tepat, apabila terdapat posisi yang kosong. Maka tidak adanya data pegawai yang akan dijadikan kandidat untuk mengisi posisi yang kosong tersebut, sehingga perlu dirancang sistem yang mampu mengidentifikasi sesuai kebutuhan posisi yang diinginkan. Yang kemudian hasilnya akan disesuaikan dengan *job description* yang dibutuhkan oleh perusahaan. Dengan cara ini, manajer di perusahaan dapat menentukan profil pegawai yang tepat sesuai *requirement* yang sudah ditetapkan oleh perusahaan.

Untuk mengatasi permasalahan di atas, kami merancang Proyek Akhir ini dengan membuat sistem pengawakan jabatan struktural yang bertujuan untuk untuk mengelola proses pencarian kandidat yang akan dicalonkan sebagai pemimpin dan staf, agar perusahaan memperoleh kandidat yang memenuhi ketentuan yang sudah ditetapkan perusahaan dengan menerapkan penilaian yang lebih lengkap dan adil. Diharapkan dengan adanya sistem ini, manajer mampu menganalisa kebutuhan program pengembangan kompetensi sumber daya manusia yang lebih baik di masa depan.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang akan dibahas adalah sebagai berikut:

1. Bagaimana proses pengisian posisi jabatan yang tepat?
2. Bagaimana cara pencarian kandidat yang sesuai dengan *requirement*?
3. Bagaimana proses *filtering* berjalan efektif?

1.3 Batasan Permasalahan

Batasan masalah yang terdapat dapat dari perumusan masalah di atas adalah sebagai berikut:

1. Aplikasi ini ditunjukan untuk pegawai Direktorat Sumber Daya Manusia Universitas Telkom.
2. Aplikasi ini dibangun dengan menggunakan sistem berbasis web.
3. Setiap pemilihan jabatan memiliki parameter berbeda, yang mengacu pada metode penilaian.

1.4 Tujuan

Tujuan yang ingin dicapai dari perancangan Proyek Akhir ini adalah:

1. Membangun aplikasi pengawakan jabatan struktural yang bertujuan untuk mengelola proses pencarian kandidat yang akan dicalonkan sebagai pemimpin dan staf, agar perusahaan memperoleh kandidat yang memenuhi ketentuan yang sudah ditetapkan perusahaan dengan menerapkan penilaian yang lebih lengkap dan adil. Diharapkan dengan adanya sistem ini, manajer mampu menganalisa kebutuhan program pengembangan kompetensi sumber daya manusia yang lebih baik di masa depan.
2. Adanya sistem pengawakan jabatan struktural, pengguna dapat menggunakan aplikasi setiap saat untuk mencari kandidat dan mengisi posisi yang kosong karena mengantikan pekerjaan lama yang telah berhenti dikarenakan pensiun, meninggal, mengundurkan diri atau diberhentikan karena suatu kebijakan tertentu.

1.5 Metode Penyelesaian Masalah

Metodologi untuk menyelesaikan masalah diatas adalah sebagai berikut:

1. Tahap studi literatur

Tahap pertama ini dilakukan dengan cara mencari, menganalisa dan mempelajari informasi yang berhubungan dengan Proyek Akhir. Topik yang berhubungan antara lain:

- (a) Data pegawai secara lengkap.
- (b) Data jabatan struktural.
- (c) Data *requirement* pencarian kandidat.

Serta teori lain yang berhubungan dengan pengembangan aplikasi. Referensi dapat dicari melalui buku, jurnal, *paper*, dan media lainnya baik *daring* maupun *luring*.

2. Tahap pencarian dan pengumpulan data

Pencarian dan pengumpulan data yang diperlukan dalam pengembangan aplikasi ini seperti data yang terdapat pada I-GRACIAS dan data yang terdapat pada Direktorat Sumber Daya Manusia.

3. Tahap perancangan sistem

Perancangan sistem aplikasi ini dimulai dengan perancangan mockup atau desain UI/UX aplikasi serta merancang *database* dan kerangka program yang akan digunakan.

4. Tahap implementasi

Tahap ini dilakukan realisasi dari perancangan sistem yang telah dibuat, seperti membuat *prototype* dan UI dari aplikasi, pembuatan *database* dan aplikasi yang sudah direncanakan pada tahap perancangan sistem.

5. Tahap pengujian dan analisis

Pengujian dan analisis ini dilakukan apabila aplikasi sudah selesai dibuat serta di *hosting* dan sesuai dengan rancangan sistem yang sudah tertulis. Di tahap ini juga dilakukan analisa permasalahan yang terjadi di aplikasi sebelum aplikasi di luncurkan dan digunakan oleh pengguna.

6. Tahap pembuatan laporan

Tahap terakhir ini bertujuan untuk membuat dokumentasi hasil penelitian dalam bentuk laporan Proyek Akhir. Laporan Proyek Akhir akan menjelaskan apapun yang berhubungan dengan perancangan dan pengujian aplikasi.

1.6 Pembagian Tugas Anggota

Pembagian tugas untuk Proyek Akhir ini adalah sebagai berikut:

1. Zakaria Wahyu Nur Utomo

Peran : *Back End Developer* dan *Database*

Tanggung Jawab:

(a) Merancang dan membuat sistem aplikasi

(b) Pembuatan buku, poster dan vidio promosi

2. Elsa Jelista Sari

Peran : *Front End Developer* dan *Analyst*

Tanggung Jawab:

(a) Pembuatan *user interface / mockup* dan pengujian aplikasi

(b) Pembuatan buku, jurnal, *user manual* dan vidio demo

BAB II

TINJAUAN PUSTAKA

2.1 Definisi Pengawakan

Menurut Rivai dan Segala (2010:198) :

“Kepuasan kerja akan tercapai bila terdapatnya kesesuaian karyawan dengan posisi pekerjaan yang mereka dapatkan. Posisi *staffing* karyawan berarti mengalokasikan para karyawan pada posisi kerja tertentu” [2].

Kemudian, Ardana (2012:18) menambahkan mengenai posisi *staffing* sebagai berikut :

“Posisi *staffing* karyawan merupakan pencocokan atau membandingkan kualifikasi yang dimiliki dengan persyaratan pekerjaan dan sekaligus memberikan tugas, pekerjaan kepada calon karyawan untuk dilaksanakan” [2].

Pengertian diatas dapat disimpulkan bahwa posisi *staffing* yang tepat tidak cukup. Untuk menunjang kinerja karyawan, melainkan membutuhkan pengalaman kerja karyawan untuk menunjang pekerjaan tersebut.

2.2 Proses Pengawakan

Menurut T. Hani Handoko (2000:230) langkah-langkah dalam proses *staffing* meliputi beberapa aspek yaitu [3]:

1. Perencanaan sumber daya manusia

Pemenuhan kebutuhan organisasi untuk mengisi posisi tertentu, untuk itu perlu adanya perencanaan yang terdiri atas:

- (a) Penentuan jabatan yang akan diisi, kemampuan yang dibutuhkan, serta jumlah yang dibutuhkan.
- (b) Pemahaman pasar tenaga kerja potensial.
- (c) Pertimbangan kondisi permintaan dan penawaran karyawan. Apabila suatu perusahaan membutuhkan tenaga kerja baru, maka perusahaan akan mencari orang yang cakap dan terampil untuk mengisi tugas yang kosong tersebut serta mempunyai motivasi untuk melaksanakan misi dan tujuan perusahaan tersebut. Perusahaan bisa memperoleh tenaga kerja tersebut melalui dua sumber yaitu, sumber dari perusahaan (*intern*)

dan sumber dari luar perusahaan (*ekstern*), sumber dari dalam perusahaan yaitu dengan menggunakan orang-orang yang bekerja dalam perusahaan tersebut terutama dalam rangka promosi dan mutasi jabatan, sedangkan sumber yang berasal dari luar perusahaan seperti sekolah-sekolah, departemen tenaga kerja, iklan, dan lain-lain.

2. Penarikan tenaga kerja

Rekrutmen karyawan dilakukan untuk menggantikan pekerjaan lama yang telah berhenti dikarenakan pensiun, meninggal, mengundurkan diri atau diberhentikan karena suatu kebijakan tertentu. Pada organisasi *fitness center*, penambahan dan rekrutmen jumlah karyawan atau instruktur juga disesuaikan dengan penambahan jumlah pendaftaran *members* baru.

3. Penyeleksian tenaga kerja

Seleksi adalah kegiatan untuk mendapatkan tenaga kerja yang paling cakap dan memenuhi persyaratan jabatan. Dalam proses seleksi ini diadakan penilaian sifat-sifat dan karakteristik calon pegawai yang diterima, yaitu calon yang memenuhi syarat sebagaimana telah ditentukan. Dalam *requirement* karyawan, terjadi tahapan pengumuman pendaftaran, tahapan pendaftaran sesuai bidang yang dibutuhkan, serangkaian tes atau seleksi, dan pengumuman kelulusan. Para peserta yang lulus seleksi akhir, dinyatakan sebagai karyawan baru yang siap berkontribusi pada organisasi.

4. Kualitas pegawai baru

Orientasi pegawai sangat penting terutama bagi perusahaan besar dimana pimpinan tidak mungkin mengadakan pengawasan langsung. Masa percobaan ini merupakan proses penerimaan pegawai dari penerimaan sampai diterimanya pegawai tersebut menjadi pegawai tetap atau secara resmi.

5. Latihan dan pengembangan karyawan

Tenaga kerja perlu dilatih dan dikembangkan agar dapat melaksanakan pekerjaannya dengan baik. Manfaat dari latihan dan pengembangan adalah untuk mempermudah seseorang melakukan tugasnya. Dengan adanya latihan dan pengembangan yang baik, perusahaan akan memperoleh tenaga kerja, yang cakap dan terlatih sehingga dapat melakukan pekerjaannya dengan efisien. Dalam melaksanakan tugasnya, seorang karyawan tidak mungkin statis, tetapi harus dinamis serta senantiasa berusaha untuk untuk dapat meningkatkan prestasi dan hasil karyanya, oleh karena itu keterampilan dan pengetahuan karyawan perlu dikembangkan melalui “*in service training*”.

6. Penilaian pelaksanaan kerja karyawan

Pada dasarnya penilaian pegawai mempunyai manfaat ganda karena dapat digunakan sebagai alat dalam mengambil keputusan seperti untuk pembayaran upah, gaji, bonus, alat dan pemberian nasehat kepada pegawai. Penilaian sebaiknya dilakukan oleh suatu tim yang terdiri dari atasan langsung sebagai ketua, psikolog, dan seseorang lainnya sebagai anggota. Penilaian karyawan mengacu pada sistem karir dan hasil prestasi kerja. Pada sistem karir yang dilihat adalah kecakapan karyawan yang bersangkutan, pengalaman dalam bekerja, kesetiaan pada organisasi, pengabdian dari segi lamanya waktu bekerja dan syarat objektif lainnya.

7. Pemberian balas jasa dan penghargaan

Kompensasi diberikan sebagai balas jasa dan penghargaan kepada karyawan. Kompensasi yang diberikan perusahaan bisa sebagai alat untuk memotivasi pegawai agar bekerja dengan lebih baik. Kompensasi merupakan kompensasi biaya yang besar bagi perusahaan. Hal ini perlu mendapatkan perhatian agar biaya yang dikeluarkan tidak sia-sia. Pemberian balas jasa ini meliputi pembayaran insentif atau gaji harus adil, layak, tepat waktu sesuai dengan peraturan yang berlaku, dan memberikan kepuasan kepada semua pihak baik karyawan maupun atasan atau pimpinan.

Dengan adanya langkah-langkah tersebut, perusahaan berharap dapat memperoleh tenaga kerja yang benar-benar sesuai dengan kebutuhan dan dapat bekerja dengan efisien dan efektif sehingga terjadi peningkatan produktivitas kerja dan sebagai dampak akhirnya perusahaan mencapai tujuannya.

2.3 Manfaat Pengawakan

Manfaat dari pengawakan terdiri dari:

1. Memposisikan pegawai sesuai dengan *job description*.
2. Karyawan bekerja dengan baik karena adanya latihan dan pengembangan yang baik.
3. Perusahaan mengalami peningkatan produktivitas kerja sehingga dapat mencapai tujuan dengan efisien.

2.4 Tujuan Pengawakan

Menurut Janet B. Parks (2007:338) tujuan penyusunan pengawakan adalah [3]:

1. Terwujudnya sinergitas pekerjaan sesuai dengan seluruh tugas dan kewajibannya.
2. Terwujudnya mekanisme kerja yang koperatif, efektif dan terpadu.
3. Memudahkan pekerjaan dengan keahlian pada bidang masing-masing menyelesaikan tugasnya dengan baik.
4. Mendorong pekerjaan untuk memberikan dana guna dan hasil guna yang maksimal bagi organisasi.

2.5 Hasil dan Manfaat Manajemen Pengawakan

Program manajemen pengawakan yang berhasil dapat membantu perusahaan untuk menjawab tantangan bisnis, memasuki wilayah pasar yang baru dan bergerak maju menyaingi kompetitor. Karyawan yang bertalenta akan lebih tertarik untuk bekerja diperusahaan yang menghargai karyawan dan memberikan kesempatan untuk terus menggapai keberhasilan.

Menurut Pella dan Inayati (2011:87) [4] :

“Manfaat program manajemen talenta yaitu tersedia terus-menerus karyawan yang mencapai potensi terbaik mereka masing-masing, maupun mengembangkan reputasi publik untuk menjadi tempat bekerja yang bagus, sekaligus memupuk loyalitas para karyawan yang telah bekerja didalam perusahaan.

2.6 MySQL dan Basis Data



Gambar 2.1: MySQL Database

Menurut Wahana Komputer (2010:21) :

“MySQL adalah *database server open source* yang cukup popular keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat *software database* ini banyak digunakan oleh praktisi untuk membangun suatu proyek. Adanya fasilitas API (*Application Programming Interface*) yang dimiliki oleh MySQL, memungkinkan bermacam – macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data MySQL.”

Tipe data MySQL, menurut Kustiyahningsih (2011:147) :

“Tipe data MySQL adalah data yang terdapat dalam sebuah tabel berupa *field – field* yang berisi nilai dari data tersebut. Nilai data dalam *field* memiliki tipe sendiri – sendiri”.

Beberapa keunggulan MySQL dibandingkan dengan database lain adalah [5] :

1. Kecepatan MySQL cepat

Para pengembang berpendapat bahwa MySQL adalah *database* yang tercepat yang didapat. Pendapat ini dapat diselidiki dengan mengunjungi <http://www.mysql.com/benchmark.html>

2. Kemudahan dalam penggunaan

MySQL adalah simple database sistem dengan performa tinggi dan tidak kompleks untuk *set up*, *administrator*, dibanding dengan sistem yang lebih besar.

3. Mendukung bahasa *Query*

MySQL memahami SQL, juga dapat mengakses MySQL menggunakan aplikasi yang mendukung ODBC.

4. Kemampuan banyak *client* dapat berhubungan *server* pada saat yang bersamaan. *Clients* dapat menggunakan *multiple database* secara bersamaan.

2.7 Framework Laravel



Gambar 2.2: Framework Laravel

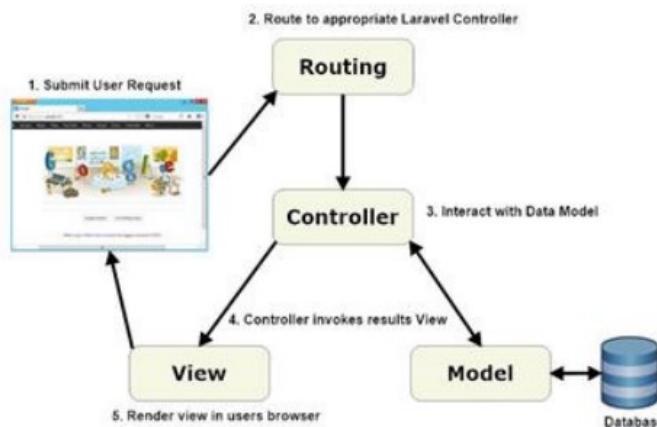
Laravel adalah sebuah *framework* web yang berbasis PHP yang tidak berbayar dan *open-source*, diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola MVC (*Model, View, Controller*). Pola MVC memiliki struktur yang berbeda dari struktur pola MVC pada umumnya.

Pengertian *framework* menurut Naista adalah :

“Suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat *website* lebih singkat dan memudahkan dalam melakukan perbaikan.”

“Salah satu *framework* yang banyak digunakan oleh *programmer* adalah *framework* laravel. Laravel adalah *framework* berbasis PHP yang sifatnya *open source*, dan menggunakan konsep *model – view – controller*. Laravel berada di bawah lisensi MIT, License dengan menggunakan Github sebagai tempat berbagi *code* (Naista, 2017).”

“Dalam penggunaanya laravel memiliki beberapa kekurangan salah satunya yaitu ukuran file yang cukup besar. Di dalam laravel terdapat file yang sifatnya default seperti vendor. File tersebut tidak boleh dihapus sembarangan sehingga ukuran website yang dibuat berukuran cukup besar. Selain itu, dibutuhkan koneksi internet untuk instalasi dan mengunduh *library* laravel, dan PHP minimal versi 5.4 untuk menjalankannya (Naista, 2017).”



Gambar 2.3: Illustrasi MVC

Terdapat 5 konsep arsitektur pada *framework* laravel yang masing-masing memiliki fungsi sebagai berikut [6]:

1. *Routes* berfungsi untuk memberi akses pada setiap *request* sesuai alur yang telah di tentukan. *Routes* memiliki 4 instruksi standar, diantaranya:
 - (a) *Get* : untuk memanggil *request*.
 - (b) *Put* : untuk mengambil data sesuai *request*.
 - (c) *Post* : untuk menambahkan data sesuai *request*.
 - (d) *Delete* : untuk menghapus data sesuai *request*.
2. *Controller* merupakan bagian penghubung antara *model* dan *view*. *Controller* memiliki perintah yang berfungsi untuk memproses bagaimana data ditampilkan dari *Model* ke *View* atau sebaliknya. *Controller* memiliki struktur untuk penulisan kode program pada laravel yaitu :
 - (a) *Index* : untuk menampilkan data keseluruhan.
 - (b) *Create* : untuk memanggil *form* yang berisi kolom inputan.
 - (c) *Store* : untuk menyimpan data ke dalam *table*.
 - (d) *Show* : untuk menampilkan data sesuai dengan ID.
 - (e) *Edit* : untuk memanggil data sesuai dengan ID yang berisi *form* inputan untuk proses *update*.
 - (f) *Update* : untuk *update* data pada tabel.
 - (g) *Delete* : untuk menghapus data sesuai ID.

3. *Model* yaitu sekumpulan data yang memiliki fungsi untuk mengelola *table* pada *database*. Struktur pemodelan data pada laravel yaitu memiliki fungsi yang terdiri dari *table*, *primary key* dan *fillable*. Dimana ketiga fungsi tersebut harus *protected*. Pada bagian *table* harus diisi dengan nama *table* yang sesuai pada database, di bagian *primary key* harus diisi sesuai *primary key* pada *table* tersebut dan pada bagian *fillable* diisi dengan bagian-bagian yang mencangkup dalam *table* tersebut.
4. *View* adalah file yang berisi kode HTML (*HyperText Markup Language*) yang berfungsi untuk menampilkan suatu data ke dalam *browser*. Format *view* pada laravel harus menggunakan istilah *blade*, contohnya: *view.blade.php*.
5. *Migrations* merupakan proses perancangan suatu *table*, dalam hal ini *migrations* berfungsi untuk *blueprint database* atau dapat diistilahkan sebagai penyedia sistem kontrol untuk skema *database*.

2.8 *Domain*

Nama *domain* (*Domain name/URL-Uniform Resource Locator*)(Ali Zaki, 2009) [7] :

Nama *domain* atau biasanya disebut dengan *domain name* atau URL adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah *website*, atau dengan kata lain *domain name* adalah alamat yang digunakan untuk menemukan sebuah *website* pada dunia internet. Contoh: <http://www/baliorange.net>. Nama *domain* diperjualbelikan secara bebas di internet dengan status sewa tahunan. Setelah nama *domain* itu terbeli di salah satu penyedia jasa pendaftaran, maka pengguna disediakan sebuah kontrol panel untuk administrasinya. Jika pengguna lupa/tidak memperpanjang masa sewa, maka Nama *domain* itu akan di lepas lagi ketersediaannya untuk umum. Nama *domain* sendiri mempunyai identifikasi ekstensi/akhiran sesuai dengan kepentingan dan lokasi keberadaan *website* tersebut, contoh nama *domain* ber-ekstensi internasional adalam com, net, org, info, biz, name, ws. Contoh nama *domain* ber-ekstensi lokasi Negara Indonesia adalah:

1. co.id untuk badan usaha yang mempunyai badan hukum sah
2. ac.id untuk lembaga pendidikan
3. go.id khusus untuk lembaga pemerintahan republik Indonesia
4. mil.id khusus untuk lembaga militer republik Indonesia

5. or.id untuk segala macam organisasi yang tidak termasuk dalam kategori "ac.id", "co.id", "go.id"
6. war.net.id untuk industri warung internet di Indonesia
7. sch.id khusus untuk Lembaga Pendidikan yang menyelenggarakan pendidikan seperti SD, SMP, dan atau SMU
8. web.id ditujukan bagi badan udaha, organisasi ataupun perseorangan yang melakukan kegiatannya di *World Wide Web*

2.9 *Hosting*

Web *hosting* dapat diartikan sebagai ruangan yang terdapat dalam *hardisk* tempat menyimpan berbagai data, file-file, gambar, video, data email, statistik, *database*, dan lain sebagainnya yang akan ditampilkan di *website*. Besarnya data yang bisa dimasukkan tergantung dari besarnya web *hosting* yang disewa/dipunya, semakin besar web *hosting* semakin besar pula data yang dapat dimasukkan dan ditampilkan dalam *website*. Web *hosting* juga diperoleh dengan menyewa. Pengguna akan memperoleh *control panel* yang terproteksi dengan *username* dan *password* untuk administrasi websitenya.

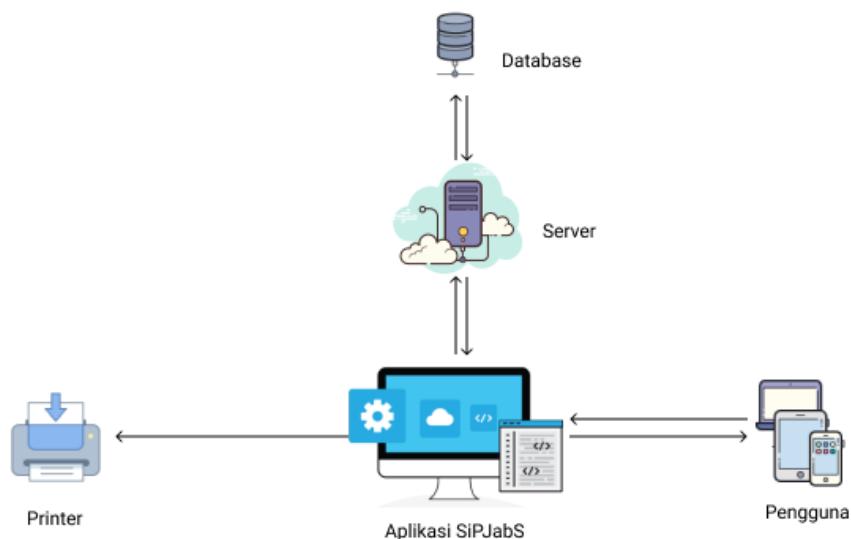
Besarnya *hosting* ditentukan ruangan *hardisk* dengan ukuran MB (Mega Byte) atau GB (Giga Byte). Lama penyewaan web *hosting* rata-rata dihitung per tahun. Penyewaan *hosting* dilakukan dari perusahaan-perusahaan penyewa web *hosting* yang banyak dijumpai baik di Indonesia maupun luar negeri. Lokasi peletakan pusat data (*datacenter*) web *hosting* bermacam-macam. Ada yang di Jakarta, Singapore, Inggris, Amerika, dll dengan harga sewa bervariasi (Ali Zaki, 2009) [7].

BAB III

ANALISIS KEBUTUHAN DAN PERANCANGAN

3.1 Sistem Arsitektur

Perancangan sistem arsitektur aplikasi **"SiPJabS : Sistem Pengawakan Jabatan Struktural"** dapat dilihat pada **Gambar 3.1.** berikut:



Gambar 3.1: Low Level Design

3.1.1 Gambaran Umum Sistem

Aplikasi **"SiPJabS : Sistem Pengawakan Jabatan Struktural"** merupakan aplikasi berbasis web yang memudahkan bagi perusahaan dalam pencarian seorang kandidat atau posisi yang kosong. Dalam pembuatan aplikasi ini dibutuhkan fitur *filtering* yang digunakan untuk pencarian kandidat baru, yang sesuai dengan ketentuan yang sudah ditetapkan oleh perusahaan. Sistem *filtering* dapat dilakukan setiap saat, untuk menggantikan pekerjaan lama yang telah berhenti dikarenakan pensiun, meninggal, mengundurkan diri atau diberhentikan karena suatu kebijakan tertentu.

Data-data pegawai yang berada di Universitas Telkom dapat dilihat dan data tersebut bersifat rahasia. Sehingga aplikasi "**SiP JabS : Sistem Pengawakan Jabatan Struktural**" hanya dapat diakses oleh orang tertentu. Aplikasi ini terdapat satu *user* yang dapat mengelola proses *filtering* dan satu *admin* yang mengelola infrastruktur *database* dan proyek *server* serta jaringan.

Sistem *filtering* pada aplikasi ini terbagi menjadi dua bagian, yang pertama merupakan *filtering* secara umum dengan isi *form* seperti jabatan minimal dan masa kerja. Yang kedua merupakan *filtering* secara khusus, dimana *user* dapat mencari kandidat dengan syarat yang lebih spesifik lagi untuk dijadikan pilihan, kemudian akan terdapat beberapa nama kandidat, apabila sudah menentukan pilihan dapat menekan tombol *button* pada nama yang akan dipilih dan akan masuk dalam kandidat sementara.

Apabila proses pencarian kandidat sudah ditemukan dengan salah satu proses *filtering* yang sudah dijelaskan diatas maka, proses selanjutnya akan masuk dalam pembuatan berita acara dan dapat dicetak berupa file pdf.

3.1.2 Target Pengguna Aplikasi

Aplikasi **SiP JabS** memiliki beberapa target pengguna diantaranya sebagai berikut:

1. User

User merupakan pegawai Direktorat Sumber Daya Manusia Universitas Telkom yang membutuhkan kandidat dengan proses *filtering* untuk mengisi posisi yang kosong atau digantikan.

2. Admin

Admin merupakan pegawai Direktorat Sumber Daya Manusia Universitas Telkom yang mengelola dan menyediakan data untuk proses *filtering*.

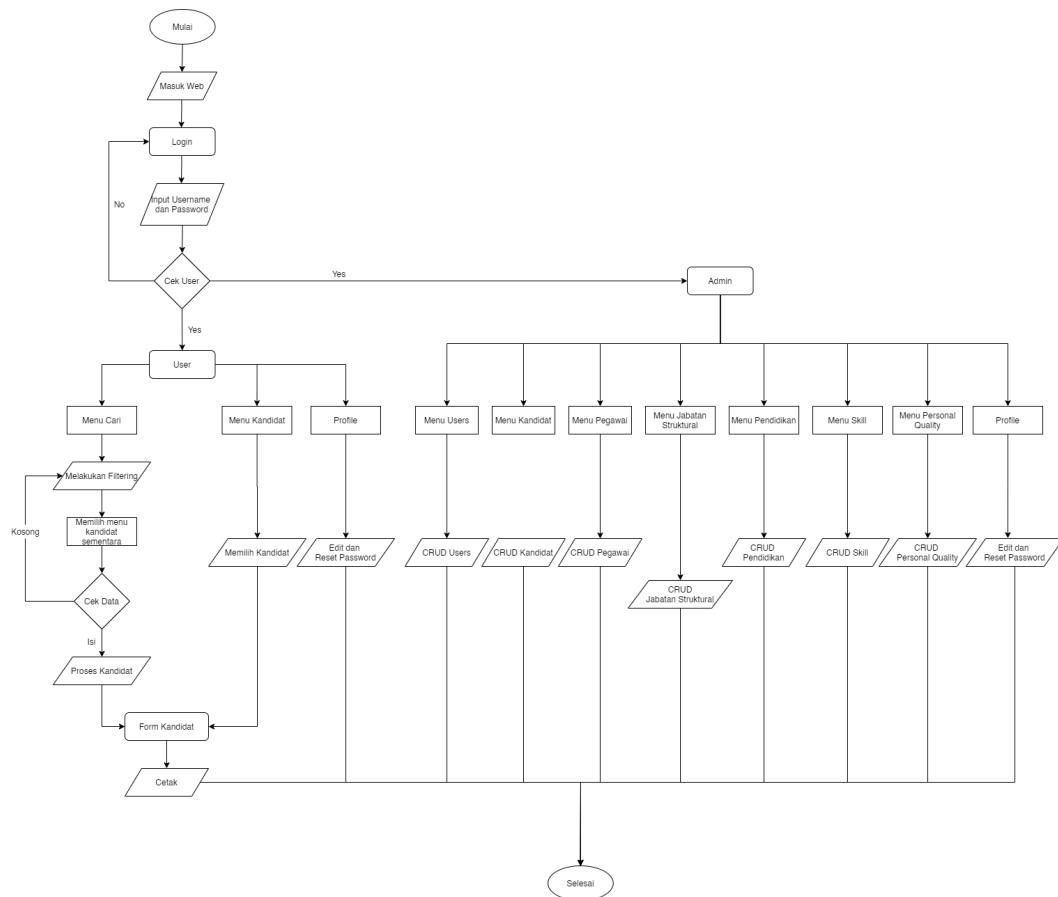
3.1.3 Spesifikasi Target Perangkat

Spesifikasi dari target perangkat untuk mengakses aplikasi **SiP JabS** adalah sebagai berikut:

1. Komputer atau laptop yang terhubung dengan koneksi internet dan dapat membuka web *browser*.
2. *Smartphone* atau tablet yang terhubung dengan koneksi internet dan dapat membuka web *browser*.

3.1.4 Diagram Alir Aplikasi

Dalam membangun aplikasi **SiP JabS**, dibutuhkan diagram alir untuk membantu *developer* dan pengguna dalam memahami sistem yang akan dibuat. Berikut merupakan *flowchart* aplikasi:



Gambar 3.2: Flowchart

Untuk flowchart pertama pengguna mengakses web dan terdapat halaman login, kemudian pengguna memasukkan *username* dan *password*. Jika akun *user* maka akan masuk halaman *user* dan jika akun *admin* maka akan masuk dalam halaman *admin*. Pada halaman *user* sendiri terdapat proses pencarian kandidat dengan sistem *filtering* dan *user* dapat mencetak pdf pada halaman data kandidat yang sudah terpilih. Pada halaman *admin* terdapat beberapa menu guna pemrosesan data yang disiapkan untuk melakukan proses *filtering*.

3.2 Kebutuhan Pengembangan Sistem

Dalam membangun aplikasi **SiP JabS**, dibutuhkan beberapa perangkat untuk mengimplementasikannya. Perangkat tersebut dibagi menjadi tiga, yaitu perangkat keras (*hardware*), perangkat lunak (*software*) dan Perangkat *server / hosting*. Adapun kebutuhan pengembangan sistem adalah sebagai berikut:

3.2.1 Kebutuhan Perangkat Keras (*Hardware*)

Hardware yang dibutuhkan dalam perancangan dan pembuatan aplikasi **SiP JabS** adalah sebagai berikut:

Tabel 3.1: Tabel Kebutuhan *Hardware*

No.	Perangkat Keras	Spesifikasi
1	Laptop MSI GL62M	<p><i>Processor</i> : Intel Core i7-7700HQ</p> <p><i>Operating System</i> : Windows 10 Education</p> <p>RAM : 8 GB</p> <p><i>Storage</i> : 128 GB SSD + 1 TB Hardisk</p> <p><i>Graphics Card</i> : nVidia Geforce GTX 1050</p> <p><i>Display</i> : 15.6" FHD, Anti-Glare (1920 x 1080)</p>
2	Laptop HP Pavilion x360	<p><i>Processor</i> : Intel Core i3-6100U</p> <p><i>Operating System</i> : Windows 10 Home</p> <p>RAM : 12 GB</p> <p><i>Storage</i> : 500 GB Hardisk</p> <p><i>Graphics Card</i> : Intel HD 520 Graphics</p> <p><i>Display</i> : 13.3" HD, Touch Screen</p>

3.2.2 Kebutuhan Perangkat Lunak (*Software*)

Software yang dibutuhkan dalam perancangan dan pembuatan aplikasi **SiPJabS** adalah sebagai berikut:

Tabel 3.2: Tabel Kebutuhan *Software*

No.	Perangkat Lunak	Kegunaan
1	Visual Studio Code	<i>Text editor</i> untuk menuliskan <i>coding</i> aplikasi
2	XAMPP	Sebagai <i>server</i> yang berdiri sendiri, yang terdiri atas program <i>Apache HTTP Server</i> , <i>MySQL database</i> , dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl
3	IBM Rational System Architect	Sebuah <i>software</i> untuk mendesain rancangan sistem aplikasi
4	Figma	Untuk mendesai <i>user interface</i> secara online
5	Microsoft Office Word	Untuk membuat dokumen dan laporan
6	TexStudio	Untuk membuat laporan dalam latex
7	Adobe Premier Pro	Editing vidio demo dan vidio promosi
8	Brave dan Mozilla Firefox	Web <i>browser</i>

3.2.3 Kebutuhan *Hosting*

Hosting yang dibutuhkan dalam perancangan dan pembuatan aplikasi **SiPJabS** adalah sebagai berikut:

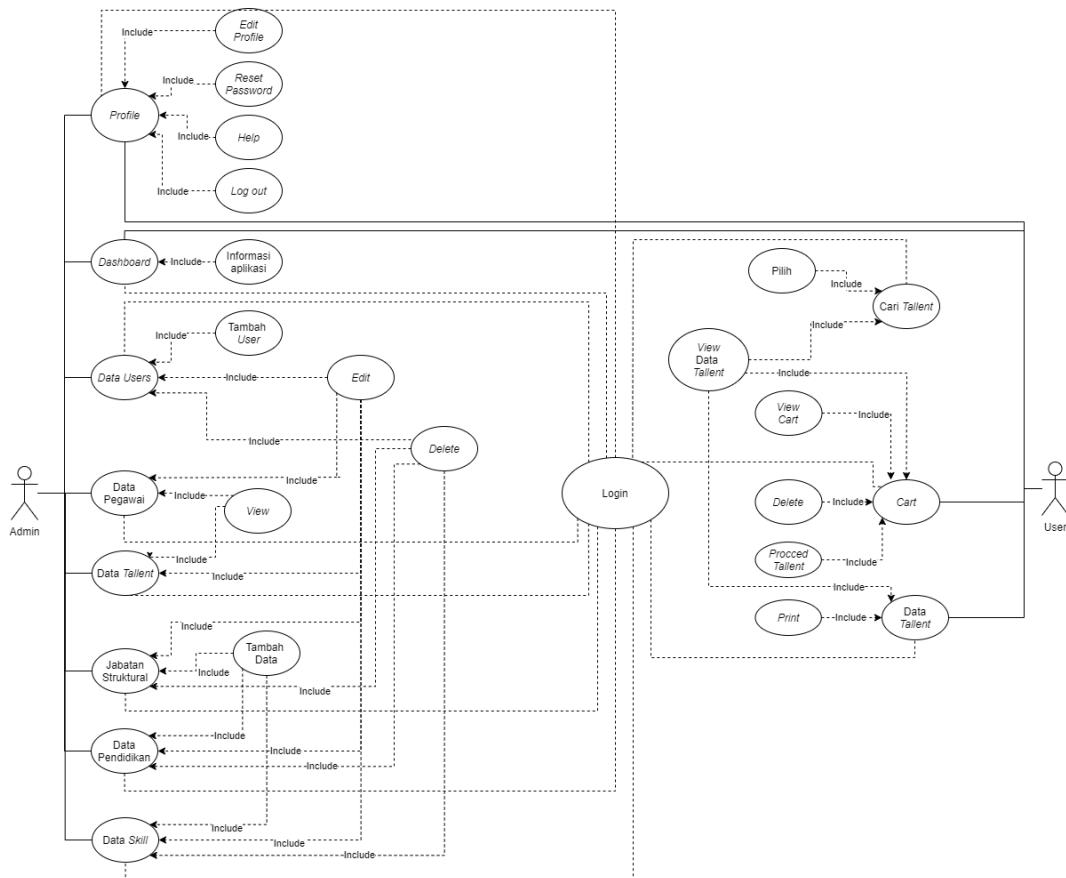
Tabel 3.3: Tabel Kebutuhan *Hosting*

No.	Server	Spesifikasi
1	Server Indonesia	<i>Storage</i> : 2 GB <i>RAM</i> : 1 GB <i>Bandwidth</i> : Unlimited <i>Processor</i> : 1 Core <i>Domain</i> : my.id

3.3 Perancangan Model Program

Perancangan model program dalam pembuatan aplikasi **SiP JabS** antara lain *Use Case Diagram*, *Use Case Scenario*, *Class Diagram*, *Entity Relationship Diagram* (ERD). Adapun perancangan model program adalah sebagai berikut:

3.3.1 Use Case Diagram



Gambar 3.3: Use Case Diagram

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo.

3.3.2 Use Case Skenario

Berikut merupakan *use case scenario* dalam pembuatan aplikasi **SiP JabS**:

1. Skenario *Login*

Nomor : SP-01

Nama Use Case : Melakukan *login*

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat menggunakan aplikasi

Deskripsi :

- Admin menginputkan *username* dan *password*
- Sistem akan mencocokkan data
- Sistem menampilkan halaman utama aplikasi

Tabel 3.4: Skenario *Login*

Aktor	Sistem
1. Menginputkan <i>username</i> dan <i>password</i>	
	2. Mencocokkan data
	3. Menampilkan halaman utama aplikasi

2. Skenario *Edit Profile*

Nomor : SP-02

Nama Use Case : Melakukan *edit profile*

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat melakukan edit pada *profile*

Deskripsi :

- Admin menuju ke halaman *profile*
- Sistem akan menampilkan halaman *profile*
- Admin memilih edit *profile*
- Sistem menampilkan *pop-up form edit profile*
- Admin menginputkan data

Tabel 3.5: Skenario Edit Profile

Aktor	Sistem
1. Menuju ke halaman <i>profile</i>	
	2. Menampilkan halaman <i>profile</i>
3. Memilih edit <i>profile</i>	
	4. Menampilkan <i>pop-up form edit profile</i>
5. Menginputkan data	
	6. Menyimpan data perubahan

3. Skenario Reset Password

Nomor : SP-03

Nama Use Case : Melakukan *reset password*

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat melakukan *reset password* pada *profile*

Deskripsi :

- Admin menuju ke halaman *profile*
- Sistem akan menampilkan halaman *profile*
- Admin memilih *reset password*
- Sistem menampilkan *pop-up reset password*
- Admin menginputkan *password* lama dan baru
- Sistem menyimpan data perubahan

Tabel 3.6: Skenario Reset Password

Aktor	Sistem
1. Menuju ke halaman <i>profile</i>	
	2. Menampilkan halaman <i>profile</i>
3. Memilih <i>reset password</i>	
	4. Menampilkan <i>pop-up reset password</i>
5. Menginputkan <i>password</i> lama dan baru	
	6. Menyimpan data perubahan

4. Skenario Tambah *Users*

Nomor : SP-04

Nama Use Case : Menambahkan *users*

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat menambahkan *users*

Deskripsi :

- Admin menuju ke halaman data *users*
- Sistem akan menampilkan halaman data *users*
- Admin memilih tambah *users*
- Sistem menampilkan halaman *form* tambah *users*
- Admin menginputkan data
- Sistem menyimpan data
- Sistem menampilkan *pop-up* tanda berhasil menambahkan *users*

Tabel 3.7: Skenario Tambah *Users*

Aktor	Sistem
1. Menuju ke halaman data <i>users</i>	
	2. Menampilkan halaman data <i>users</i>
3. Memilih tambah <i>users</i>	
	4. Menampilkan halaman <i>form</i> tambah <i>users</i>
5. Menginputkan data	
	6. Menyimpan data
	7. Menampilkan <i>pop-up</i> tanda berhasil menambahkan <i>users</i>

5. Skenario Edit *Users*

Nomor : SP-05

Nama Use Case : Melakukan edit data *users*

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat mengedit data *users*

Deskripsi :

- Admin menuju ke halaman data *users*
- Sistem akan menampilkan halaman data *users*
- Admin memilih edit pada salah satu *users*
- Sistem menampilkan *pop-up form edit users*
- Admin menginputkan data
- Sistem menyimpan data perubahan
- Sistem menampilkan *pop-up* tanda berhasil mengedit data

Tabel 3.8: Skenario Edit *Users*

Aktor	Sistem
1. Menuju ke halaman data <i>users</i>	
	2. Menampilkan halaman data <i>users</i>
3. Memilih <i>edit</i> pada salah satu <i>users</i>	
	4. Menampilkan <i>pop-up form edit users</i>
5. Menginputkan data	
	7. Menyimpan data
	8. Menampilkan <i>pop-up</i> tanda berhasil edit <i>users</i>

6. Skenario *Delete Users*

Nomor : SP-06

Nama Use Case : Melakukan *delete* data *users*

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat *delete* data *users*

Deskripsi :

- Admin menuju ke halaman data *users*
- Sistem akan menampilkan halaman data *users*
- Admin memilih *delete* pada salah satu *users*
- Sistem menampilkan *pop-up* tanda berhasil hapus data

Tabel 3.9: Skenario *Delete Users*

Aktor	Sistem
1. Menuju ke halaman data <i>users</i>	
	2. Menampilkan halaman data <i>users</i>
3. Memilih <i>delete</i> pada salah satu <i>users</i>	
	4. Menampilkan <i>pop-up</i> tanda berhasil <i>delete</i> data

7. Skenario *View Pegawai*

Nomor : SP-07

Nama Use Case : Melakukan *view* data pegawai

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat melihat data pegawai

Deskripsi :

- Admin menuju ke halaman data pegawai
- Sistem akan menampilkan halaman data pegawai
- Admin memilih *view* pada salah satu pegawai
- Sistem menampilkan *pop-up* detail pegawai
- Admin dapat melihat data detail pegawai

Tabel 3.10: Skenario View Pegawai

Aktor	Sistem
1. Menuju ke halaman data pegawai	
	2. Menampilkan halaman data pegawai
3. Memilih <i>view</i> pada salah satu data pegawai	
	4. Menampilkan <i>pop-up</i> detail pegawai
5. Melihat data detail pegawai	

8. Skenario Tambah Pegawai

Nomor : SP-08

Nama Use Case : Melakukan tambah data pegawai

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat menambahkan data pegawai

Deskripsi :

- Admin menuju ke halaman data pegawai
- Sistem akan menampilkan halaman data pegawai
- Admin memilih tambah data pegawai
- Sistem menampilkan *form* tambah pegawai
- Admin menginputkan data
- Sistem menyimpan data
- Sistem menampilkan *pop-up* berhasil menambahkan data

Tabel 3.11: Skenario Tambah Pegawai

Aktor	Sistem
1. Menuju ke halaman data pegawai	
	2. Menampilkan halaman data pegawai
3. Memilih tambah data pegawai	
	4. Menampilkan <i>form</i> tambah pegawai
5. Menginputkan data	
	6. Menyimpan data
	7. Menampilkan <i>pop-up</i> berhasil menambahkan data

9. Skenario Edit Pegawai

Nomor : SP-09

Nama Use Case : Melakukan edit data pegawai

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat mengedit data pegawai

Deskripsi :

- Admin menuju ke halaman data pegawai
- Sistem akan menampilkan halaman data pegawai
- Admin memilih edit pada salah satu pegawai
- Sistem menampilkan *form* edit data pegawai
- Admin menginputkan data
- Sistem menyimpan data perubahan
- Sistem menampilkan *pop-up* berhasil merubah data

Tabel 3.12: Skenario Edit Pegawai

Aktor	Sistem
1. Menuju ke halaman data pegawai	
	2. Menampilkan halaman data pegawai
3. Memilih edit pada salah satu pegawai	
	4. Menampilkan <i>form</i> edit data pegawai
5. Menginputkan data	
	6. Menyimpan data perubahan
	7. Menampilkan <i>pop-up</i> berhasil merubah data

10. Skenario *View* Kandidat

Nomor : SP-10

Nama Use Case : Melakukan *view* data kandidat

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat melihat data kandidat

Deskripsi :

- Admin menuju ke halaman data kandidat
- Sistem akan menampilkan halaman data kandidat
- Admin memilih *view* pada salah satu kandidat
- Sistem menampilkan *pop-up* detail data kandidat
- Admin dapat melihat data detail kandidat

Tabel 3.13: Skenario *View* Kandidat

Aktor	Sistem
1. Menuju ke halaman data kandidat	
	2. Menampilkan halaman data kandidat
3. Memilih <i>view</i> pada salah satu kandidat	
	4. Menampilkan <i>pop-up</i> detail data kandidat
5. Melihat data detail kandidat	

11. Skenario *Print Form* Kandidat

Nomor : SP-11

Nama Use Case : Melakukan *print form* data kandidat

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat mencetak *form* data kandidat

Deskripsi :

- Admin menuju ke halaman data kandidat
- Sistem akan menampilkan halaman data kandidat
- Admin memilih *print pdf* pada salah satu data kandidat

Tabel 3.14: Skenario *Print Form* Kandidat

Aktor	Sistem
1. Menuju ke halaman data kandidat	
	2. Menampilkan halaman data kandidat
3. Memilih <i>print pdf</i> pada salah satu data kandidat	

12. Skenario Tambah Unit Kerja

Nomor : SP-12

Nama Use Case : Menambahkan data unit kerja

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat menambahkan data unit kerja

Deskripsi :

- Admin menuju ke halaman data unit kerja
- Sistem akan menampilkan halaman data unit kerja
- Admin memilih tambah unit kerja
- Sistem menampilkan *pop-up* form tambah unit kerja
- Admin menginputkan data
- Sistem menyimpan data
- Sistem menampilkan *pop-up* tanda data berhasil ditambahkan

Tabel 3.15: Skenario Tambah Unit Kerja

Aktor	Sistem
1. Menuju ke halaman data unit kerja	
	2. Menampilkan halaman data unit kerja
3. Memilih tambah unit kerja	
	4. Menampilkan <i>pop-up</i> form tambah unit kerja
5. Menginputkan data	
	6. Menyimpan data
	7. Menampilkan <i>pop-up</i> tanda berhasil menambahkan data

13. Skenario Edit Unit Kerja

Nomor : SP-13

Nama Use Case : Melakukan edit data unit kerja

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat mengedit data unit kerja

Deskripsi :

- Admin menuju ke halaman data unit kerja
- Sistem akan menampilkan halaman data unit kerja
- Admin memilih edit pada suatu data unit kerja
- Sistem menampilkan *pop-up form* edit unit kerja
- Admin menginputkan data
- Sistem menyimpan data
- Sistem menampilkan *pop-up* tanda berhasil edit data

Tabel 3.16: Skenario Edit Unit Kerja

Aktor	Sistem
1. Menuju ke halaman data unit kerja	
3. Memilih edit pada suatu data unit kerja	2. Menampilkan halaman data unit kerja
5. Menginputkan data	4. Menampilkan <i>pop-up form</i> edit unit kerja
	6. Menyimpan data
	7. Menampilkan <i>pop-up</i> tanda berhasil edit data

14. Skenario View Unit Kerja

Nomor : SP-14

Nama Use Case : Melihat pegawai yang memiliki data unit kerja

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat melihat data pegawai sesuai unit kerja

Deskripsi :

- Admin menuju ke halaman data unit kerja
- Sistem akan menampilkan halaman data unit kerja
- Admin memilih *view* pada suatu data unit kerja
- Sistem menampilkan *pop-up* data pegawai yang sesuai unit kerja
- Admin dapat melihat data pegawai yang sesuai unit kerja

Tabel 3.17: Skenario View Unit Kerja

Aktor	Sistem
1. Menuju ke halaman data unit kerja	
	2. Menampilkan halaman data unit kerja
3. Memilih <i>view</i> pada suatu data unit kerja	
	4. Menampilkan <i>pop-up</i> data pegawai yang sesuai unit kerja
5. Melihat data pegawai yang sesuai unit kerja	

15. Skenario Tambah Jabatan

Nomor : SP-15

Nama Use Case : Menambahkan data jabatan

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat menambahkan data jabatan

Deskripsi :

- Admin menuju ke halaman data jabatan
- Sistem akan menampilkan halaman data jabatan

- Admin memilih tambah jabatan
- Sistem menampilkan *pop-up* tambah jabatan
- Admin menginputkan data
- Sistem menyimpan data
- Sistem menampilkan *pop-up* tanda berhasil ditambahkan

Tabel 3.18: Skenario Tambah Jabatan

Aktor	Sistem
1. Menuju ke halaman data jabatan	
	2. Menampilkan halaman data jabatan
3. Memilih tambah jabatan	
	4. Menampilkan <i>pop-up</i> tambah jabatan
5. Menginputkan data	
	6. Menyimpan data
	7. Menampilkan <i>pop-up</i> tanda berhasil menambahkan data

16. Skenario Edit Jabatan

Nomor : SP-16

Nama Use Case : Melakukan edit data jabatan

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat mengedit data jabatan

Deskripsi :

- Admin menuju ke halaman data jabatan
- Sistem akan menampilkan halaman data jabatan
- Admin memilih edit pada suatu data jabatan
- Sistem menampilkan *pop-up form* edit jabatan
- Admin menginputkan data
- Sistem menyimpan data perubahan
- Sistem menampilkan *pop-up* tanda berhasil edit data

Tabel 3.19: Skenario Edit Jabatan

Aktor	Sistem
1. Menuju ke halaman data jabatan	
	2. Menampilkan halaman data jabatan
3. Memilih edit pada suatu jabatan	
	4. Menampilkan <i>pop-up form</i> edit jabatan
5. Menginputkan data	
	6. Menyimpan data perubahan
	7. Menampilkan <i>pop-up</i> tanda berhasil edit data

17. Skenario *View* Jabatan

Nomor : SP-17

Nama Use Case : Melihat pegawai yang memiliki data jabatan

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat melihat data pegawai sesuai jabatan

Deskripsi :

- Admin menuju ke halaman data jabatan
- Sistem akan menampilkan halaman data jabatan
- Admin memilih *view* pada suatu data jabatan
- Sistem menampilkan *pop-up* data pegawai yang sesuai jabatan

Tabel 3.20: Skenario *View* Jabatan

Aktor	Sistem
1. Menuju ke halaman data jabatan	
	2. Menampilkan halaman data jabatan
3. Memilih <i>view</i> pada suatu data jabatan	
	4. Menampilkan <i>pop-up</i> data pegawai yang sesuai jabatan

18. Skenario Tambah Unit Bagian

Nomor : SP-18

Nama Use Case : Menambahkan data unit bagian

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat menambahkan data unit bagian

Deskripsi :

- Admin menuju ke halaman data unit bagian
- Sistem akan menampilkan halaman data unit bagian
- Admin memilih tambah unit bagian
- Sistem menampilkan *pop-up* form tambah unit bagian
- Admin menginputkan data
- Sistem menyimpan data
- Sistem menampilkan *pop-up* tanda berhasil ditambahkan

Tabel 3.21: Skenario Tambah Unit Bagian

Aktor	Sistem
1. Menuju ke halaman data unit bagian	
	2. Menampilkan halaman data unit bagian
3. Memilih tambah unit bagian	
	4. Menampilkan <i>pop-up</i> form tambah unit bagian
5. Menginputkan data	
	6. Menyimpan data
	7. Menampilkan <i>pop-up</i> tanda berhasil menambahkan data

19. Skenario Edit Unit Bagian

Nomor : SP-19

Nama Use Case : Melakukan edit data unit bagian

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat mengedit data unit bagian

Deskripsi :

- Admin menuju ke halaman data unit bagian
- Sistem akan menampilkan halaman data unit bagian
- Admin memilih edit pada suatu data unit bagian
- Sistem menampilkan *pop-up form* edit unit bagian
- Admin menginputkan data
- Sistem menyimpan data perubahan
- Sistem menampilkan *pop-up* tanda berhasil di edit

Tabel 3.22: Skenario Edit Unit Bagian

Aktor	Sistem
1. Menuju ke halaman data unit bagian	
	2. Menampilkan halaman data unit bagian
3. Memilih edit pada suatu data unit bagian	
	4. Menampilkan <i>pop-up form</i> edit unit bagian
5. Menginputkan data	
	6. Menyimpan data perubahan
	7. Menampilkan <i>pop-up</i> tanda berhasil di edit

20. Skenario View Unit Bagian

Nomor : SP-20

Nama Use Case : Melihat pegawai yang memiliki data unit bagian

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat melihat data pegawai sesuai unit bagian

Deskripsi :

- Admin menuju ke halaman data unit bagian
- Sistem akan menampilkan halaman data unit bagian
- Admin memilih *view* pada suatu data unit bagian
- Sistem menampilkan *pop-up* data pegawai yang sesuai unit bagian

Tabel 3.23: Skenario View Unit Bagian

Aktor	Sistem
1. Menuju ke halaman data unit bagian	
	2. Menampilkan halaman data unit bagian
3. Memilih <i>view</i> pada suatu data unit bagian	
	4. Menampilkan <i>pop-up</i> data pegawai yang sesuai unit bagian

21. Skenario Tambah Jabatan Struktural

Nomor : SP-21

Nama Use Case : Menambahkan data jabatan struktural

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat menambahkan data jabatan struktural

Deskripsi :

- Admin menuju ke halaman data jabatan struktural
- Sistem akan menampilkan halaman data jabatan struktural
- Admin memilih tambah jabatan struktural
- Sistem menampilkan *pop-up form* tambah jabatan struktural
- Admin menginputkan data
- Sistem menyimpan data
- Sistem menampilkan *pop-up* tanda berhasil menambahkan data

Tabel 3.24: Skenario Tambah Jabatan Struktural

Aktor	Sistem
1. Menuju ke halaman data jabatan struktural	
	2. Menampilkan halaman data jabatan struktural
3. Memilih tambah data jabatan struktural	
	4. Menampilkan <i>pop-up form</i> tambah jabatan struktural
5. Menginputkan data	
	6. Menyimpan data
	7. Menampilkan <i>pop-up</i> tanda berhasil menambahkan data

22. Skenario Edit Jabatan Struktural

Nomor : SP-22

Nama Use Case : Melakukan edit data jabatan struktural

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat mengedit data jabatan struktural

Deskripsi :

- Admin menuju ke halaman data jabatan struktural
- Sistem akan menampilkan halaman data jabatan struktural
- Admin memilih edit pada suatu data jabatan struktural
- Sistem menampilkan *pop-up form* edit jabatan struktural
- Admin menginputkan data
- Sistem menyimpan data perubahan
- Sistem menampilkan *pop-up* tanda berhasil diedit

Tabel 3.25: Skenario Edit Jabatan Struktural

Aktor	Sistem
1. Menuju halaman data jabatan struktural	
	2. Menampilkan halaman data jabatan struktural
3. Memilih edit pada suatu jabatan struktural	
	4. Menampilkan <i>pop-up form</i> edit jabatan struktural
5. Menginputkan data	
	6. Menyimpan data perubahan
	7. Menampilkan <i>pop-up</i> tanda berhasil edit data

23. Skenario *View* Jabatan Struktural

Nomor : SP-23

Nama Use Case : Melihat pegawai yang memiliki data jabatan struktural

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat melihat data pegawai sesuai jabatan struktural

Deskripsi :

- Admin menuju ke halaman data jabatan struktural
- Sistem akan menampilkan halaman data jabatan struktural
- Admin memilih *view* pada suatu data jabatan struktural
- Sistem menampilkan *pop-up* data pegawai yang sesuai jabatan struktural

Tabel 3.26: Skenario *View* Jabatan Struktural

Aktor	Sistem
1. Menuju ke halaman data jabatan struktural	
	2. Menampilkan halaman data jabatan struktural
3. Memilih <i>view</i> pada suatu data jabatan struktural	
	4. Menampilkan <i>pop-up</i> data pegawai yang sesuai jabatan struktural

24. Skenario Tambah Pendidikan

Nomor : SP-24

Nama Use Case : Menambahkan data pendidikan

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat menambahkan data pendidikan

Deskripsi :

- Admin menuju ke halaman data pendidikan
- Sistem akan menampilkan halaman data pendidikan
- Admin memilih tambah pendidikan
- Sistem menampilkan *pop-up form* tambah pendidikan
- Admin menginputkan data
- Sistem menyimpan data
- Sistem menampilkan *pop-up* tanda berhasil ditambahkan

Tabel 3.27: Skenario Tambah Pendidikan

Aktor	Sistem
1. Menuju ke halaman data pendidikan	
	2. Menampilkan halaman data pendidikan
3. Memilih tambah pendidikan	
	4. Menampilkan <i>pop-up form</i> tambah pendidikan
5. Menginputkan data	
	6. Menyimpan data
	7. Menampilkan <i>pop-up</i> tanda berhasil menambahkan data

25. Skenario Edit Pendidikan

Nomor : SP-25

Nama Use Case : Melakukan edit data pendidikan

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat mengedit data pendidikan

Deskripsi :

- Admin menuju ke halaman data pendidikan
- Sistem akan menampilkan halaman data pendidikan
- Admin memilih edit pada suatu data pendidikan
- Sistem menampilkan *pop-up form* edit pendidikan
- Admin menginputkan data
- Sistem menyimpan data perubahan
- Sistem menampilkan *pop-up* tanda berhasil di edit

Tabel 3.28: Skenario Edit Pendidikan

Aktor	Sistem
1. Menuju ke halaman data pendidikan	
	2. Menampilkan halaman data pendidikan
3. Memilih edit pada suatu pendidikan	
	4. Menampilkan <i>pop-up form</i> edit pendidikan
5. Menginputkan data	
	6. Menyimpan data perubahan
	7. Menampilkan <i>pop-up</i> tanda berhasil edit data

26. Skenario View Pendidikan

Nomor : SP-26

Nama Use Case : Melihat pegawai yang memiliki data pendidikan

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat melihat data pegawai sesuai pendidikan

Deskripsi :

- Admin menuju ke halaman data pendidikan
- Sistem akan menampilkan halaman data pendidikan
- Admin memilih *view* pada suatu data pendidikan
- Sistem menampilkan *pop-up* data pegawai sesuai pendidikan

Tabel 3.29: Skenario View Pendidikan

Aktor	Sistem
1. Menuju ke halaman data pendidikan	
3. Memilih <i>view</i> pada suatu data pendidikan	2. Menampilkan halaman data pendidikan
	4. Menampilkan <i>pop-up</i> data pegawai sesuai pendidikan

27. Skenario Tambah Skill

Nomor : SP-27

Nama Use Case : Menambahkan data skill

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat menambahkan data skill

Deskripsi :

- Admin menuju ke halaman data skill
- Sistem akan menampilkan halaman data skill
- Admin memilih tambah skill
- Sistem menampilkan *pop-up form* tambah skill
- Admin menginputkan data
- Sistem menyimpan data
- Sistem menampilkan *pop-up* tanda berhasil ditambahkan

Tabel 3.30: Skenario Tambah Skill

Aktor	Sistem
1. Menuju ke halaman data skill	
3. Memilih tambah skill	2. Menampilkan halaman data skill
5. Menginputkan data	4. Menampilkan <i>pop-up form</i> tambah skill
	6. Menyimpan data
	7. Menampilkan <i>pop-up</i> tanda berhasil menambahkan data

28. Skenario Edit Skill

Nomor : SP-28

Nama Use Case : Melakukan edit data *skill*

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat mengedit data *skill*

Deskripsi :

- Admin menuju ke halaman data *skill*
- Sistem akan menampilkan halaman data *skill*
- Admin memilih edit pada suatu data *skill*
- Sistem menampilkan *pop-up form* edit *skill*
- Admin menginputkan data
- Sistem menyimpan data perubahan
- Sistem menampilkan *pop-up* tanda berhasil diedit

Tabel 3.31: Skenario Edit Skill

Aktor	Sistem
1. Menuju ke halaman data <i>skill</i>	
	2. Menampilkan halaman data <i>skill</i>
3. Memilih edit pada suatu data <i>skill</i>	
	4. Menampilkan <i>pop-up form</i> edit <i>skill</i>
5. Menginputkan data	
	6. Menyimpan data perubahan
	7. Menampilkan <i>pop-up</i> tanda berhasil edit data

29. Skenario View Skill

Nomor : SP-29

Nama Use Case : Melihat pegawai yang memiliki data *skill*

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat melihat data pegawai yang sesuai *skill*

Deskripsi :

- Admin menuju ke halaman data *skill*
- Sistem akan menampilkan halaman data *skill*
- Admin memilih *view* pada suatu data *skill*
- Sistem menampilkan *pop-up* data pegawai yang sesuai *skill*

Tabel 3.32: Skenario *View Skill*

Aktor	Sistem
1. Menuju ke halaman data <i>skill</i>	
	2. Menampilkan halaman data <i>skill</i>
3. Memilih view pada suatu data <i>skill</i>	
	4. Menampilkan <i>pop-up</i> data pegawai yang sesuai <i>skill</i>

30. Skenario Tambah *Personal Quality*

Nomor : SP-30

Nama Use Case : Menambahkan data *personal quality*

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat menambahkan data *personal quality*

Deskripsi :

- Admin menuju ke halaman data *personal quality*
- Sistem akan menampilkan halaman data *personal quality*
- Admin memilih tambah *personal quality*
- Sistem menampilkan *pop-up form* tambah *personal quality*
- Admin menginputkan data
- Sistem menyimpan data
- Sistem menampilkan *pop-up* tanda berhasil ditambahkan

Tabel 3.33: Skenario Tambah *Personal Quality*

Aktor	Sistem
1. Menuju ke halaman data <i>personal quality</i>	
	2. Menampilkan halaman data <i>personal quality</i>
3. Memilih tambah <i>personal quality</i>	
	4. Menampilkan <i>pop-up form</i> tambah <i>personal quality</i>
5. Menginputkan data	
	6. Menyimpan data
	7. Menampilkan <i>pop-up</i> tanda berhasil menambahkan data

31. Skenario Edit *Personal Quality*

Nomor : SP-31

Nama Use Case : Melakukan edit data *personal quality*

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat mengedit data *personal quality*

Deskripsi :

- Admin menuju ke halaman data *personal quality*
- Sistem akan menampilkan halaman data *personal quality*
- Admin memilih edit pada suatu data *personal quality*
- Sistem menampilkan *pop-up form* edit *personal quality*
- Admin menginputkan data
- Sistem menyimpan data perubahan
- Sistem menampilkan *pop-up* tanda berhasil dieredit

Tabel 3.34: Skenario Edit Personal Quality

Aktor	Sistem
1. Menuju ke halaman data <i>personal quality</i>	
	2. Menampilkan halaman data <i>personal quality</i>
3. Memilih edit pada suatu <i>personal quality</i>	
	4. Menampilkan <i>pop-up form</i> edit <i>personal quality</i>
5. Menginputkan data	
	6. Menyimpan data perubahan
	7. Menampilkan <i>pop-up</i> tanda berhasil edit data

32. Skenario View Personal Quality

Nomor : SP-32

Nama Use Case : Melihat pegawai yang memiliki data *personal quality*

Aktor : Admin

Tipe : *Primary*

Tujuan : Admin dapat melihat data pegawai sesuai *personal quality*

Deskripsi :

- Admin menuju ke halaman data *personal quality*
- Sistem akan menampilkan halaman data *personal quality*
- Admin memilih *view* pada suatu data *personal quality*
- Sistem menampilkan *pop-up* data pegawai yang sesuai *personal quality*

Tabel 3.35: Skenario View Personal Quality

Aktor	Sistem
1. Menuju ke halaman data <i>personal quality</i>	
	2. Menampilkan halaman data <i>personal quality</i>
3. Memilih <i>view</i> pada suatu <i>personal quality</i>	
	4. Menampilkan <i>pop-up</i> data pegawai yang sesuai <i>personal quality</i>

33. Skenario Cari Kandidat

Nomor : SP-33

Nama Use Case : Mencari kandidat baru

Aktor : *User*

Tipe : *Primary*

Tujuan : *User* dapat mencari kandidat baru

Deskripsi :

- *User* menuju ke halaman cari kandidat
- Sistem akan menampilkan halaman cari kandidat
- *User* mengisi persyaratan umum
- Sitem menampilkan daftar data kandidat
- *User* melakukan *filtering*
- Sitem menampilkan daftar data kandidat sesuai *filtering*

Tabel 3.36: Skenario Cari Kandidat

Aktor	Sistem
1. Menuju ke halaman cari kandidat	
	2. Menampilkan halaman cari kandidat
3. Mengisi persyaratan umum	
	4. Menampilkan daftar data kandidat
5. Melakukan <i>filtering</i>	6. Menampilkan daftar data kandidat sesuai <i>filtering</i>

34. Skenario Memilih Kandidat

Nomor : SP-34

Nama Use Case : Memilih kandidat

Aktor : *User*

Tipe : *Primary*

Tujuan : *User* dapat melakukan pemilihan kandidat

Deskripsi :

- *User* menuju ke halaman cari kandidat
- Sistem akan menampilkan halaman cari kandidat
- *User* mengisi persyaratan umum

- Sitem menampilkan daftar data kandidat
- *User* melakukan *filtering*
- Sitem menampilkan daftar data kandidat sesuai *filtering*
- *User* memilih kandidat
- Sistem menampilkan *pop-up* tanda pemilihan kandidat berhasil

Tabel 3.37: Skenario Memilih Kandidat

Aktor	Sistem
1. Menuju ke halaman cari kandidat	
	2. Menampilkan halaman cari kandidat
3. Mengisi persyaratan umum	
	4. Menampilkan daftar data kandidat
5. Melakukan <i>filtering</i>	
	6. Menampilkan daftar data kandidat sesuai <i>filtering</i>
7. Memilih kandidat	
	8. Menampilkan <i>pop-up</i> tanda berhasil memilih kandidat

35. Skenario *View* Kandidat Sementara

Nomor : SP-35

Nama Use Case : Melakukan *view* kandidat sementara

Aktor : *User*

Tipe : *Primary*

Tujuan : *User* dapat melihat data kandidat sementara

Deskripsi :

- *User* menuju ke *icon* orang
- Sistem akan menampilkan *dropdown* kandidat sementara
- *User* memilih *view* kandidat sementara
- Sitem menampilkan data kandidat sementara
- *User* melihat data kandidat sementara

Tabel 3.38: Skenario View Kandidat Sementara

Aktor	Sistem
1. Menuju ke <i>icon</i> orang	
	2. Menampilkan halaman <i>dropdown</i> kandidat sementara
3. Memilih <i>view</i> kandidat sementara	
	4. Menampilkan data kandidat sementara
5. Melihat data kandidat sementara	

36. Skenario Memproses Kandidat Sementara

Nomor : SP-36

Nama Use Case : Memproses kandidat sementara

Aktor : *User*

Tipe : *Primary*

Tujuan : *User* dapat memproses kandidat sementara

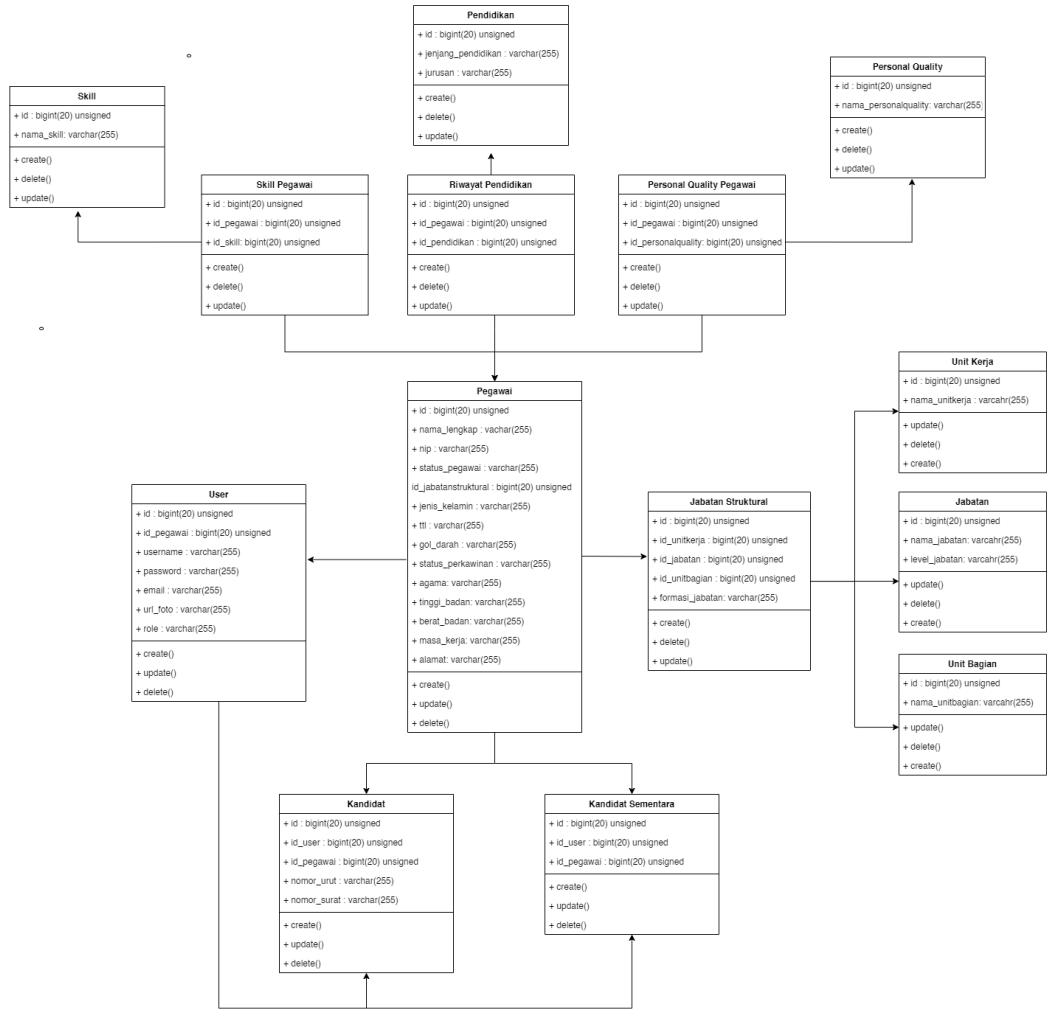
Deskripsi :

- User menuju ke *icon* orang
- Sistem akan menampilkan *dropdown* kandidat sementara
- User memilih *view* kandidat sementara
- Sitem menampilkan data kandidat sementara
- User memilih proses kandidat sementara
- Sistem menampilkan *pop-up* berhasil memproses kandidat sementara

Tabel 3.39: Skenario Proses Kandidat Sementara

Aktor	Sistem
1. Menuju ke <i>icon</i> orang	
	2. Menampilkan halaman <i>dropdown</i> kandidat sementara
3. Memilih <i>view</i> kandidat sementara	
	4. Menampilkan data kandidat sementara
5. Memilih proses kandidat sementara	
	6. Menampilkan <i>pop-up</i> berhasil memproses kandidat sementara

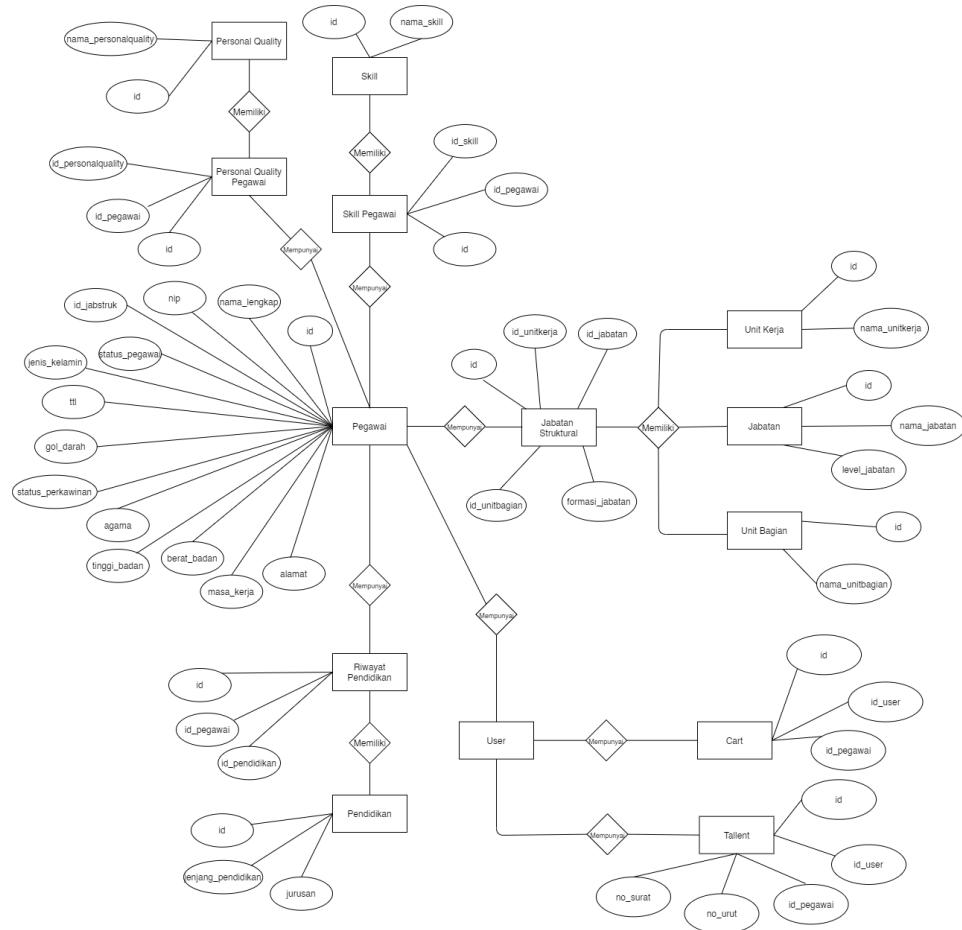
3.3.3 Class Diagram



Gambar 3.4: Class Diagram

Terdapat banyak relasi antar tabel pada gambar diatas, terutama pada tabel pegawai yang hampir mempunyai relasi di setiap tabel. Data pegawai harus sedemikian lengkap guna menunjang proses *filtering*. Data pegawai berupa data jabatan, riwayat pendidikan, *skill*, *personal quality* dan data pribadi lainnya.

3.3.4 Entity Relationship Diagram



Gambar 3.5: ERD

3.4 Perancangan Aplikasi

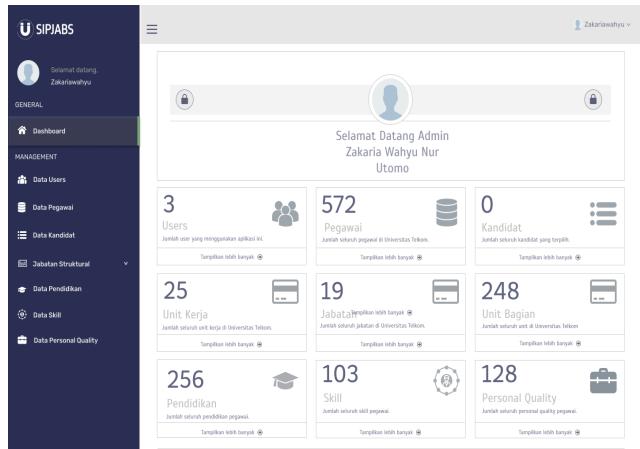
Dalam perancangan aplikasi **SiP JabS** diperlukan perancangan antar muka dan perancangan design level tinggi. Perancangan antar muka akan menjelaskan gambaran awal **developer** sebelum masuk pada bagian *front-end*. Sedangkan perancangan *design* level tinggi berguna untuk mengingatkan **developer** tentang sistem kerja pada aplikasi yang akan dibuat.

3.4.1 Perancangan Antar Muka

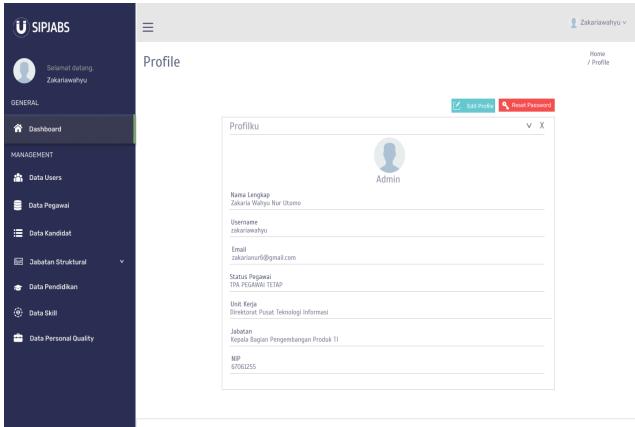
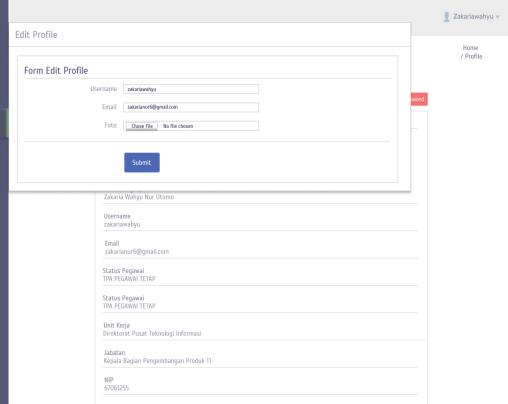
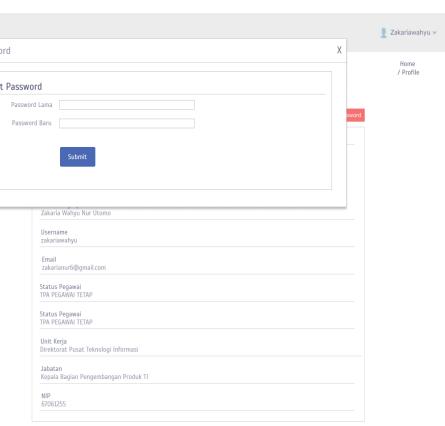
Pada tahap kebutuhan antar muka terdapat gambaran mengenai aplikasi **"SiP-JabS: Sistem Pengawakan Jabatan Struktural"**, berikut merupakan mockup dari aplikasi **SiP JabS** yang sudah dibuat.

3.4.1.1 Perancangan Antar Muka Admin

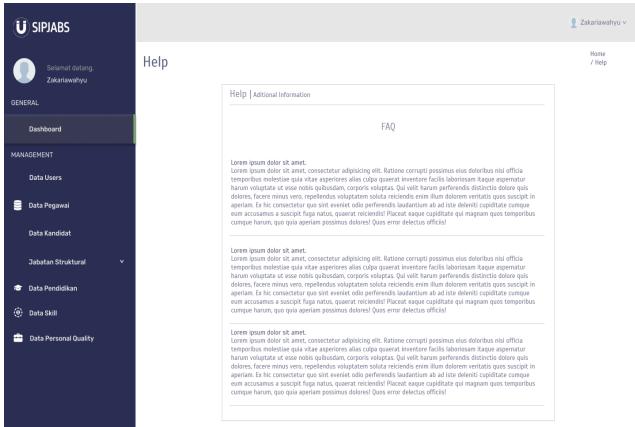
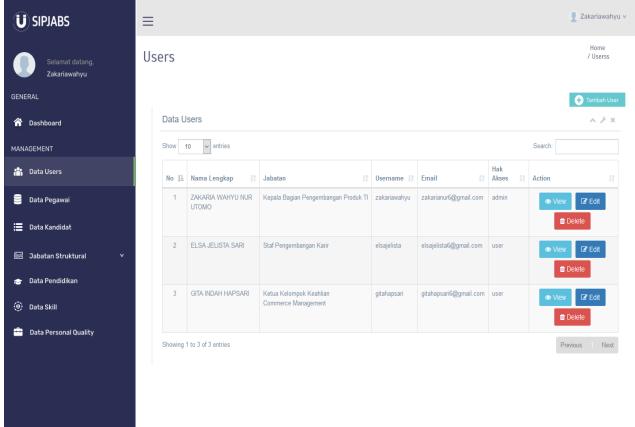
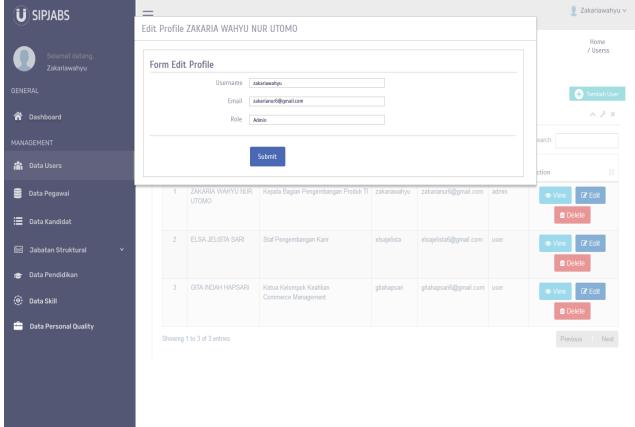
Tabel 3.40: Tabel Perancangan Antar Muka Admin

No	Gambar	Keterangan
1.		Halaman <i>login</i> merupakan tampilan awal apabila admin membuka aplikasi SIPJABS , admin dapat menginputkan <i>username</i> dan <i>password</i> untuk melakukan <i>login</i> .
2.		Didalam <i>dashboard</i> admin terdapat jumlah <i>users</i> , jumlah pegawai, kandidat yang sudah dipilih, unit kerja, jabatan, unit bagian, pendidikan, skill dan personal quality yang dimiliki para pegawai Universitas Telkom.

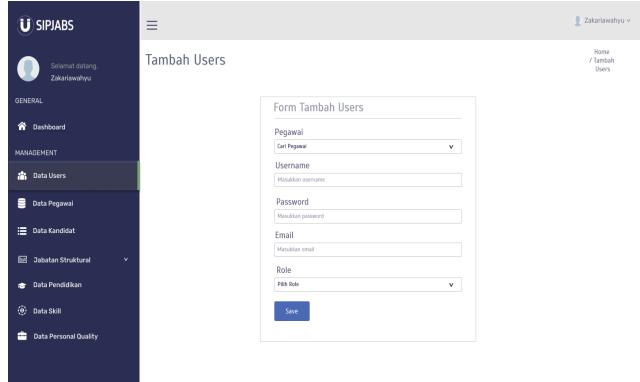
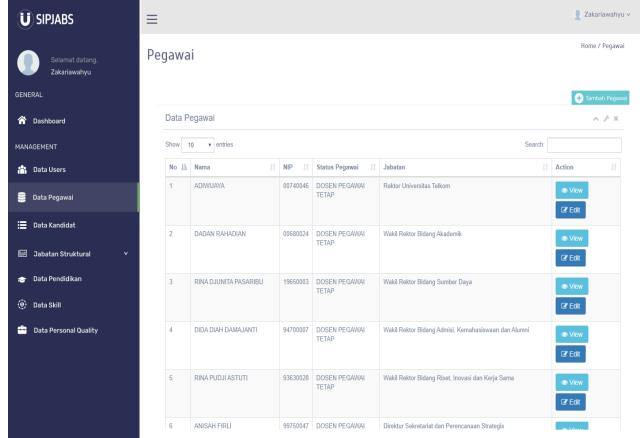
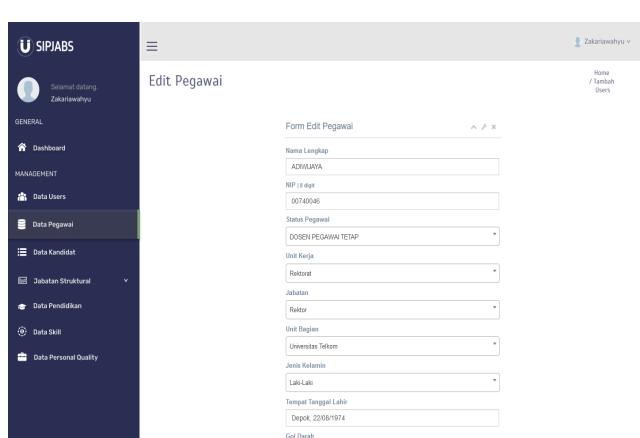
Tabel 3.41: Tabel Perancangan Antar Muka Admin (1)

No	Gambar	Keterangan
3.		Halaman <i>profile</i> admin akan menampilkan data <i>profile</i> dari admin tersebut. Kemudian admin juga dapat mengedit <i>profile</i> dan mereset password.
4.		Admin dapat mengubah <i>username</i> , menginputkan <i>email</i> , dan menambahkan foto <i>profile</i> .
5.		Admin harus menginputkan <i>password</i> yang lama serta yang baru, setelah itu admin dapat mengubah <i>password</i> .

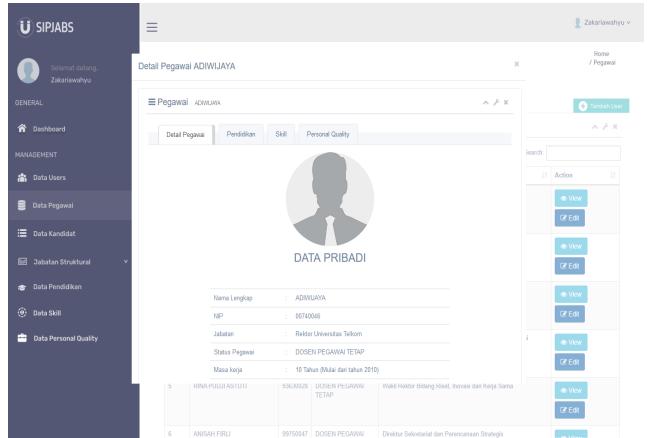
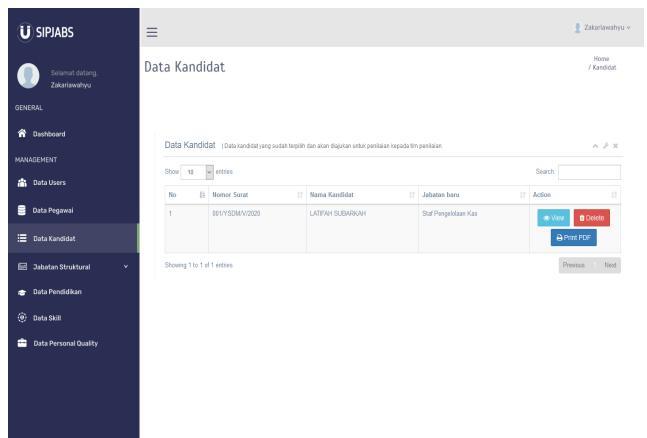
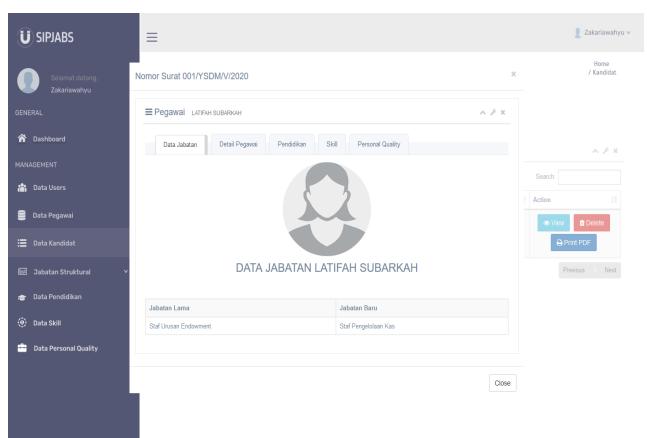
Tabel 3.42: Tabel Perancangan Antar Muka Admin (2)

No	Gambar	Keterangan
6.		Halaman <i>help</i> berisi infomasi tentang aplikasi.
7.		Halaman data <i>user</i> akan menampilkan nama-nama yang dapat mengakses aplikasi SIPJabS sebagai admin dan <i>user</i> .
8.		Pada halaman ini admin dapat mengedit <i>username</i> , <i>email</i> , dan <i>role</i> sebagai admin atau <i>user</i> .

Tabel 3.43: Tabel Perancangan Antar Muka Admin (3)

No	Gambar	Keterangan
9.		Admin dapat menambahkan <i>user</i> dengan mengisi <i>form</i> tambah <i>user</i> dan menyimpannya.
10.		Admin dapat melihat daftar data pegawai yang ada di Universitas Telkom secara detail.
11.		Admin dapat mengedit data pegawai secara lengkap.

Tabel 3.44: Tabel Perancangan Antar Muka Admin (4)

No	Gambar	Keterangan
12.		Admin dapat melihat data pribadi pegawai secara lengkap dan data detail informasi lainnya.
13.		Halaman ini akan menampilkan data kandidat yang sudah di pilih oleh user sesuai dengan <i>job description</i> untuk menggantikan atau mengisi posisi yang kosong.
14.		Admin dapat melihat data detail kandidat yang sudah dipilih.

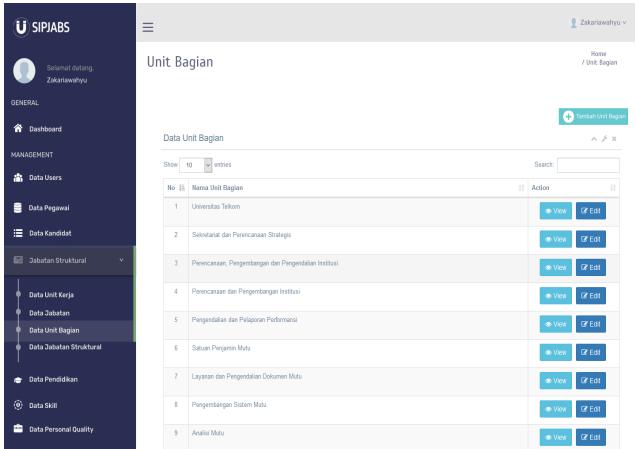
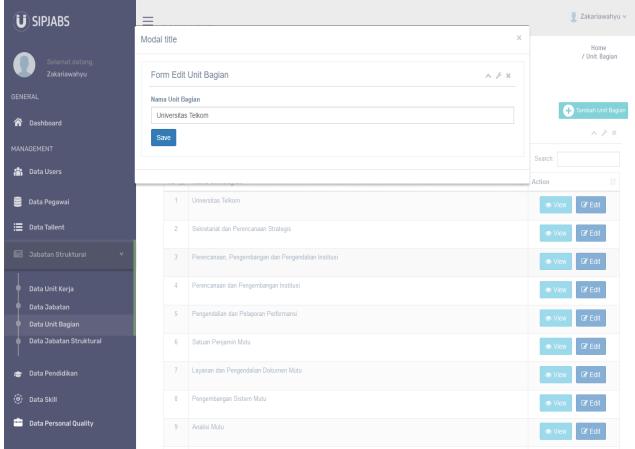
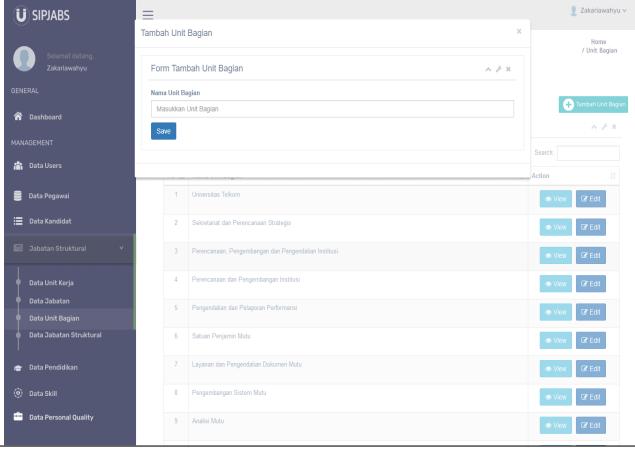
Tabel 3.45: Tabel Perancangan Antar Muka Admin (5)

No	Gambar	Keterangan
15.		Halaman ini akan menunjukkan semua unit kerja dimulai dari rektorat hingga fakultas yang ada di Universitas Telkom.
16.		Admin dapat mengedit form unit kerja apabila terdapat kebijakan baru.
17.		Admin dapat menambahkan data unit kerja dengan mengisi form tersebut, namun harus sesuai dengan kebijakan yang telah ditetapkan.

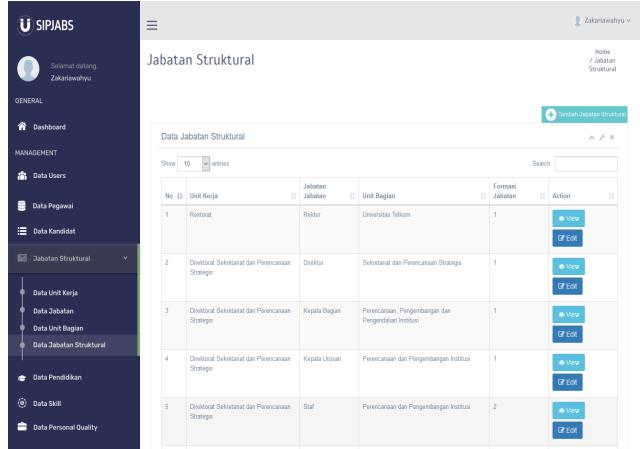
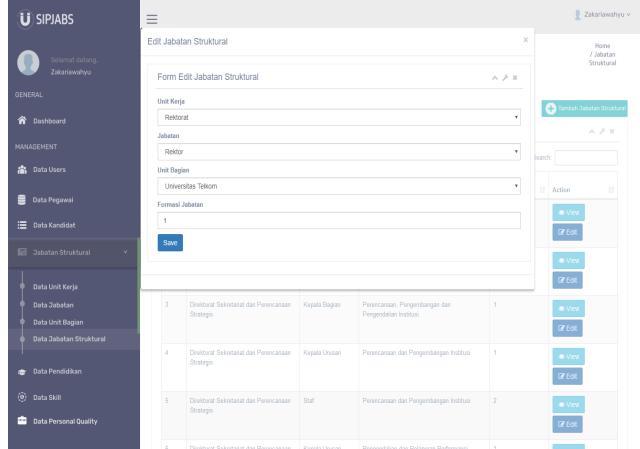
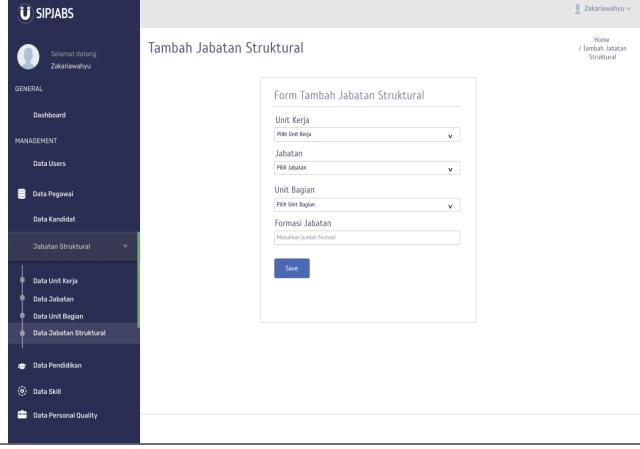
Tabel 3.46: Tabel Perancangan Antar Muka Admin (6)

No	Gambar	Keterangan
18.		Halaman ini akan menampilkan data jabatan yang berada di Universitas Telkom
19.	 Screenshot of the 'Edit Jabatan' (Edit Job Position) form. The form has fields for 'Nama Jabatan' (Name) set to 'Direktor' and 'Level Jabatan' (Level) set to '3'. A 'Save' button is present. Below the form is a list of existing job positions with columns for Name, Level, and Action (View, Edit). The data is identical to the list in row 18.	Admin dapat mengedit data jabatan sesuai dengan nama jabatan yang sudah ditetapkan.
20.	 Screenshot of the 'Tambah Jabatan' (Add Job Position) form. The form has fields for 'Nama Jabatan' (Name) with placeholder 'Masukkan jabatan' and 'Level Jabatan' (Level) with placeholder 'Pilih Level'. A 'Save' button is present. Below the form is a list of existing job positions with columns for Name, Level, and Action (View, Edit). The data is identical to the list in row 18.	Admin harus melengkapi form tersebut untuk dapat menambahkan data jabatan yang baru.

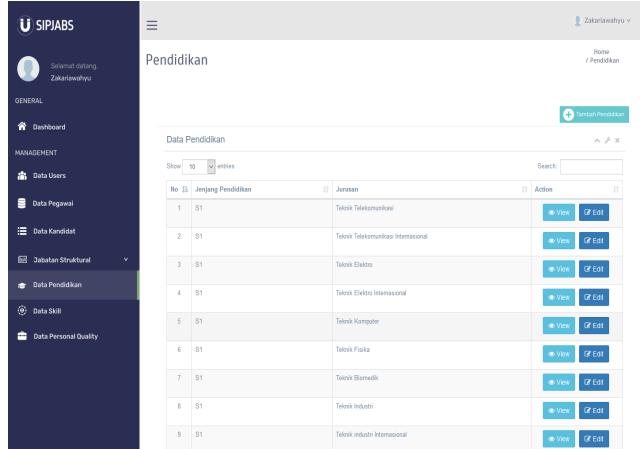
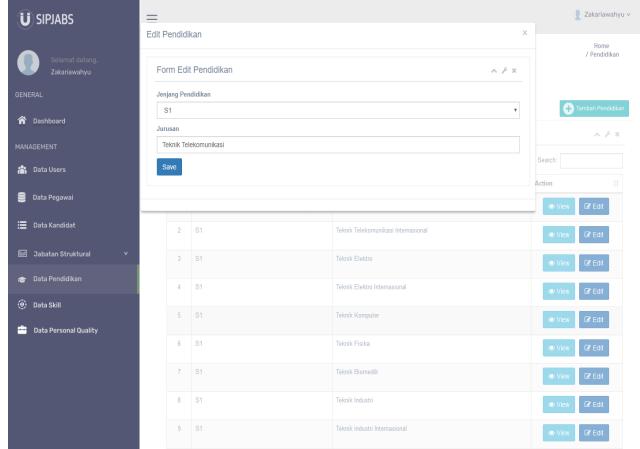
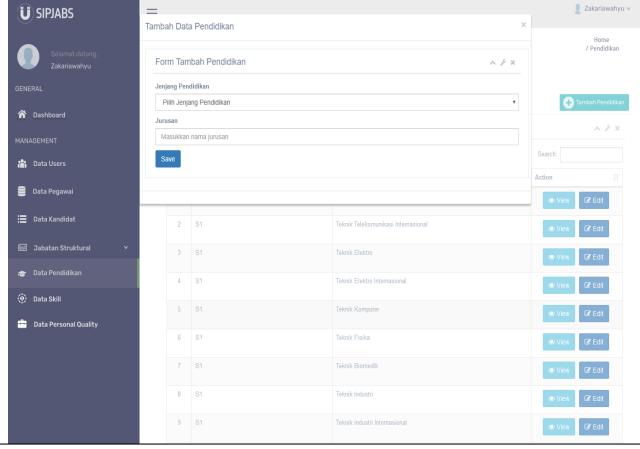
Tabel 3.47: Tabel Perancangan Antar Muka Admin (7)

No	Gambar	Keterangan
21.		Halaman ini akan menampilkan satuan kerja yang terdapat di Universitas Telkom.
22.		Admin dapat mengedit data unit bagian apabila ada perubahan yang sudah ditetapkan.
23.		Admin harus menginputkan nama unit bagian tersebut untuk dapat menambahkan data data unit bagian yang baru.

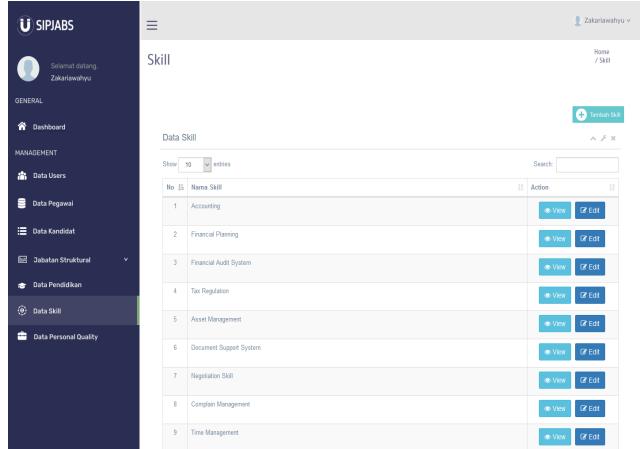
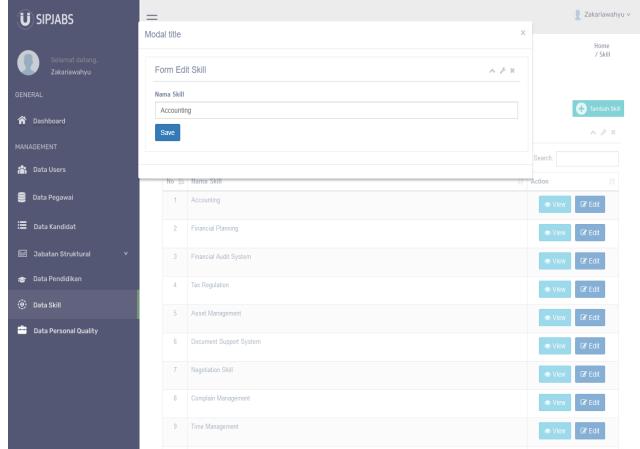
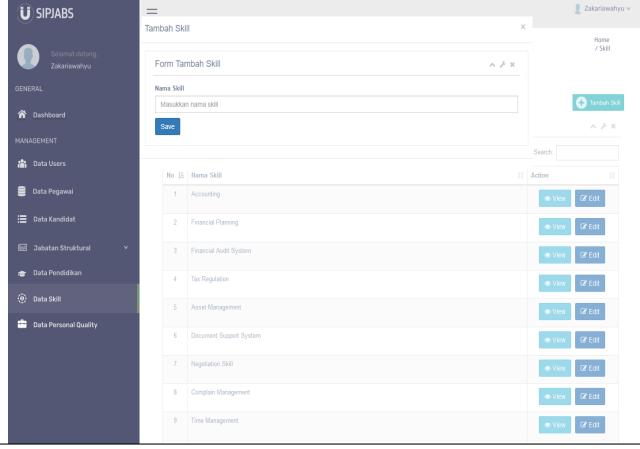
Tabel 3.48: Tabel Perancangan Antar Muka Admin (8)

No	Gambar	Keterangan
24.		Halaman ini menampilkan jabatan yang secara tegas ada di Universitas Telkom.
25.		Admin dapat mengedit data dan harus mengisi form sesuai dengan ketetapan.
26.		Admin harus melengkapi form untuk dapat menambahkan data jabatan struktural baru yang sudah ditetapkan.

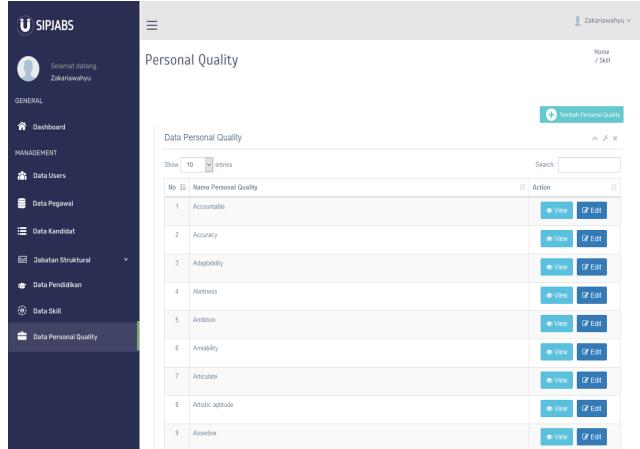
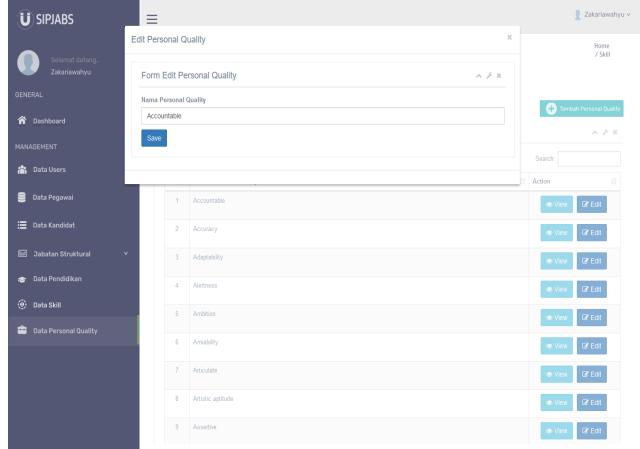
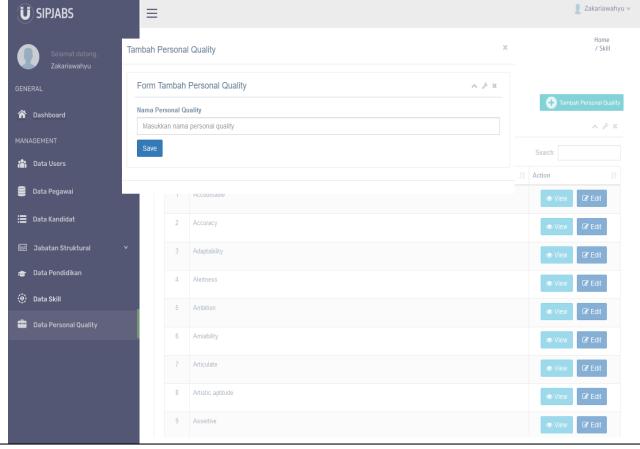
Tabel 3.49: Tabel Perancangan Antar Muka Admin (9)

No	Gambar	Keterangan
27.		Halaman ini akan menampilkan data pendidikan yang dimiliki pegawai Universitas Telkom.
28.		Apabila ingin mengedit maka admin harus meng-inputkan jenjang pendidikan serta jurusan.
29.		Admin dapat menambahkan data pendidikan apabila belum ada data pendidikan yang dimiliki pegawai belum terinput.

Tabel 3.50: Tabel Perancangan Antar Muka Admin (10)

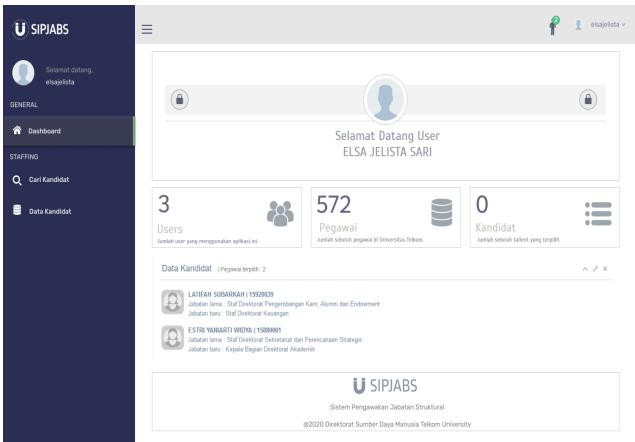
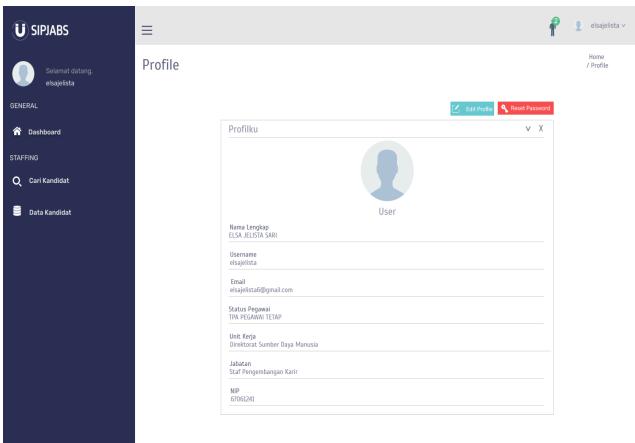
No	Gambar	Keterangan
30.	 A screenshot of a web application interface titled 'Skill'. On the left is a dark sidebar menu with 'GENERAL' and 'MANAGEMENT' sections, including 'Data Users', 'Data Pegawai', 'Data Kandidat', 'Jabatan Struktural', 'Data Pendidikan', 'Data Skill', and 'Data Personal Quality'. The main content area shows a table titled 'Data Skill' with 10 entries. Each entry has a 'View' and 'Edit' button. The entries are: 1 Accounting, 2 Financial Planning, 3 Financial Audit System, 4 Tax Regulation, 5 Asset Management, 6 Document Support System, 7 Negotiation Skill, 8 Complain Management, and 9 Time Management.	Halaman ini akan menampilkan <i>skill</i> yang dimiliki pegawai Universitas Telkom untuk menunjang pekerjaan.
31.	 A screenshot of a modal window titled 'Form Edit Skill'. It contains a single input field labeled 'Nama Skill' with the value 'Accounting' and a 'Save' button below it. In the background, the same list of skills as in the previous screenshot is visible.	Admin dapat mengedit data <i>skill</i> .
32.	 A screenshot of a modal window titled 'Tambah Skill'. It contains a single input field labeled 'Nama Skill' with the placeholder 'Masukan nama skill' and a 'Save' button below it. In the background, the same list of skills as in the previous screenshots is visible.	Admin dapat menambahkan data <i>skill</i> baru apabila terdapat data <i>skill</i> yang belum diinputkan.

Tabel 3.51: Tabel Perancangan Antar Muka Admin (11)

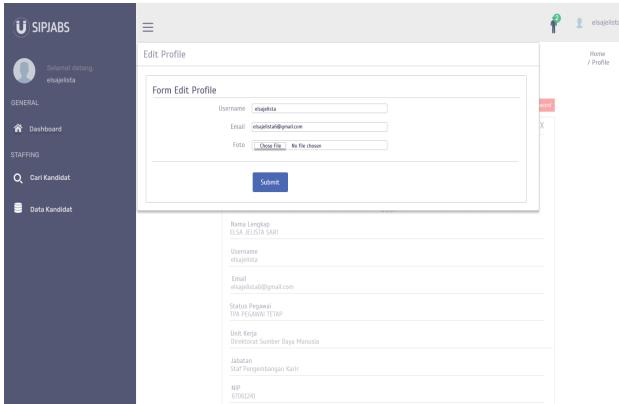
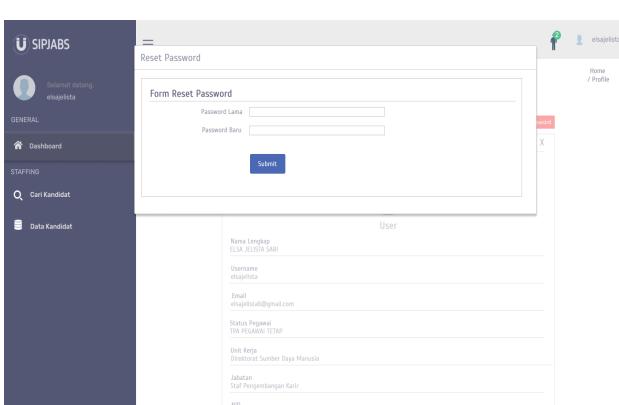
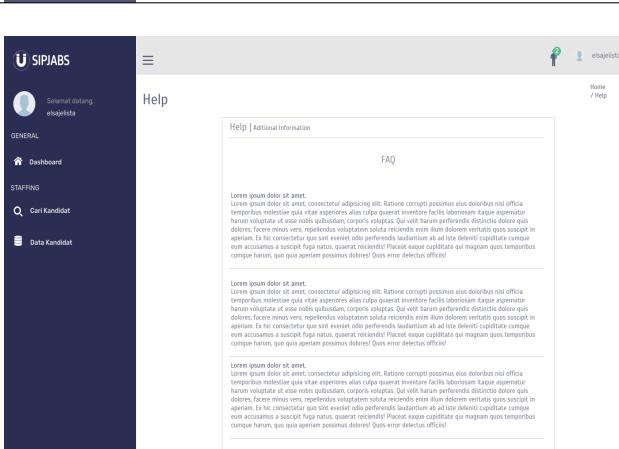
No	Gambar	Keterangan
33.		Halaman ini akan menampilkan <i>personal quality</i> yang dimiliki pegawai Universitas Telkom untuk menunjang pekerjaan.
34.		Admin dapat mengedit data <i>personal quality</i> .
36.		Admin dapat menambahkan data <i>personal quality</i> baru apabila terdapat data <i>personal quality</i> yang belum diinputkan.

3.4.1.2 Perancangan Antar Muka User

Tabel 3.52: Tabel Perancangan Antar Muka User

No	Gambar	Keterangan
1.		Halaman <i>login</i> merupakan tampilan awal apabila <i>user</i> membuka aplikasi <i>SiP JabS</i> , <i>user</i> dapat menginputkan <i>username</i> dan <i>password</i> untuk melakukan <i>login</i> .
2.		Halaman <i>dashboard</i> akan menampilkan jumlah <i>user</i> yang dapat mengakses aplikasi SiP JabS , jumlah pegawai yang ada di Universitas Telkom serta jumlah kandidat yang sudah dipilih.
3.		Halaman <i>profile user</i> akan menampilkan data nama lengkap, <i>username</i> , <i>email</i> , status pegawai, unit kerja, jabatan, serta NIP dari <i>user</i> tersebut. Kemudian <i>user</i> juga dapat mengedit <i>profile</i> dan mereset <i>password</i> .

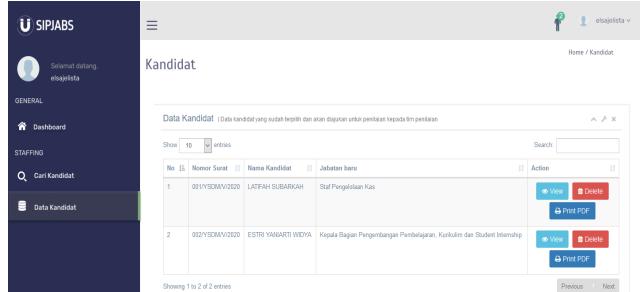
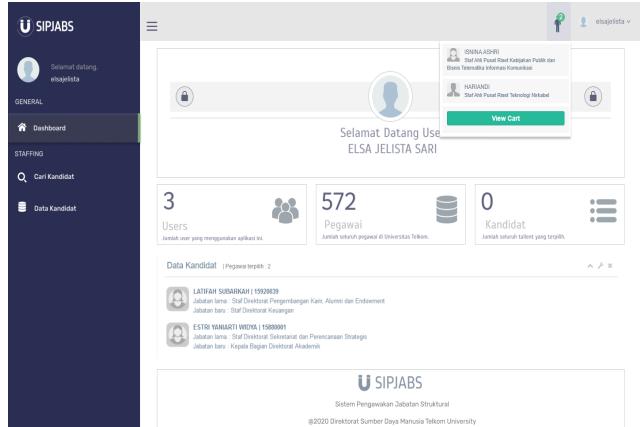
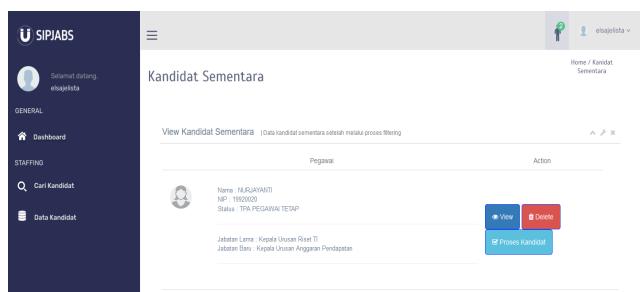
Tabel 3.53: Tabel Perancangan Antar Muka User (1)

No	Gambar	Keterangan
4.		User dapat mengubah <i>username</i> , menginputkan <i>email</i> , dan menambahkan foto <i>profile</i>
5.		User harus menginputkan <i>password</i> yang lama serta yang baru, setelah itu user dapat mereset <i>password</i> .
6.		Pada halaman <i>help</i> akan mengenai informasi dari sistem aplikasi ini

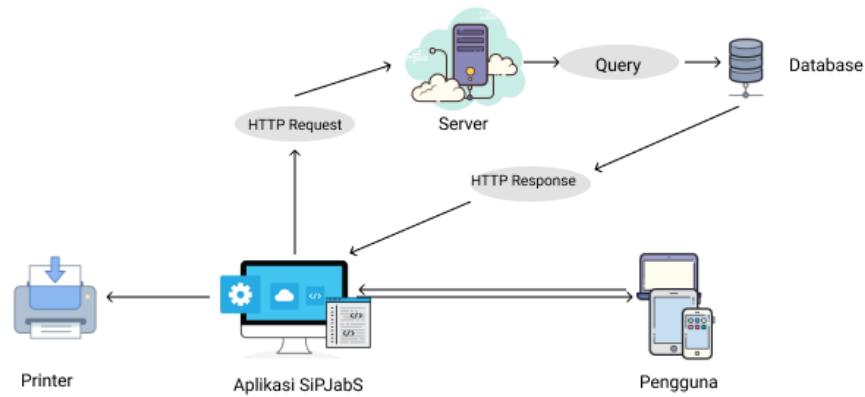
Tabel 3.54: Tabel Perancangan Antar Muka User (2)

No	Gambar	Keterangan
7.		User dapat memilih jabatan dan masa kerja yang diinginkan untuk mengantikan atau mengisi posisi yang kosong.
8.		Halaman ini akan menampilkan kandidat yang sudah dipilih dengan syarat tententu.
9.		Halaman ini akan menampilkan data pribadi dari kandidat tersebut.

Tabel 3.55: Tabel Perancangan Antar Muka User (3)

No	Gambar	Keterangan
10.		Halaman ini akan menampilkan data kandidat yang sudah diproses dari kandidat sementara tersebut.
11.		Halaman ini akan menampilkan data kandidat sementara yang sudah dipilih dan masih ada kemungkinan bisa diubah sebelum ditetapkan menjadi kandidat.
12.		Halaman ini akan menampilkan data kandidat sementara yang sudah dipilih dan akan diproses ditetapkan untuk menjadi kandidat.

3.4.2 Perancangan Level Tinggi



Gambar 3.6: High Level Design

Alur perancanaan level tinggi pada aplikasi **"SiP JabS : Sistem Pengawakan Jabatan Struktural"** dimulai pengguna melakukan perintah dah akan di salurkan melalui *HTTP Request* ke *server*. Pengambilan data akan di *filter* berdasarkan dengan *query* yang dibuat berdasarkan data yang diperoleh dari *database* yang ada. Kemudian *database* akan memberikan umpan balik berupa *HTTP Response* berdasarkan *request* data yang akan ditampilkan kepada pengguna.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Aplikasi

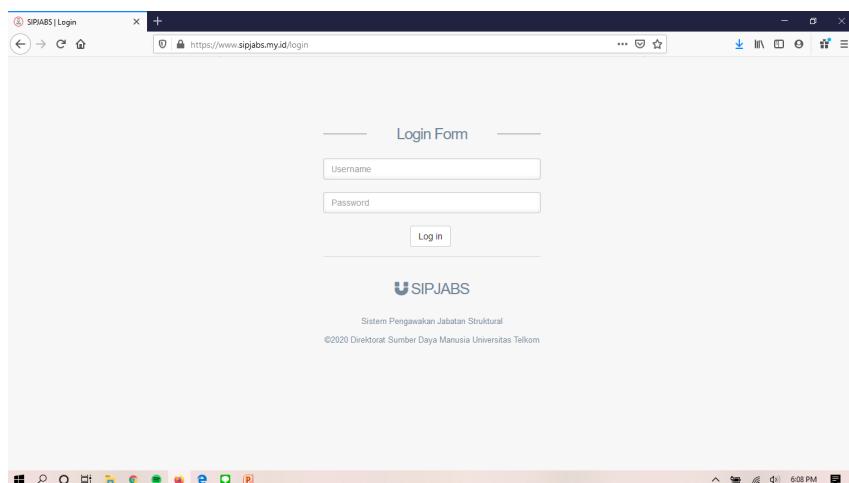
Implementasi aplikasi merupakan tahapan realisasi dari perancangan yang telah dibuat. Pada tahap ini akan dilakukan penjelasan mengenai antarmuka aplikasi “**SiPJabS: Sistem Pengawakan Jabatan Struktural di Universitas Telkom**”.

4.1.1 Implementasi Antarmuka Aplikasi

Antarmuka merupakan tampilan yang telah diimplmentasikan dari rancangan yang telah dibuat. Berikut merupakan hasil implementasi antarmuka aplikasi yang telah dibuat:

4.1.1.1 Implementasi Admin

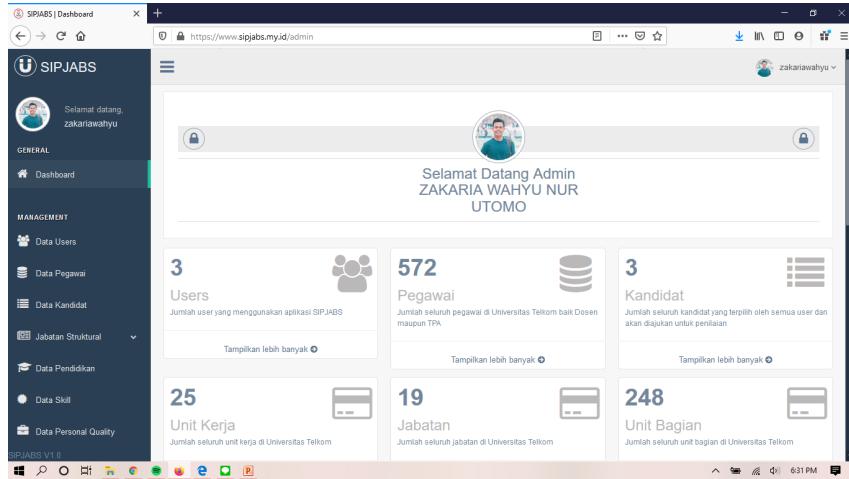
1. Halaman *Login*



Gambar 4.1: Halaman *Login* Admin

Gambar tampilan diatas merupakan implementasi dari BAB III dimana admin harus menginputkan *username* dan *password* untuk dapat melanjutkan penggunaan aplikasi, apabila sudah diinput lanjut untuk mengklik *button* “*login*”.

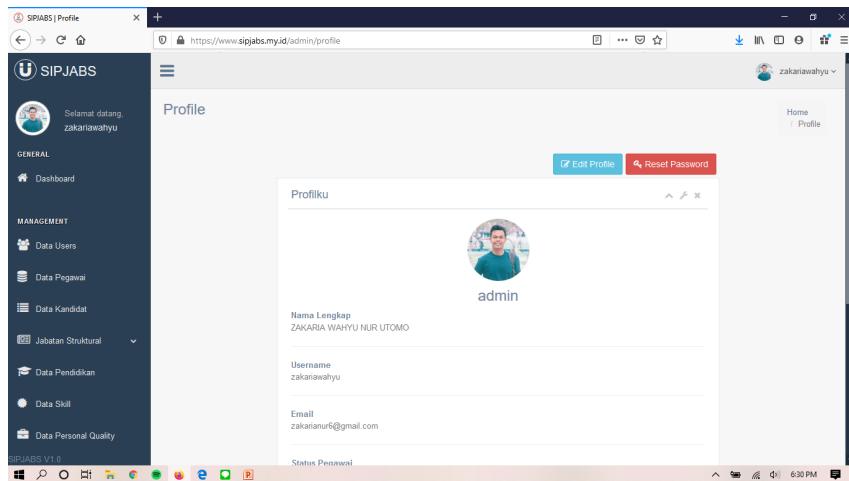
2. Halaman *Dashboard*



Gambar 4.2: Halaman *Dashboard* Admin

Gambar diatas menjelaskan isi dari *dashboard* admin, dibagi menjadi beberapa bagian diantaranya: terdapat jumlah *user* yang dapat mengakses aplikasi SiPJaBSe sebagai admin maupun *user*. Apabila admin menambahkan atau menghapus *user* maka jumlah tersebut dapat berubah. Terdapat jumlah pegawai di Universitas Telkom. Kemudian jumlah kandidat yang sudah dipilih oleh user untuk menggantikan atau mengisi posisi yang kosong. Dibawahnya terdapat unit kerja yang didalamnya terdapat beberapa jabatan, dan didalam unit kerja juga dibagi menjadi unit bagian, agar pekerjaan lebih mudah untuk dikerjakan. Kemudian terdapat pendidikan yang dimiliki oleh pegawai Universitas Telkom, dan terakhir terdapat *skill* yang dimiliki pegawai Universitas Telkom guna menunjang kerja.

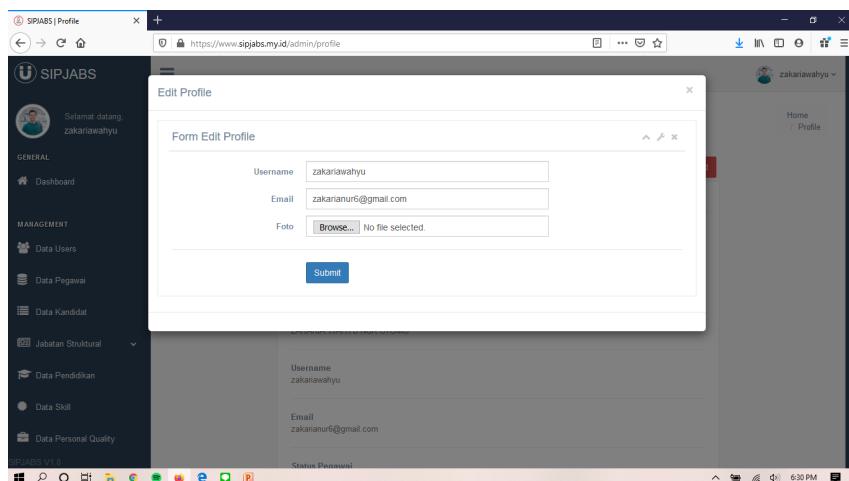
3. Halaman *Profile*



Gambar 4.3: Halaman *Profile* Admin

Implementasi diatas menjelaskan bahwasannya admin dapat melihat *profile* dengan mengklik *username* yang berada di atas kanan, kemudian akan terdapat *dropdown*, lalu klik *profile*, maka akan tampil *profile* dari admin.

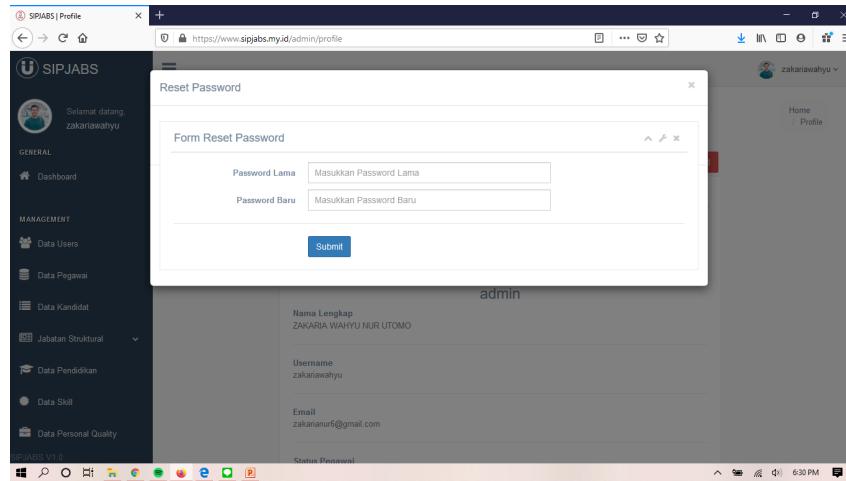
4. Halaman *Edit Profile*



Gambar 4.4: Halaman *Edit Profile* Admin

Gambar diatas mengimplementasikan bahwasannya admin dapat mengedit data *profile* dengan mengklik button “edit profile” kemudian dapat mengganti *username* atau email dan foto. Lalu admin harus mengklik “submit” untuk dapat menyimpan data yang sudah diperbarui.

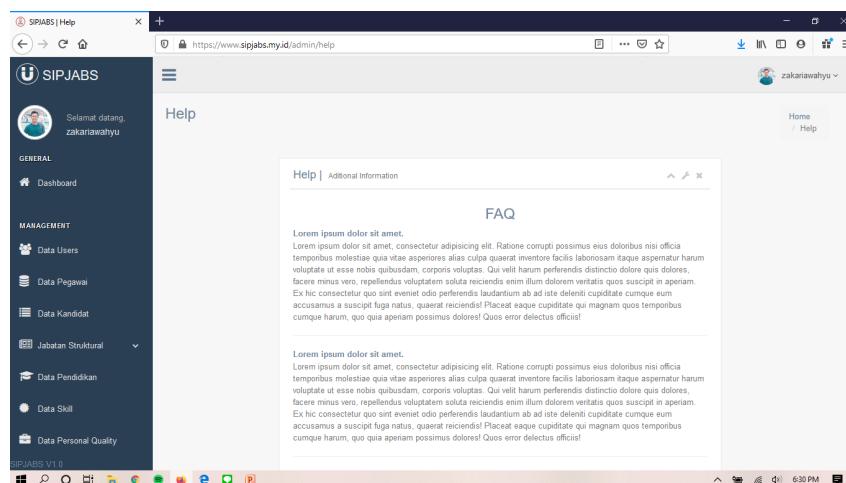
5. Halaman *Reset Password*



Gambar 4.5: Halaman *Reset Password* Admin

Implementasi diatas menandakan bahwasannya admin dapat mengganti *password* apabila *password* terebut sudah diketahui oleh pihak lain, dengan klik button “*reset password*” kemudian akan tampil halaman *pop-up* seperti diatas, admin cukup menginputkan *password* lama dan *password* baru, lalu klik “*submit*” untuk menyimpan *password* yang sudah diganti.

6. Halaman *Help*



Gambar 4.6: Halaman *Help* Admin

7. Halaman Data *Users*

No	Nama Lengkap	Jabatan	Username	Email	Hak Akses	Action
1	ZAKARIA WAHYU NUR UTOMO	Kepala Bagian Pengembangan Produk TI	zakarawahyu	zakaranur6@gmail.com	admin	View Edit Delete
2	ELSA JELISTA SARI	Staf Pengembangan Karir	elsajelista	elsajelista@gmail.com	user	View Edit Delete
3	GITA INDAH HAPSARI	Ketua Kelompok Kehilangan Commerce Management	gitahapsari	gitahapsari@gmail.com	user	View Edit Delete

Gambar 4.7: Halaman Data *Users* - Admin

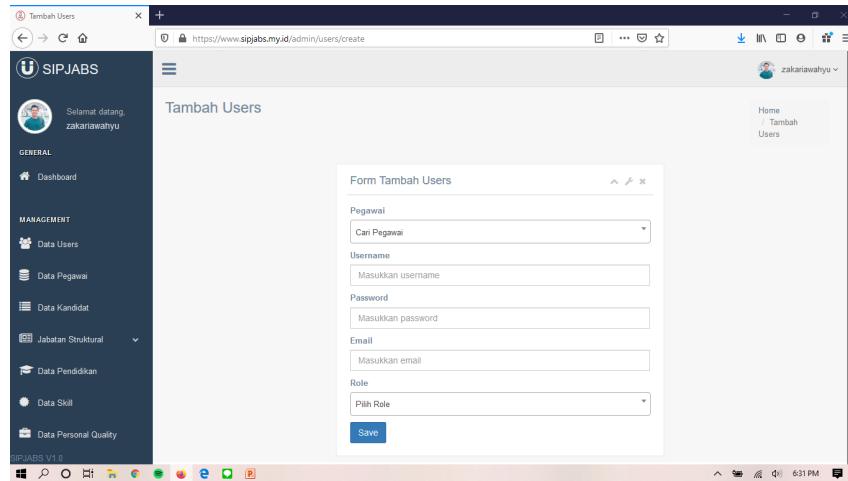
Gambar diatas menjelaskan isi dari halaman data *users*, dimana admin dapat mengelola *users* yang bisa mengakses aplikasi **SiPJabS**, karena aplikasi ini hanya dapat diakses oleh orang-orang tententu.

8. Halaman *Edit Data Users*

Gambar 4.8: Halaman *Edit Data Users* - Admin

Gambar diatas menjelaskan bahwasannya data *users* dapat diedit seperti *username*, email, dan *role* yang digunakan sebagai admin atau *user*. Apabila data sudah diubah admin dapat mengklik button “Submit” untuk menyelesaikan proses.

9. Halaman Tambah *Users*



Gambar 4.9: Halaman Tambah *Users* - Admin

Dari tampilan diatas dapat dijelaskan, hanya admin saja yang dapat menambahkan data *user*, namun harus sesuai dengan perintah dan *job description* pegawai. Dengan mencari nama pegawai, menginputkan *username*, *password*, email, dan *rolenya* sebagai admin atau *user*. Untuk mengakhiri proses admin dapat menyimpan apabila sudah disimpan akan terdapat *pop-up*.

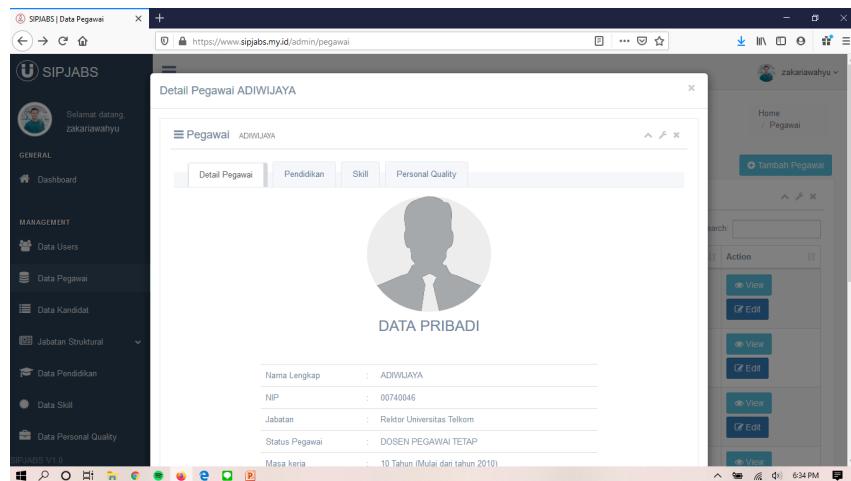
10. Halaman Data Pegawai

No	Nama	NIP	Status Pegawai	Jabatan	Action
1	ADIWUJAYA	00740046	DOSEN PEGAWAI TETAP	Rector Universitas Telkom	View Edit
2	DADAN RAHADIAN	00680024	DOSEN PEGAWAI TETAP	Wakil Rektor Bidang Akademik	View Edit
3	RINA DJUNITA PASARIBU	19650003	DOSEN PEGAWAI TETAP	Wakil Rektor Bidang Sumber Daya	View Edit
4	DIDA DIAH DAMAJANTI	94700007	DOSEN PEGAWAI	Wakil Rektor Bidang Admisi, Kemahasiswaan dan Alumni	View

Gambar 4.10: Halaman Data Pegawai - Admin

Dari tampilan diatas dijelaskan bahwasannya admin dapat melihat data seluruh pegawai yang ada di Universitas Telkom dengan mengklik menu “Data Pegawai” yang ada di sebelah kiri.

11. Halaman View Detail Pegawai



Gambar 4.11: Halaman View Detail Pegawai - Admin

Untuk implementasi *view detail* pegawai admin dapat mengklik *button view*, maka akan tampil seperti gambar diatas. Terdapat data pribadi dari pegawai yang dapat dilihat diantaranya: nama lengkap, NIP, jabatan, status pegawai, masa kerja, tempat lahir, tanggal lahir, status perkawinan, golongan darah, jenis kelamin, agama, tinggi badan, berat badan, dan alamat.

12. Halaman Edit Pegawai

Nama Lengkap	: ADIWIJAYA
NIP	: 00740046
Status Pegawai	: DOSEN PEGAWAI TETAP
Unit Kerja	: Rektor
Jabatan	: Rektor
Unit Bagian	: Universitas Telkom

Gambar 4.12: Halaman Edit Pegawai - Admin

Untuk implementasi *edit* pegawai admin dapat mengklik *button edit*, maka akan tampil seperti gambar diatas dan dapat mengedit data yang ada.

13. Halaman Data Kandidat

Gambar 4.13: Halaman Data Kandidat - Admin

Gambar diatas menampilkan halaman data kandidat yang dapat dilihat apabila admin mengklik menu “Data Kandidat”. Terdapat data kandidat yang sudah dipilih oleh *user*, sesuai dengan *requirement* yang telah ditetapkan perusahaan guna mengganti atau mengisi posisi yang kosong.

14. Halaman View Detail Kandidat

Jabatan Lama	Jabatan Baru
Staf Usulan Endowment	Staf Pengelolaan Kas

Gambar 4.14: Halaman View Detail Kandidat - Admin

Tampilan diatas akan muncul apabila admin mengklik button “view”. Kemudian tampil *detail* data pegawai yang suda dipilih oleh *user*, seperti data pribadi, jenjang pendidikan, dan *skill*.

15. Halaman Data Unit Kerja

No	Nama Unit Kerja	Action
1	Rektorat	View Edit
2	Direktorat Sekretariat dan Perencanaan Strategis	View Edit
3	Direktorat Akademik	View Edit
4	Direktorat Pascasarjana & Advance Learning	View Edit
5	Direktorat Logistik dan Aset	View Edit
6	Direktorat Keuangan	View Edit

Gambar 4.15: Halaman Data Unit Kerja - Admin

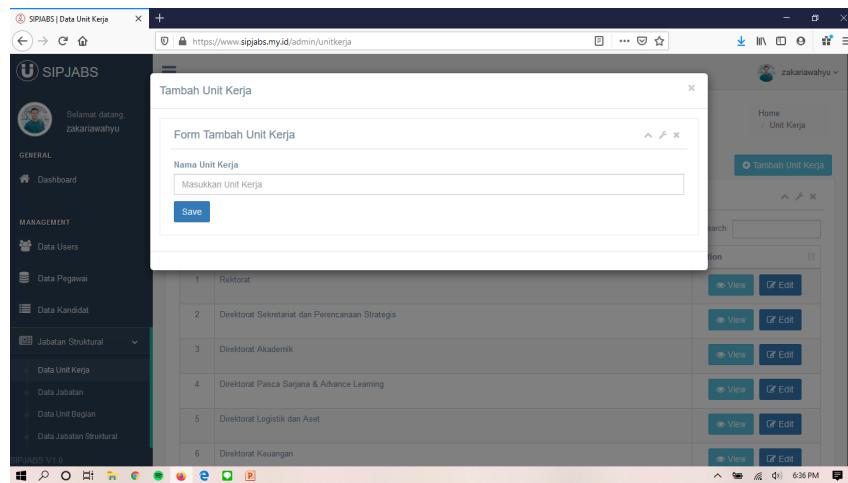
Untuk implementasi diatas admin harus klik “jabatan struktural” kemudian akan terdapat *dropdown*, lalu klik “data unit kerja” maka akan menampilkan data unit kerja yang berada di Universitas Telkom dari rektorat hingga ke fakultas.

16. Halaman *Edit* Unit Kerja

Gambar 4.16: Halaman *Edit* Unit Kerja - Admin

Gambar diatas akan muncul apabila admin mengklik *button “edit”* kemudian akan tampil *pop-up* edit unit kerja, apabila nama unit kerja mengalami perubahan yang sudah ditetapkan.

17. Halaman Tambah Unit Kerja



Gambar 4.17: Halaman Tambah Unit Kerja - Admin

Implementasi diatas akan muncul apabila admin mengklik button “tambah unit kerja”, apabila terdapat unit kerja baru yang sudah ditetapkan, maka admin dapat menambahkannya. Untuk menyelesaikan proses admin harus mengklik button “save”, kemudian data akan tersimpan.

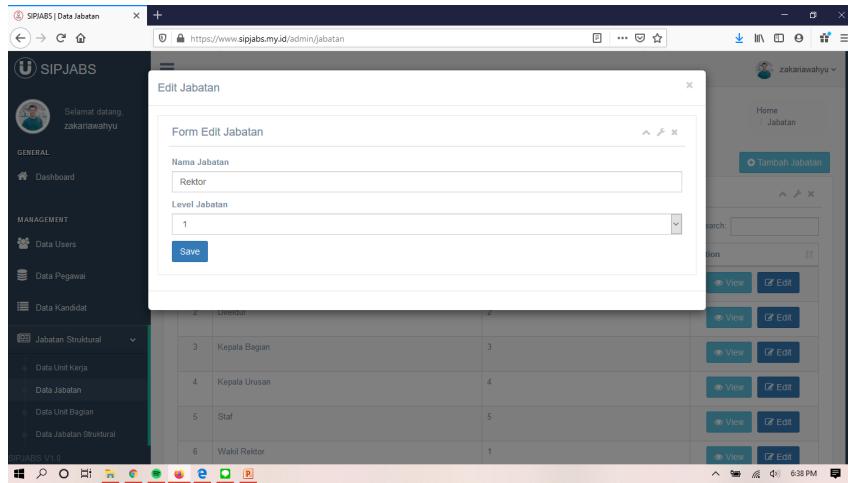
18. Halaman Data Jabatan

No	Nama Jabatan	Level Jabatan	Action
1	Rektor	1	[View] [Edit]
2	Direktur	2	[View] [Edit]
3	Kepala Bagian	3	[View] [Edit]
4	Kepala Unusan	4	[View] [Edit]
5	Staf	5	[View] [Edit]
6	Wakil Rektor	1	[View] [Edit]

Gambar 4.18: Halaman Data Jabatan - Admin

Gambar diatas menampilkan data jabatan apabila admin mengklik menu “data jabatan” pada dropdown ”jabatan struktural”. Terdapat nama-nama jabatan yang ada di Universitas Telkom yang akan terhubung dengan unit kerja dan unit bagian.

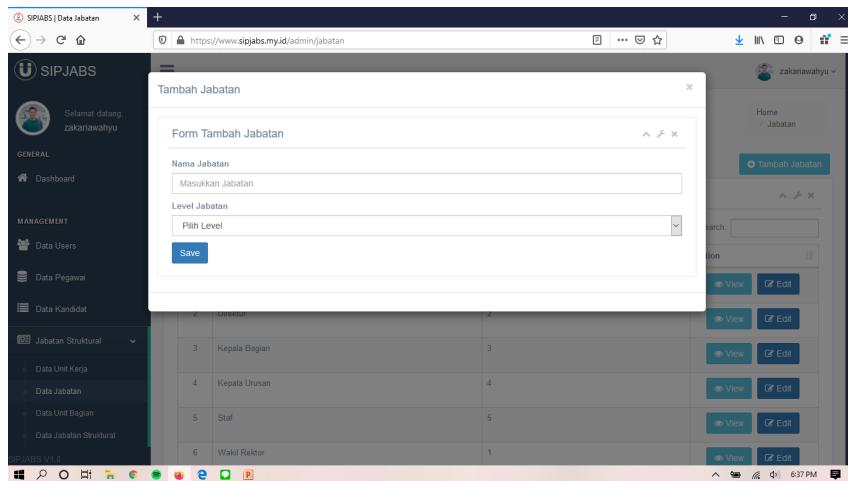
19. Halaman *Edit* Jabatan



Gambar 4.19: Halaman *Edit* Jabatan - Admin

Untuk tampilan diatas adalah *form edit* jabatan, admin dapat mengklik *button “edit”* kemudian menginputkan nama jabatan serta level jabatan, dan menyimpannya untuk mengakhiri proses, kemudian data akan tersimpan.

20. Halaman Tambah Jabatan



Gambar 4.20: Halaman Tambah Jabatan - Admin

Gambar diatas menampilkan *pop-up form* tambah jabatan, tampilan tersebut akan muncul apabila admin mengklik *button “tambah jabatan”*. Kemudian admin menginputkan nama jabatan dan level jabatan untuk membuat jabatan baru yang sudah ditetapkan. Lalu simpan data tersebut dengan mengklik *button “save”* maka data akan tersimpan di jabatan.

21. Halaman Data Unit Bagian

The screenshot shows a web-based administrative interface for managing university departments. The left sidebar has a dark theme with white text and icons. It includes sections for GENERAL (Dashboard), MANAGEMENT (Data Users, Data Pegawai, Data Kandidat), and JABATAN STRUKTURAL (Data Unit Kerja, Data Jabatan, Data Unit Bagian, Data Jabatan Struktural). The main content area is titled "Unit Bagian" and displays a table of "Data Unit Bagian". The table has columns for "No", "Nama Unit Bagian", and "Action". The data listed is as follows:

No	Nama Unit Bagian	Action
1	Universitas Telkom	View Edit
2	Sekretariat dan Perencanaan Strategis	View Edit
3	Perencanaan, Pengembangan dan Pengendalian Institusi	View Edit
4	Perencanaan dan Pengembangan Institusi	View Edit
5	Pengendalian dan Pelaporan Performansi	View Edit
6	Satuan Penjamin Mutu	View Edit

Gambar 4.21: Halaman Data Unit Bagian - Admin

Untuk tampilan diatas adalah tampilan dari data unit bagian, didapat apabila admin mengklik *dropdown* “jabatan struktural” pada menu kemudian mengklik “data unit bagian”, halaman ini menjelaskan bagian dari unit kerja yang ada di Universitas Telkom. Diharapkan dapat menyelesaikan pekerjaan sesuai dengan unit bagian masing-masing secara efektif dan maksimal.

22. Halaman Edit Unit Bagian

The screenshot shows the "Edit Unit Bagian" form overlaid on the main "Data Unit Bagian" table. The form has a title "Form Edit Unit Bagian" and a single input field labeled "Nama Unit Bagian" containing the value "Universitas Telkom". Below the input field is a blue "Save" button. The background table remains the same as in the previous screenshot, listing six department entries.

Gambar 4.22: Halaman Edit Unit Bagian - Admin

Gambar diatas menampilkan *form edit* unit bagian, didapat apabila admin mengklik *button* “edit” kemudian admin dapat menginputkan nama unit bagian yang baru dan sesuai dengan ketetapan, terakhir klik *button* “save” sehingga data akan tersimpan kembali.

23. Halaman Tambah Unit Bagian

The screenshot shows a web browser window for the SIPJABS application. The main content is a modal dialog titled "Tambah Unit Bagian" (Add Department). It contains a single input field labeled "Nama Unit Bagian" (Department Name) with the placeholder "Masukkan Unit Bagian". Below the input field is a blue "Save" button. In the background, there is a table listing six existing departments, each with a "View" and "Edit" button. The departments listed are: 1. Universitas Telkom, 2. Sekretariat dan Perencanaan Strategis, 3. Perencanaan, Pengembangan dan Pengendalian Institusi, 4. Perencanaan dan Pengembangan Institusi, 5. Pengembangan dan Pelaporan Performansi, and 6. Satuan Penjamin Mutu.

Gambar 4.23: Halaman Tambah Unit Bagian - Admin

Implementasi diatas menunjukkan bahwasannya admin dapat menambah data unit bagian dengan cara klik *button* “tambah unit bagian” dan menginputkan nama unit bagian dan menyimpannya dengan mengklik *button* “save”, kemudian data akan tersimpan didalam data unit bagian.

24. Halaman Data Jabatan Struktural

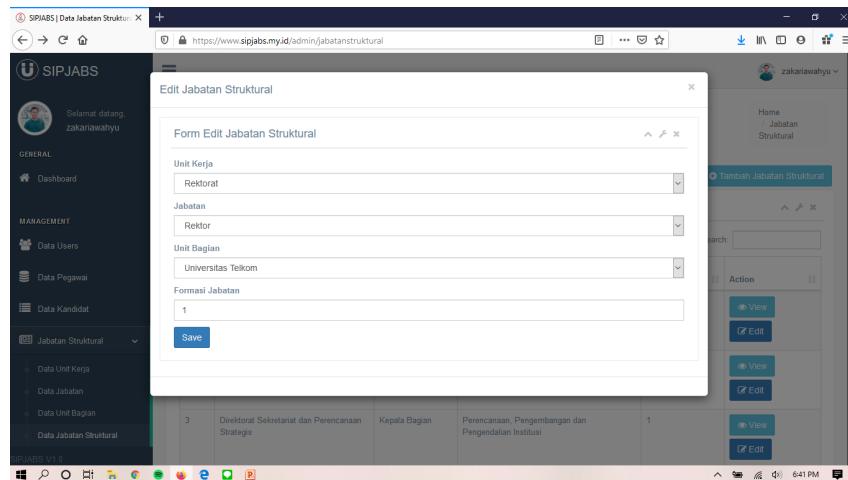
The screenshot shows a web browser window for the SIPJABS application. The main content is a table titled "Data Jabatan Struktural". The table has columns: No, Unit Kerja, Jabatan, Unit Bagian, Formasi Jabatan, and Action. There are three rows of data:

No	Unit Kerja	Jabatan	Unit Bagian	Formasi Jabatan	Action
1	Rекторat	Rector	Universitas Telkom	1	View Edit
2	Direktorat Sekretariat dan Perencanaan Strategis	Direktur	Sekretariat dan Perencanaan Strategis	1	View Edit
3	Direktorat Sekretariat dan Perencanaan Strategis	Kepala Bagian	Perencanaan, Pengembangan dan Pengendalian Institusi	1	View Edit

Gambar 4.24: Halaman Data Jabatan Struktural - Admin

Untuk implementasi dari tampilan diatas admin dapat mengklik *dropdown* “jabatan struktural” kemudian mengklik menu “data jabatan struktural”, maka akan tampil halaman jabatan struktural yang ada di Universitas Telkom. Terdiri dari unit kerja, jabatan, unit bagian serta formasi jabatan.

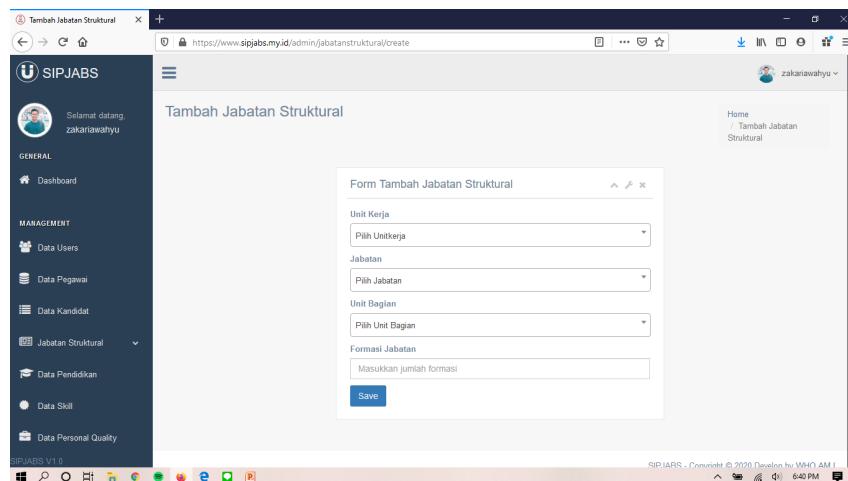
25. Halaman *Edit* Jabatan Struktural



Gambar 4.25: Halaman *Edit* Jabatan Struktural - Admin

Gambar diatas mengimplementasikan *pop-up* dari *form edit* jabatan struktural, dimana admin harus menginputkan unit kerja, jabatan, unit bagian, dan formasi jabatan, lalu untuk mengakhiri proses admin harus mengklik *button* “*save*” kemudian data yang diedit akan tersimpan.

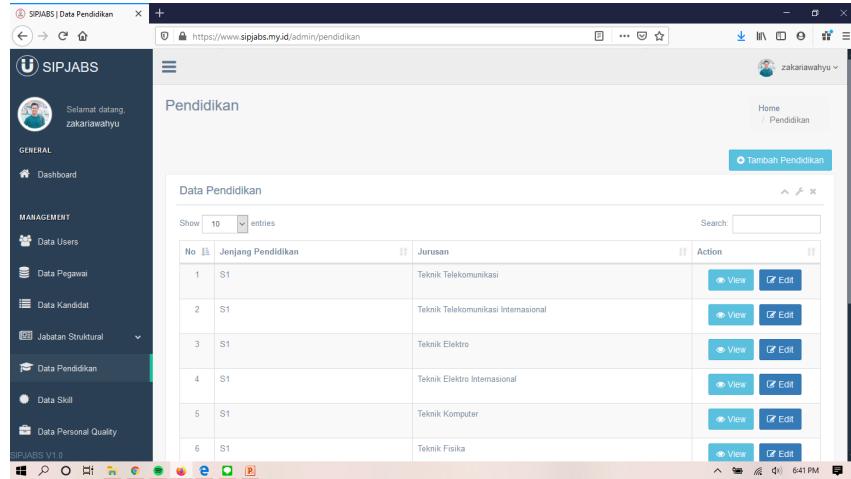
26. Halaman Tambah Jabatan Struktural



Gambar 4.26: Halaman Tambah Jabatan Struktural - Admin

Untuk tampilan diatas adalah *form tambah* jabatan stuktural, didapat apabila admin mengklik *button* “tambah jabatan struktural”, kemudian admin harus menginputkan data seperti unit kerja, jabatan, unit bagian, dan formasi jabatan, kemudian klik *button* “*save*” untuk mengakhiri proses.

27. Halaman Data Pendidikan



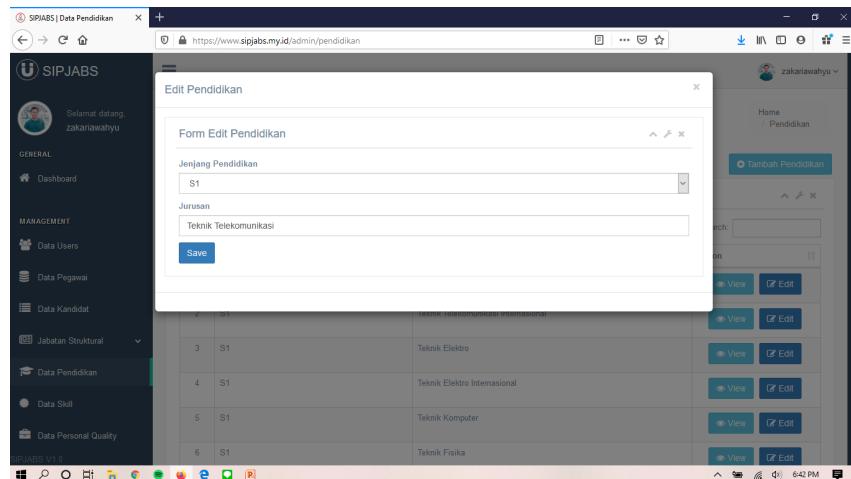
The screenshot shows a web-based administrative interface for managing employee data. The left sidebar contains a navigation menu with categories like GENERAL (Dashboard), MANAGEMENT (Data Users, Data Pegawai, Data Kandidat, Jabatan Struktural, Data Pendidikan, Data Skill, Data Personal Quality), and UPJABS V1.0. The main content area is titled "Pendidikan" and displays a table titled "Data Pendidikan". The table has columns for "No", "Jenjang Pendidikan", "Jurusan", and "Action". There are six entries listed:

No	Jenjang Pendidikan	Jurusan	Action
1	S1	Teknik Telekomunikasi	View Edit
2	S1	Teknik Telekomunikasi Internasional	View Edit
3	S1	Teknik Elektro	View Edit
4	S1	Teknik Elektro Internasional	View Edit
5	S1	Teknik Komputer	View Edit
6	S1	Teknik Fisika	View Edit

Gambar 4.27: Halaman Data Pendidikan - Admin

Gambar diatas menjelaskan data pendidikan semua pegawai Universitas Tekom, sehingga admin dapat mengetahui bahwasannya pegawai di Universitas Telkom memiliki jenjang pendidikan dan jurusan apa saja. Didapat apabila admin megklik menu “data pendidikan”.

28. Halaman *Edit* Pendidikan

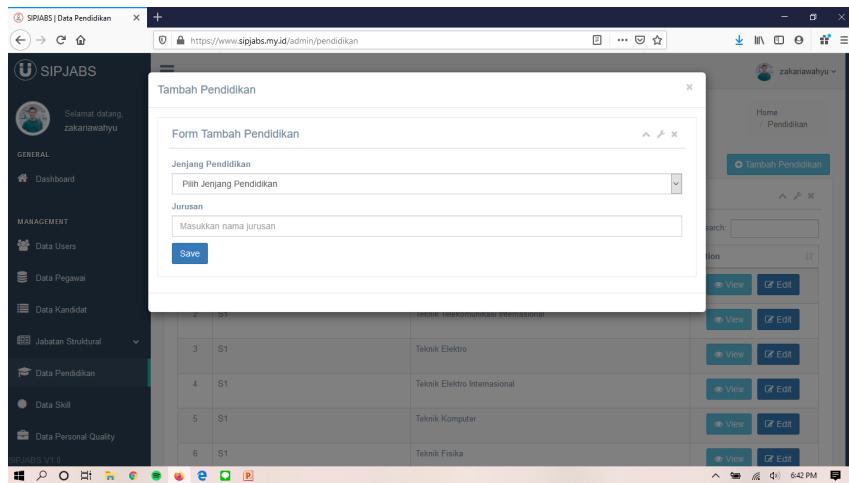


The screenshot shows a modal dialog box titled "Edit Pendidikan" overlying the main "Data Pendidikan" table. The dialog box contains two input fields: "Jenjang Pendidikan" (with a dropdown menu showing "S1") and "Jurusan" (with a dropdown menu showing "Teknik Telekomunikasi"). Below these fields is a blue "Save" button. The background table remains visible, showing the same six rows of data as in Figure 4.27.

Gambar 4.28: Halaman *Edit* Pendidikan - Admin

Untuk tampilan diatas adalah tampilan *pop-up* dari *form edit* pendidikan, apabila admin ingin mengedit data pendidikan dengan mengklik button “edit” kemudian memilih jenjang pendidikan dan menginputkan jurusan, apabila sudah sesuai klik button “save” untuk menyimpan data.

29. Halaman Tambah Pendidikan



Gambar 4.29: Halaman Tambah Pendidikan - Admin

Gambar diatas menampilkan *pop-up* tambah data pendidikan, apabila terdapat pegawai baru, namun data pendidikan belum terdaftar maka admin dapat menambahkannya dengan memilih jenjang pendidikan dan menginput jurusan, kemudian diakhiri dengan mengklik button “*save*” dan data akan tersimpan.

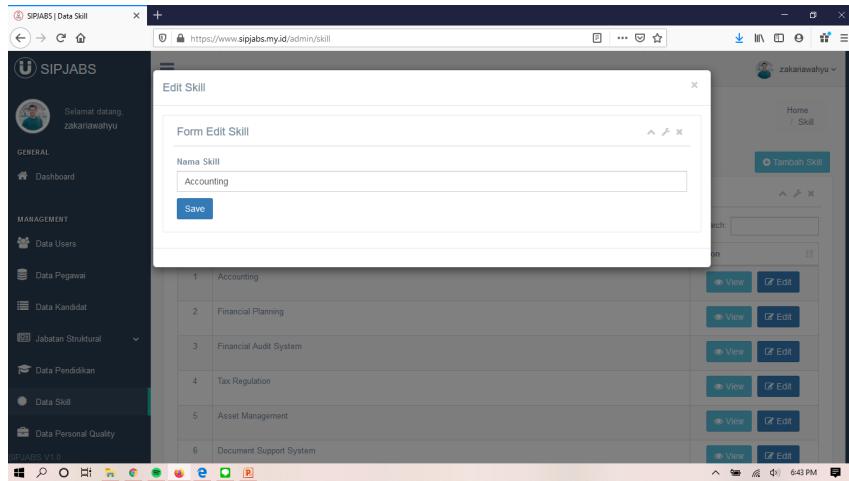
30. Halaman Data Skill

Data Skill		Action
Show	10 entries	Search:
No	Nama Skill	
1	Accounting	View Edit
2	Financial Planning	View Edit
3	Financial Audit System	View Edit
4	Tax Regulation	View Edit
5	Asset Management	View Edit
6	Document Support System	View Edit

Gambar 4.30: Halaman Data Skill - Admin

Gambar diatas merupakan implementasi data *skill* yang ada di Universitas Telkom, data tersebut ada karena setiap pegawai memiliki *skill* kemudian data *skill* dari semua pegawai disatukan didalam data *skill*, untuk menunjang pekerjaan.

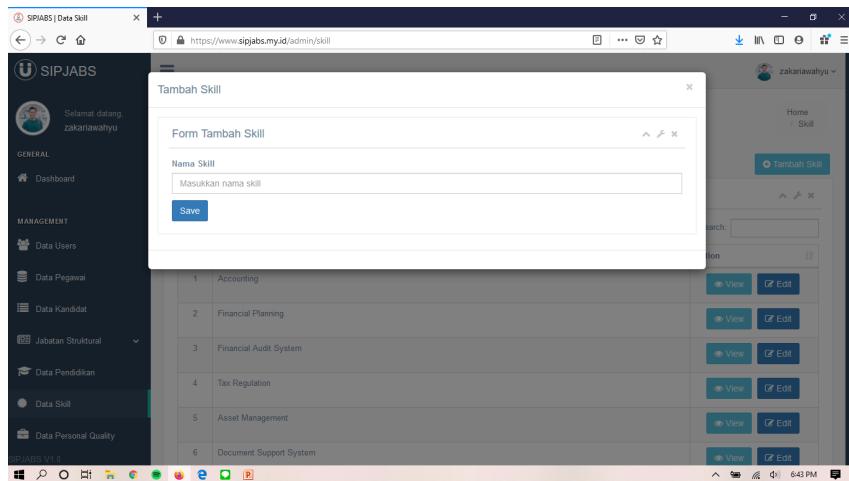
31. Halaman *Edit Skill*



Gambar 4.31: Halaman *Edit Skill* - Admin

Implementasi diatas menampilkan *pop-up form edit skill*, dimana admin dapat mengedit nama *skill* menjadi lebih sesuai, dan menyimpannya kembali dengan mengklik button “*save*”.

32. Halaman Tambah Skill



Gambar 4.32: Halaman Tambah Skill - Admin

Untuk tampilan diatas adalah tambah data *skill*, admin dapat menambahkan data *skill* dengan mengkil button “tambah *skill*” apabila *skill* pada pegawai belum terdaftar pada data *skill*, lalu inputkan nama *skill* dan menyimpannya dengan klik button “*save*”.

33. Halaman Data *Personal Quality*

No	Nama Personal Quality	Action
1	Accountable	View Edit
2	Accuracy	View Edit
3	Adaptability	View Edit
4	Alertness	View Edit
5	Ambition	View Edit

Gambar 4.33: Halaman Data *Personal Quality* - Admin

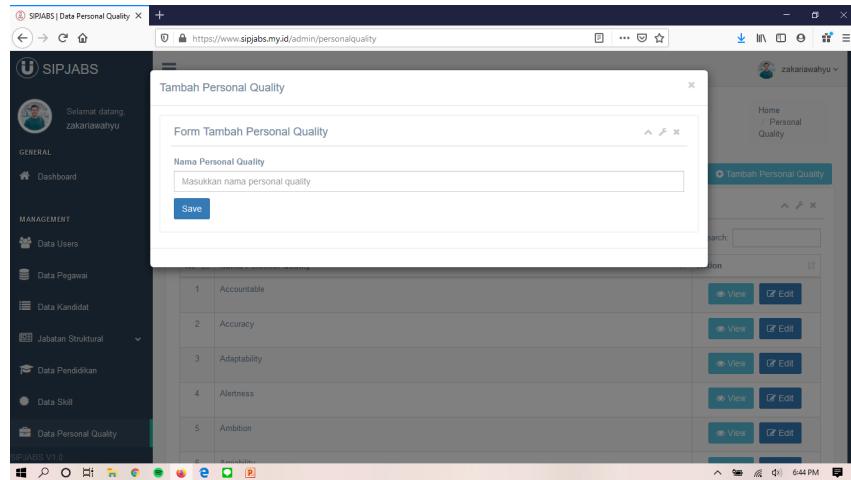
Gambar diatas merupakan implementasi data *personal quality* yang ada di Universitas Telkom, data tersebut ada karena setiap pegawai memiliki *personal quality* kemudian data *skill* dari semua pegawai disatukan didalam data *personal quality*, untuk menunjang pekerjaan.

34. Halaman *Edit Personal Quality*

Gambar 4.34: Halaman *Edit Personal Wuality* - Admin

Implementasi diatas menampilkan *pop-up form edit personal quality*, dimana admin dapat mengedit nama *personal quality* menjadi lebih sesuai, dan menyimpannya kembali dengan mengklik button “save”.

35. Halaman Tambah Personal Quality

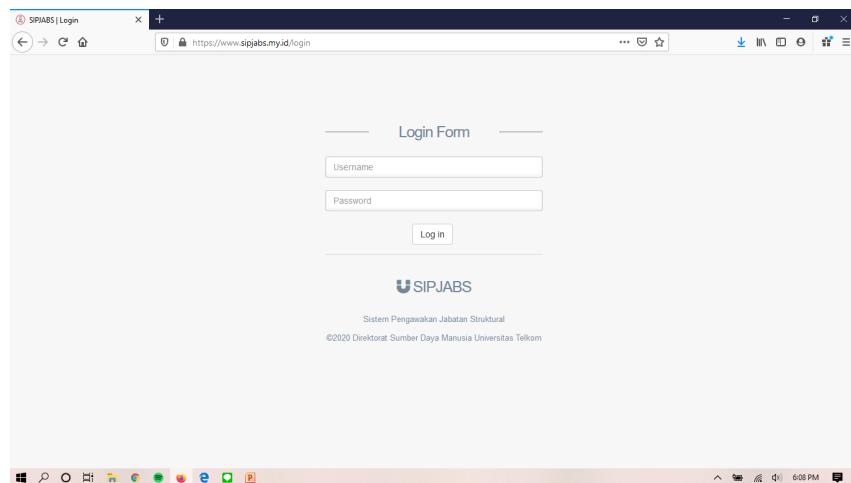


Gambar 4.35: Halaman Tambah Personal Quality - Admin

Untuk tampilan diatas adalah tambah data *personal quality*, admin dapat menambahkan data *personal quality* dengan mengkil button “tambah *personal quality*”.

4.1.1.2 Implementasi User

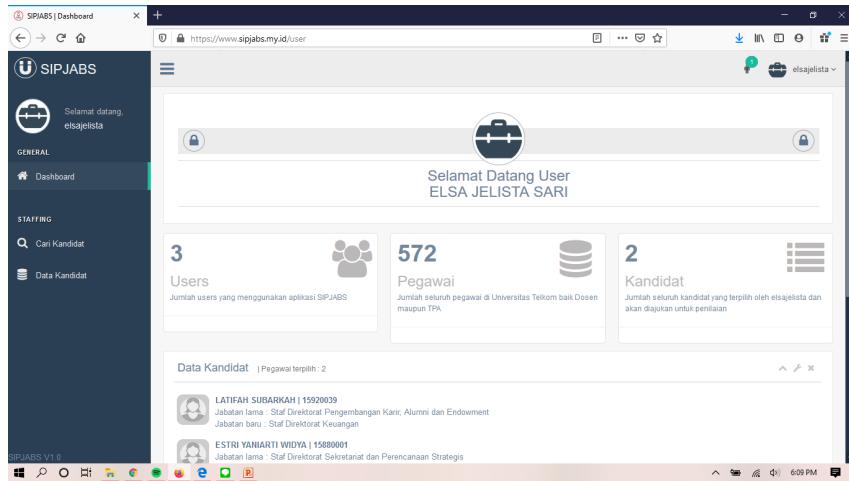
1. Halaman Login



Gambar 4.36: Halaman Login User

Tampilan diatas merupakan implementasi dari BAB III dimana *user* harus menginputkan *username* dan *password* untuk dapat melanjutkan penggunaan aplikasi, apabila sudah diinputkan lanjut untuk mengklik button “log in”.

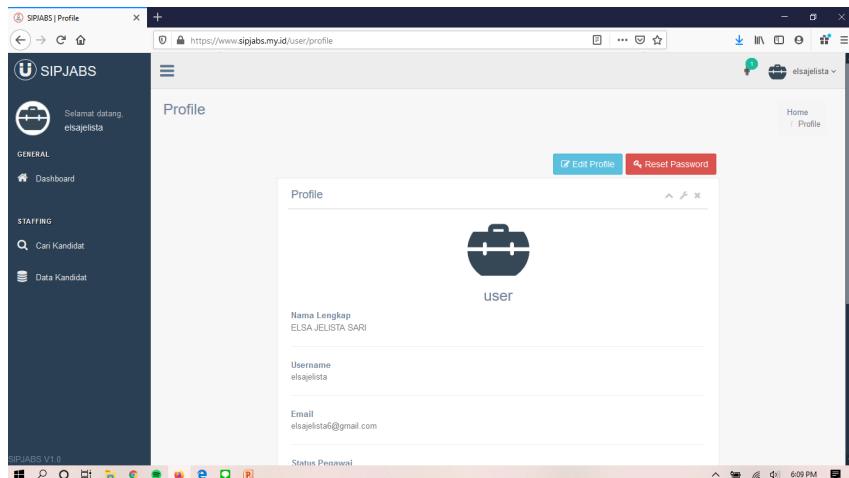
2. Halaman *Dashboard*



Gambar 4.37: Halaman *Dashboard User*

Gambar diatas mengimplementasikan isi dari *dashboard user*, berbeda dengan *dashboard* yang dimiliki admin, *dashboard user* hanya menampilkan jumlah *user* yang dapat mengakses aplikasi **SiP JabS**, jumlah pegawai yang terdaftar di Universitas Telkom, serta jumlah kandidat yang sudah dipilih.

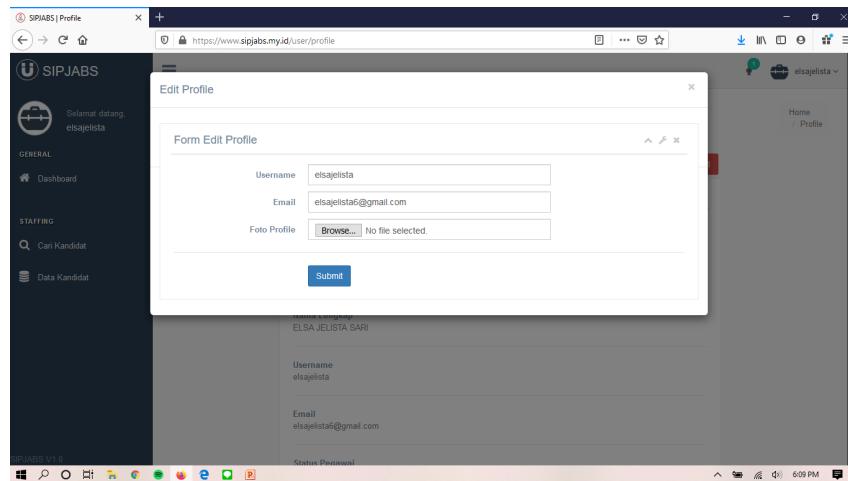
3. Halaman *Profile*



Gambar 4.38: Halaman *Profile User*

Implementasi diatas menunjukkan isi *profile* dari *user*, terdapat nama lengkap, *username*, email, status pegawai, unit kerja, jabatan, dan NIP. Admin dapat membuka halaman tersebut dengan cara klik *username* yang ada di atas kanan, kemudian akan terdapat *dropdown* dan klik *profile*, maka halaman akan tampil seperti diatas.

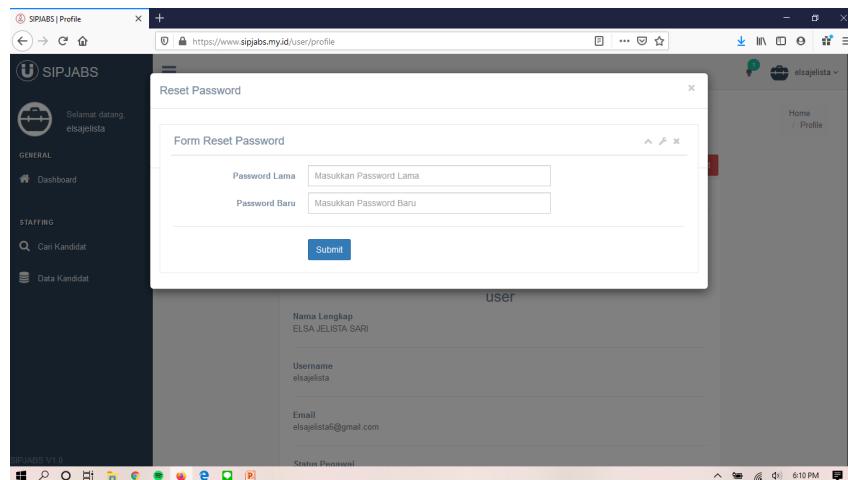
4. Halaman *Edit Profile*



Gambar 4.39: Halaman *Edit Profile User*

Gambar diatas mengimplementasikan *edit profile user* dengan *pop-up*, *user* dapat mengganti *username*, email atau foto. Kemudian klik “*submit*” untuk menyimpan data yang sudah diganti agar tersimpan.

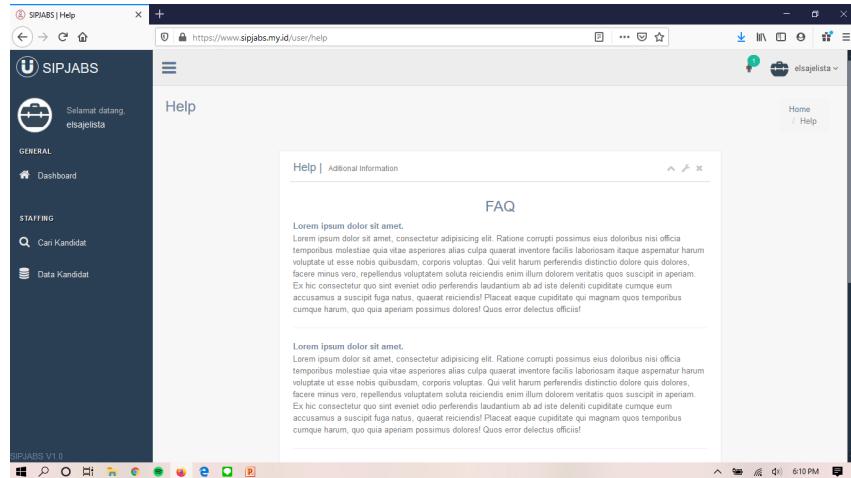
5. Halaman *Reset Password*



Gambar 4.40: Halaman *Reset Password User*

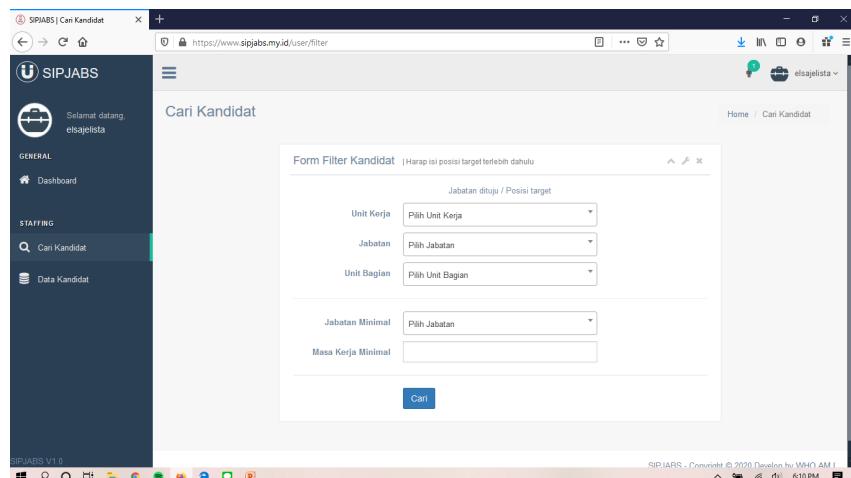
Implementasi diatas menjelaskan bahwasannya *user* dapat mengganti *password*, apabila *password* tersebut sudah diketahui oleh pihak lain, *user* dapat mengklik button “*reset password*” kemudian akan tampil *pop-up* seperti diatas, *user* harus menginputkan *password* lama dan *password* yang baru, lalu klik “*submit*” untuk menyimpan *password* yang baru.

6. Halaman Help



Gambar 4.41: Halaman Help User

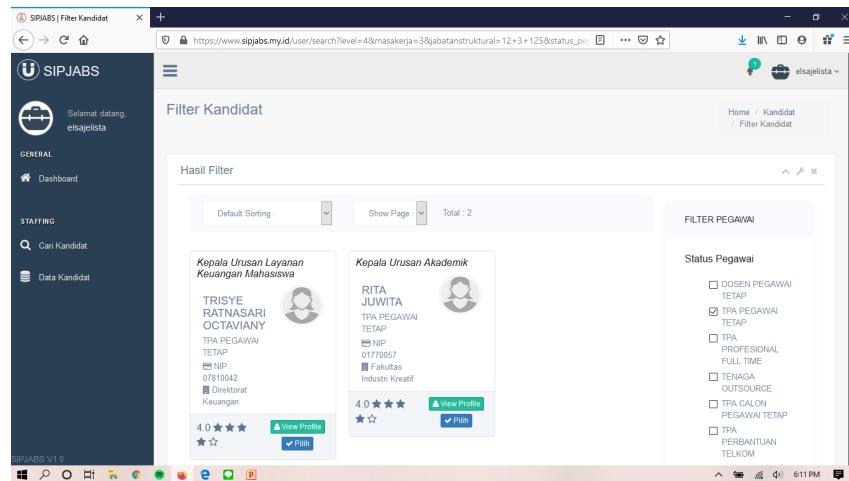
7. Halaman Cari Kandidat



Gambar 4.42: Halaman Cari Kandidat - User

Gambar diatas mengimplementasikan apabila *user* ingin mencari kandidat yang dibutuhkan sesuai *requirement* perusahaan untuk menggantikan atau mengisi posisi yang kosong, dengan cara klik “cari kandidat” pada menu, lalu akan muncul halaman seperti diatas, user harus memilih jabatan yang dituju, jabatan minimal yang dibutuhkan dan menginputkan masa kerja. Dan klik button “cari”.

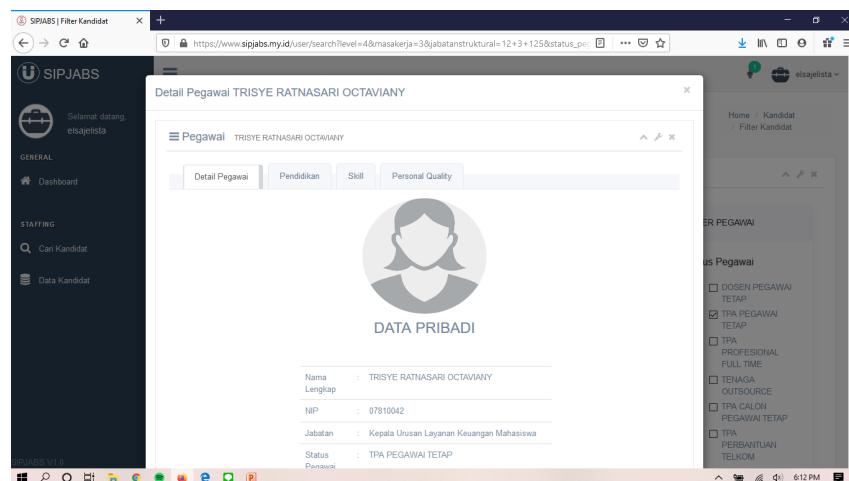
8. Halaman *Filtering*



Gambar 4.43: Halaman *Filtering* - *User*

Implementasi diatas menampilkan data kandidat yang tersedia, kemudian *user* dapat menyeleksi kembali dengan memilih *filter* pegawai, jenjang pendidikan, jurusan dan *skill* yang dimiliki oleh pegawai.

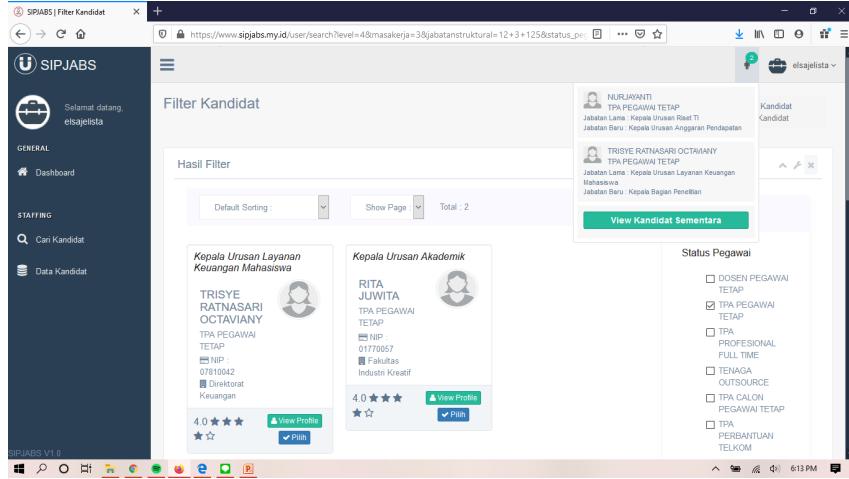
9. Halaman *View Detail Pegawai*



Gambar 4.44: Halaman *View Detail Pegawai* - *User*

Gambar diatas mengimplementasikan bahwasannya *user* dapat melihat data pribadi yang dimiliki kandidat, dengan cara klik button “view” maka akan tampil halaman seperti diatas dan menampilkan data pribadi pegawai beserta data riwayat pendidikan, skill dan personal quality.

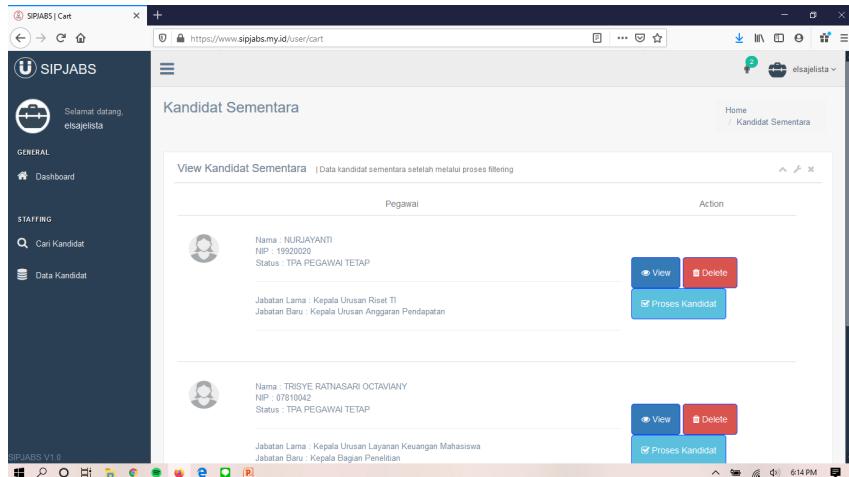
10. Halaman Kandidat Sementara



Gambar 4.45: Halaman *Cart - User*

Gambar diatas menjelaskan bahwasannya *user* sudah memilih kandidat yang sesuai dengan yang dibutuhkan untuk mengganti atau mengisi posisi yang kosong.

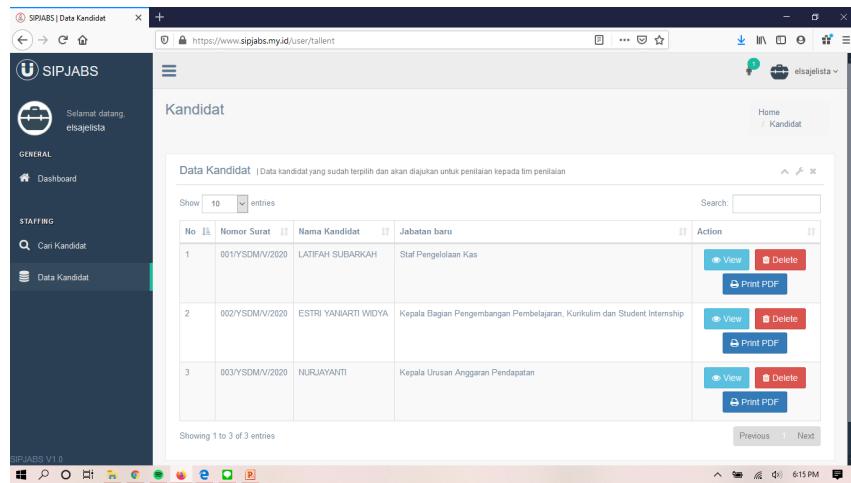
11. Halaman *View Kandidat Sementara*



Gambar 4.46: Halaman *View Kandidat Sementara*

Gambar diatas mengimplementasikan halaman *view* pada kandidat sementara, apabila *user* telah memilih kandidat maka data tersebut akan masuk kedalam kandidat sementara, sebelum diproses ke data kandidat.

12. Halaman Data Kandidat



Gambar 4.47: Halaman Data Kandidat

Gambar diatas mengimplementasikan isi dari data kandidat yang dimiliki *user*, data tersebut sudah dipilih sesuai dengan *requirement* perusahaan dan sesuai dengan *job descripton* yang dimiliki pegawai dan selanjutnya akan diajukan untuk penilaian.

4.1.2 Struktur Kode

Struktur kode merupakan kumpulan kode atau script yang berfungsi untuk menjalankan setiap perintah dalam setiap proses pembuatan aplikasi. Berikut struktur kode yang ada pada aplikasi SiPJabS:

Tabel 4.1: Tabel Struktur Kode

No	Struktur Kode	Controller	Method	Deskripsi
1	User	Cart Controller	index() show() destroy() addCart()	Berfungsi untuk melakukan proses menampilkan, menambah dan menghapus data kandidat sementara
2	User	Filter Controller	index() show() filterTallent() getJabatan() fetch()	Berfungsi untuk melakukan proses filtering pegawai untuk mencari kandidat berdasarkan requirement

Tabel 4.2: Tabel Struktur Kode (1)

No	Struktur Kode	Controller	Method	Deskripsi
3	User	Profile Controller	index() show() edit() update() resetPass()	Berfungsi untuk melakukan proses melihat profile akun, reset password dan edit profile pada user
4	User	Tallent Controller	index() show() destroy() cetak pdf() addTallent()	Berfungsi untuk melakukan pengelolaan seperti menambahkan data kandidat terpilih, menghapus dan print pdf
5	User	User Controller	index()	Berfungsi menampilkan halaman dashboard user
6	Admin	Admin Controller	index()	Berfungsi menampilkan halaman dashboard admin
7	Admin	Jabatan Controller	index() create() store() show() edit() update()	Berfungsi untuk melakukan proses CRUD pada jabatan
8	Admin	Jabatan Struktural Controller	index() create() store() show() edit() update()	Berfungsi untuk melakukan proses CRUD pada jabatan struktural
9	Admin	Pegawai Controller	index() create() store() show() edit() update()	Berfungsi untuk melakukan proses CRUD pada pegawai

Tabel 4.3: Tabel Struktur Kode (2)

No	Struktur Kode	Controller	Method	Deskripsi
10	Admin	Pendidikan Controller	index() create() store() show() edit() update()	Berfungsi untuk melakukan proses CRUD pada pendidikan
11	Admin	Personal Quality Controller	index() create() store() show() edit() update()	Berfungsi untuk melakukan proses CRUD pada personal quality
12	Admin	Profile Controller	index() show() edit() update() resetPass()	Berfungsi untuk melakukan proses melihat profile akun, reset password dan edit profile pada admin
13	Admin	Tallent Controller	index() show() destroy() cetak pdf()	Berfungsi untuk melakukan proses menampilkan, delete dan cetak pdf kandidat

4.2 Pengujian Aplikasi

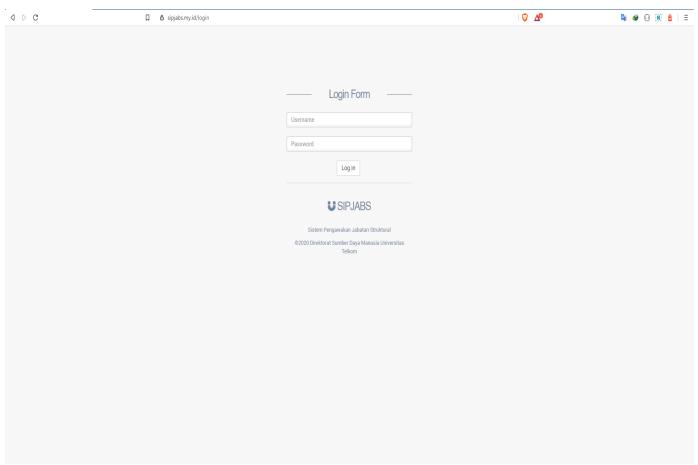
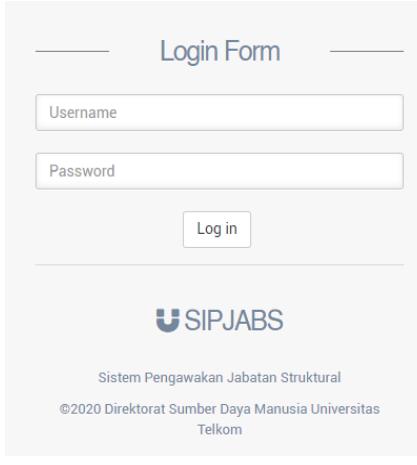
Pengujian aplikasi merupakan proses yang bertujuan untuk dapat menilai kesesuaian serta dapat menentukan kualitas dari aplikasi/sistem yang telah dibuat dengan cara menemukan bug dari aplikasi/sistem. Metode testing yang akan dilakukan adalah Pengujian Alpha dan Pengujian Beta.

4.2.1 Pengujian Alpha

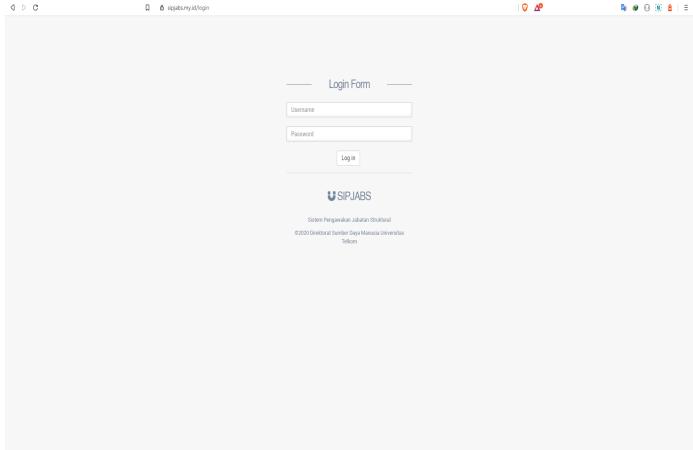
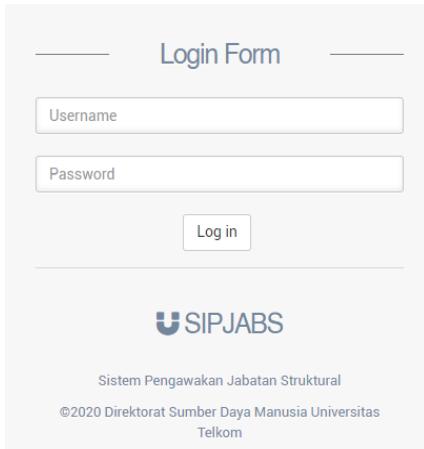
Pengujian Aplha dilakukan sebelum aplikasi di distribusikan, salah satunya dilakukan pengujian dengan cara BlackBox. Pada pengujian blackbox untuk mengetahui fungsionalitas dan kesesuaian sistem berdasarkan rancangan yang telah dibuat.

4.2.1.1 Pengujian Fungsionalitas

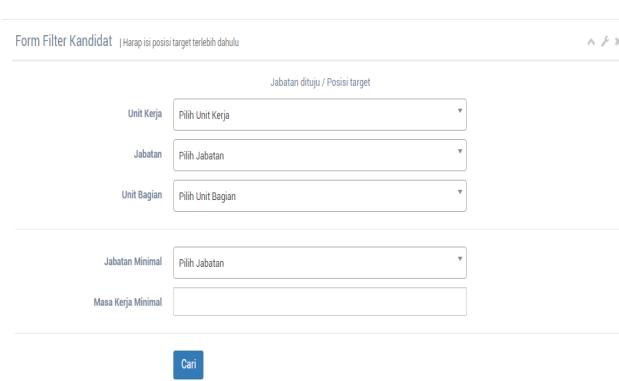
Tabel 4.4: Tabel Pengujian Login Admin

Nomor Test	PF-O1
Pengguna	Admin
Judul	Login Admin
Teknik	<p>1. Membuka website dengan mengunjungi link www.sipjabs.my.id dan akan muncul halaman login</p>  <p>2. Inputkan username : zakariawahyu dan password : admin pada form login berikut.</p>  <p>3. Klik button 'Login' untuk menyelesaikan proses login</p>
Kriteria Keberhasilan	Sistem akan menampilkan halaman dashboard admin
Kondisi khusus	Username dan password sudah diberikan oleh admin
Hasil	Valid : Yes Invalid : No

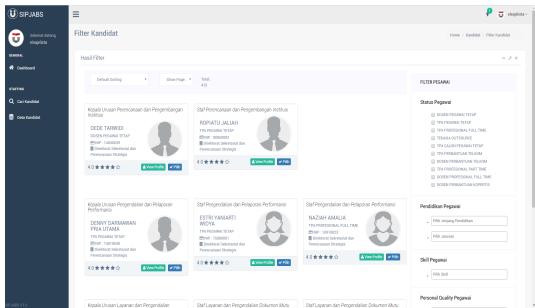
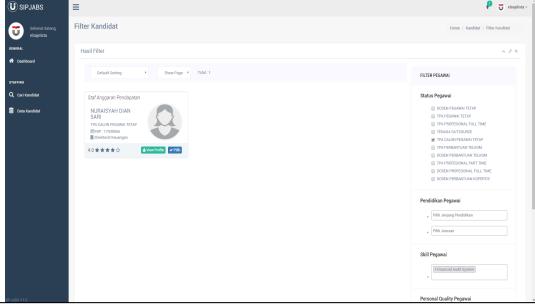
Tabel 4.5: Tabel Pengujian Login User

Nomor Test	PF-O2
Pengguna	User
Judul	Login User
Teknik	<p>1. Membuka website dengan mengunjungi link www.sipjabs.my.id dan akan muncul halaman login</p>  <p>2. Inputkan username : elsajelista dan password : user pada form login berikut.</p>  <p>3. Klik button 'Login' untuk menyelesaikan proses login</p>
Kriteria Keberhasilan	Sistem akan menampilkan halaman dashboard user
Kondisi khusus	Username dan password sudah diberikan oleh admin
Hasil	Valid : Yes Invalid : No

Tabel 4.6: Tabel Pengujian Cari Kandidat

Nomor Test	PF-03
Pengguna	User
Judul	Cari Kandidat
Teknik	<p>1. Klik “Cari Kandidat” pada menu yang ada di sebelah kiri</p>  <p>2. Inputkan persyaratan umum seperti jabatan yang akan dituju atau posisi target, kemudian dengan minimal jabatan dan kerja minimal.</p>  <p>3. Klik button ‘Cari’ untuk melakukan proses pencarian</p>
Kriteria Keberhasilan	Sistem akan menampilkan nama-nama kandidat
Kondisi khusus	-
Hasil	Valid : Yes Invalid : No

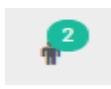
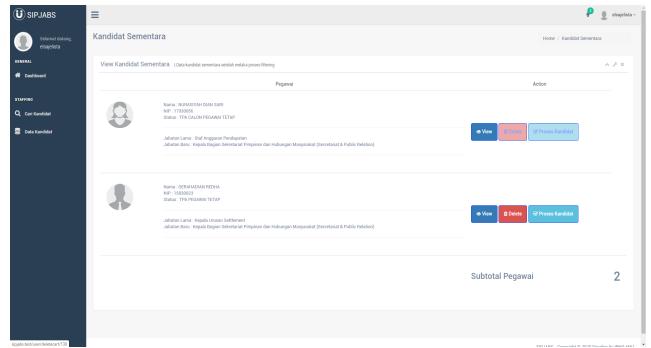
Tabel 4.7: Tabel Pengujian Filtering

Nomor Test	PF-04
Pengguna	User
Judul	Melakukan Filtering
Teknik	<p>1. Klik button “Cari” untuk memulai mencari kandidat</p> <p>2. Maka sistem akan menampilkan halaman filter kandidat seperti dibawah ini sesuai persyaratan yang telah di inputkan sebelumnya.</p>  <p>3. User dapat memilih persyaratan lebih spesifik untuk menfilter kandidat seperti gambar dibawah ini.</p>  <p>4. Lalu sistem akan menampilkan nama-nama kandidat dari hasil filter secara spesifik.</p> 
Kriteria Keberhasilan	Sistem akan menampilkan nama-nama kandidat
Kondisi khusus	-
Hasil	Valid : Yes Invalid : No

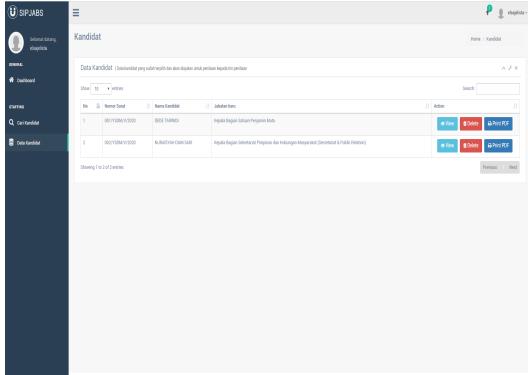
Tabel 4.8: Tabel Pengujian Pilih Kandidat

Nomor Test	PF-05
Pengguna	User
Judul	Melakukan Pemilihan Kandidat
Teknik	<p>1. Setelah menfilter kandidat, user dapat memilih kandidat untuk dijadikan kandidat sementara.</p> <p>2. Sistem akan menampilkan beberapa nama kandidat yang telah terfilter, kemudian user dapat klik button pilih pada salah satu nama kandidat yang akan dipilih.</p>  <p>3. Data tersebut akan masuk kedalam kandidat sementara</p> 
Kriteria Keberhasilan	Sistem akan menampilkan pop-up berhasil ditambahkan kedalam kandidat sementara
Kondisi khusus	-
Hasil	Valid : Yes Invalid : No

Tabel 4.9: Tabel Pengujian Proses Kandidat

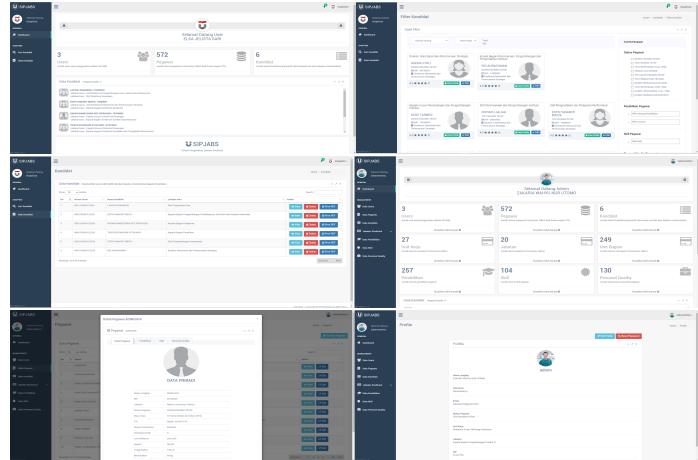
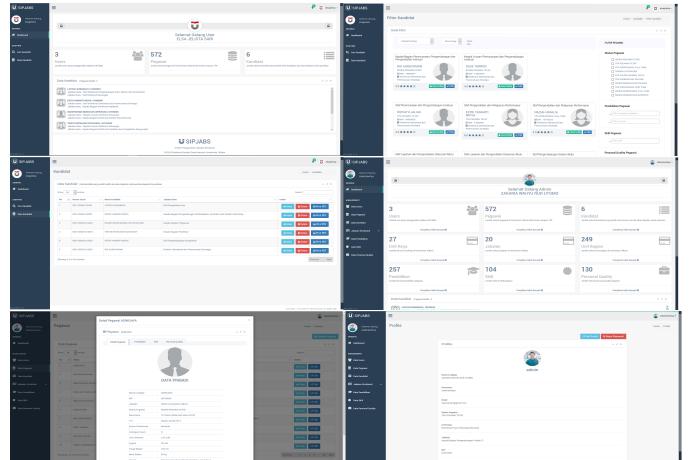
Nomor Test	PF-06
Pengguna	User
Judul	Melakukan Pemrosesan Kandidat
Teknik	<p>1. Untuk melanjutkan proses kandidat yang sudah dipilih tadi maka user dapat memprosesnya dengan mengklik icon “Kandidat Sementara” berada pada pojok kanan atas.</p>  <p>2. Kemudian untuk melihat nama-nama kandidat sementara klik button “View Kandidat Sementara”</p> <div style="text-align: center; margin: 10px 0;"> View Kandidat Sementara </div> <p>3. Maka sistem akan menampilkan halaman kandidat sementara dengan detail data yang telah ditentukan sebelumnya</p>  <p>4. Kemudian user dapat memilih nama yang akan di proses untuk dijadikan kandidat dengan mengklik button ”Proses kandidat”</p> <p>5. Maka data akan masuk ke dalam data kandidat</p>
Kriteria Keberhasilan	Sistem akan menampilkan pop-up berhasil memproses kandidat
Kondisi khusus	-
Hasil	Valid : Yes Invalid : No

Tabel 4.10: Tabel Pengujian Cetak PDF

Nomor Test	PF-07
Pengguna	User
Judul	Melakukan Cetak PDF
Teknik	<p>1. Untuk mencetak hasil kandidat maka user harus masuk ke halaman data kandidat dengan cara klik "Data Kandidat" pada menu.</p>  <p>2. Kemudian sistem akan menampilkan halaman data kandidat dengan nama-nama kandidat yang sudah diproses</p>  <p>3. Lalu user dapat mencetak hasil kandidat dengan klik button "Print PDF" pada salah satu nama kandidat yang hasilnya akan dicetak.</p> <p>4. Sistem akan melakukan proses download laporan kandidat dalam bentuk PDF yang dapat diunduh oleh user</p>
Kriteria Keberhasilan	Terdapat file pdf yang bisa di unduh
Kondisi khusus	-
Hasil	Valid : Yes Invalid : No

4.2.1.2 Pengujian Kesesuaian

Tabel 4.11: Tabel Pengujian Kesesuaian

Nomor Test	PK-01
Judul	Melakukan pengujian kesesuaian pada aplikasi
Teknik	1. Membuka aplikasi pada google chrome 2. Membuka aplikasi pada mozilla firefox
Kriteria Keberhasilan	Menampilkan output yang sama di semua platform browser yang digunakan untuk menguji
Alat Pengujian	1. Laptop MSI GL62M 2. Google Chrome 3. Mozilla Firefox
Kondisi khusus	-
Hasil	<p>1. Google Chrome</p>  <p>2. Mozilla Firefox</p> 

4.2.2 Pengujian Beta

Pengujian beta dilakukan dengan pengujian secara langsung kepada pengguna, salah satu pengujian beta yaitu dengan menggunakan Usability Testing. Berikut hasil pengujinya:

Tabel 4.12: Tabel Usability Testing

No	Pertanyaan	Aspek Usability				Total Skor	Likert
		STS	TS	S	SS		
Aspek Fungsionalitas							
1	Apakah semua fitur aplikasi SiPJabS dapat berjalan sesuai dengan fungsinya?			5	3	27	84,3%
2	Apakah aplikasi SiPJabS mudah untuk digunakan dan dioperasikan?			4	4	28	87,5%
3	Apakah respon aplikasi SiPJabS baik dan cepat?		1	5	2	23	78,1%
Rata-rata							83,2%
Aspek Keguaan							
4	Apakah aplikasi SiPJabS mudah untuk mencari kandidat secara tepat dan efektif?		1	6	1	24	75,5%
5	Apakah persyaratan yang diberikan (Filter Pegawai) sudah sesuai dengan job description yang akan dicari?		2	4	2	24	75,5%
6	Apakah aplikasi SiPJabS sudah layak untuk digunakan pada perusahaan?		2	4	2	24	75,5%
Rata-rata							75,5%
Aspek UI/UX							
7	Apakah bentuk button (tombol) pada aplikasi SiPJabS sudah sesuai?		1	5	2	25	78,1%
8	Apakah warna yang diterapkan pada aplikasi SiPJabS sudah sesuai?		2	4	2	24	75,5%
9	Apakah icon pada aplikasi SiPJabS sudah sesuai?			7	1	8	78,1%
Rata-rata							77,3%

Keterangan skor nilai:

1. Sangat Tidak Setuju = 1
2. Tidak Setuju = 2
3. Setuju = 3
4. Sangat Setuju = 4

Rumus : $Hasilakhir = \frac{TotalSkor}{Y} \times 100\%$

- Jumlah responden = 8
- Y = Jumlah Responden x 4

4.3 Diskusi Hasil Pengujian

Pada pengujian aplikasi SiPJabS menggunakan 2 metode yaitu dengan pengujian alpha dan pengujian beta. Untuk pengujian alpa terdiri dari pengujian fungsionalitas dan pengujian kesesuaian. Hasil dari pengujian fungsionalitas adalah baik dan memandakan fungsionalitas dari SiPJabS dapat berfungsi dengan baik sedangkan untuk hasil pengujian keseuaian adalah baik dengan sesuainya aplikasi ketika dibuka dalam Google Chrome dan Mozilla Firefox.

Untuk pengujian beta menggunakan metode usability testing. Usability testing adalah mencari permasalahan dalam penggunaan aplikasi dan dilakukan dengan membuat sebuah kuesioner untuk menguji seberapa jauh pemahaman pengguna terhadap aplikasi serta menentukan kepuasan pengguna dengan aplikasi sipjabs. Berikut hasil usability testing:

1. Fungsionalitas berjalan dengan baik dengan rata-rata skor likert 83,7% dengan data tersebut dapat diketahui bahwa pengguna setuju aplikasi SiPJabS memiliki fungsionalitas baik dan dapat digunakan.
2. Aplikasi membantu dalam pencarian kandidat dengan rata-rata skor likert 75,5% dengan data tersebut dapat disimpulkan aplikasi SiPJabS dapat membantu pengguna dalam pencarian kandidat secara cepat dan efisien.
3. Aspek desain UI/UX memiliki rata-rata skor likert 77,3 % dengan data tersebut aspek UI/UX dapat disimpulkan bahwa pengguna setuju desain UI/UX aplikasi SiPJabS baik dan dapat digunakan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan rancangan dan dari hasil analisa pengujian aplikasi yang telah dilakukan, adapun kesimpulan aplikasi Sistem Pengawakan Jabatan Struktural adalah sebagai berikut :

1. Aplikasi SiPJabS dibangun dengan menggunakan back-end framework laravel, fornt-end menggunakan bootstrap dan javascript serta dabatase menggunakan mysql. Dan di hosting menggunakan server indonesia yang cukup stabil.
2. Aplikasi SiPJabS dapat membantu pegawai direktorat sumber daya manusia universitas telkom bagian pengembangan karir dalam mencari kandidat baru secara cepat dan efisien.
3. Hasil dari responden adalah setuju dengan fitur aplikasi yang sudah dibuat sesuai dengan tujuan aplikasi.

5.2 Saran

Berdasarkan hasil dari pengujian usability testing, terdapat beberapa saran untuk pengembangan aplikasi SiPJabS agar lebih baik lagi. Adapun saran adalah sebagai berikut:

1. Karena keterbatasan penyampaian data dari pengembangan karir, aplikasi SiPJabS secara kelengkapan data belum lengkap dan belum dapat digunakan secara maksimal. Untuk pengembangan selanjutnya, pengumpulan data dapat dibuka saat bekerja di kantor dan diawasi oleh Kepala Urusan Pengembangan Karir.
2. Penambahan fitur tambah data secara banyak menggunakan excel, sehingga tidak perlu menginputkan satu-satu.

DAFTAR PUSTAKA

- [1] Nicho, “Fungsi pengawasan dalam manajemen (controlling) dan jenisnya.” <https://nichonotes.blogspot.com/2018/11/fungsi-pengawasan.html>. Diakses : pada 31 Desember 2019, pukul 09.11 WIB.
- [2] M. M. W. Muhammad Ekhsan, Aziz Fathoni, “Pengaruh pengalaman kerja, posisi staffing dan distribusi rasa keadilan terhadap kepuasan karyawan pt sai apparel industries semarang,” vol. 2017, p. 18, 2017.
- [3] T. H. Handoko, “Manajemen,” vol. 1999, p. 25, 1999.
- [4] Lusika, “Pengelolaan surat elektronik (e-mail) di kantor badan kepegawaian daerah (bkd) provinsi diy,” vol. 2019, p. 13, 2019.
- [5] D. H. S. Alex Surya Rahardjo, “Aplikasi e-commerce dengan menggunakan mysql dan php4. jurnal universitas kristen petra,” vol. 2002, p. 47, 2002.
- [6] H. I. G. Purbasari Ayi, “Pemanfaatan framework laravel dalam pembangunan aplikasi e-travel berbasis website,” *Jurnal STIMIK Atma Luhur PingkalPinang*, vol. 2018, 2018.
- [7] R. Harminingtyas, “Analisis layanan website sebagai media promosi, media transaksi dan media informasi dan pengaruhnya terhadap brand image perusahaan pada hotel ciputra di kota semarang,” vol. 2014, 2014.