

TA - Irfandi (turnitin)

by cekturnitin@yopmail.com 1

Submission date: 28-Mar-2023 04:00AM (UTC+0300)

Submission ID: 2003228015

File name: TA_-_Irfandi_turnitin.pdf (2M)

Word count: 8735

Character count: 55960

RANCANG BANGUN SISTEM INFORMASI MANAJEMEN PEGAWAI BERBASIS MOBILE PADA PT XYZ MENGGUNAKAN FRAMEWORK REACT NATIVE

Oleh
Irfandi Iqbal Abimanyu

RINGKASAN

PT XYZ merupakan Badan Usaha Milik Negara (BUMN) yang bertanggung jawab dalam pembangkitan dan penyediaan listrik di Sumatera bagian tengah dan selatan. Pengelolaan arsip kepegawaian PT XYZ masih dilakukan secara manual. Proses perubahan data dan pencarian berkas pegawai yang lama menyebabkan kinerja proses administrasi pegawai PT XYZ menurun. Permasalahan dalam proses administrasi tersebut dapat diselesaikan dengan membangun sebuah sistem informasi manajemen pegawai berbasis *mobile*. Sistem ini dibangun menggunakan kerangka kerja pengembangan aplikasi *mobile react native* dan komponen *user interface nativebase* serta menggunakan metode *Rapid Application development* (RAD). Sistem informasi ini bertujuan untuk memfasilitasi pegawai dalam proses pencarian data dan memudahkan pegawai ketika ingin mendapatkan data kepegawaian seperti data pribadi, data lamaran, riwayat pendidikan, riwayat mutasi dan riwayat cuti yang berkaitan dengan dirinya secara *real-time*. Pengujian pada sistem ini menggunakan metode *System Usability Scale (SUS)* dan mendapatkan skor akhir 87,1. Sistem ini mendapatkan predikat *acceptable* dengan *grade B* dan mendapatkan *adjective rating excellent*.

2 **BAB I. PENDAHULUAN**

1.1. Latar Belakang

Perkembangan teknologi informasi dewasa ini semakin pesat dan masif, kemajuan teknologi informasi ini memicu berbagai bidang untuk menerapkan teknologi informasi di setiap aktivitasnya. Hampir semua instansi baik pendidikan, pemerintahan, maupun swasta berlomba-lomba menerapkan teknologi informasi untuk memfasilitasi proses pelayanan sehingga mampu memberikan pelayanan yang terbaik bagi pelanggannya atau mempermudah pengelolaan internal instansinya. Salah satu bentuk penerapan teknologi informasi yang umum dijumpai di setiap instansi adalah sistem informasi manajemen pegawai. Sistem informasi manajemen merupakan sebuah sistem yang menyediakan fungsi manajemen seperti perencanaan, pengendalian, dan operasional pada sebuah instansi dengan cara yang efektif sehingga menghasilkan informasi yang dibutuhkan secara cepat, tepat dan akurat untuk membantu proses pengambilan keputusan dengan mudah (Sadikin & Wiranda, 2022). Penerapan sistem informasi manajemen kepegawaian dapat meningkatkan efektivitas pengelolaan sumber daya manusia. Sistem informasi manajemen kepegawaian tersebut dapat mempercepat dan mempermudah pengelolaan data pegawai, serta memudahkan dalam pengambilan keputusan terkait pengelolaan sumber daya manusia (A. Wibisono, 2020).

Pengelolaan arsip merupakan salah satu bagian dari sistem manajemen informasi yang penting dilakukan di sebuah instansi. Arsip yang dikelola dengan baik dan teratur mampu meningkatkan kinerja instansi dalam kegiatan administrasi dan pengambilan keputusan. Apabila arsip tidak dikelola dengan baik maka akan menghambat instansi dalam mengambil keputusan dan memperlambat proses administrasi (Hendriyani, 2021). Arsip yang dikelola di sebuah instansi meliputi semua arsip yang berkaitan dengan instansi tersebut, salah satunya yaitu arsip kepegawaian. Untuk itu, optimalisasi pengelolaan arsip kepegawaian menjadi penting dilakukan pada PT XYZ untuk mempermudah proses administrasi baik perubahan data seperti mutasi dan cuti, maupun pengarsipan data kepegawaian lainnya yang sebelumnya dilakukan secara manual menjadi otomatis menggunakan sebuah sistem informasi manajemen.

36

PT XYZ merupakan Badan Usaha Milik Negara (BUMN) yang bertanggung jawab membangkitkan dan menyediakan listrik di Sumatera bagian tengah dan selatan. Instansi tersebut merupakan pusat administrasi semua pembangkitan yang ada di provinsi Lampung. Saat ini pengelolaan arsip pada PT XYZ masih dilakukan dengan cara manual mulai dari pegawai diterima sampai dengan pegawai pensiun. Pemberkasan arsip pegawai masih disimpan di dalam ordner map dan dikelompokkan berdasarkan tahun masuk pegawai. Proses perubahan data, mutasi, cuti dan pencarian berkas pegawai yang memakan waktu lama menyebabkan menurunnya kinerja proses administrasi pegawai di PT XYZ.

Permasalahan dalam proses administrasi tersebut dapat diselesaikan dengan membangun sebuah sistem informasi manajemen pegawai berbasis *mobile*. Sistem ini secara utuh terdiri dari 2 bagian utama yaitu *web* dan *mobile*. Penulis hanya berfokus pada pengembangan sistem berbasis *mobile* yang menggunakan *back-end* (REST API) yang sama dengan sistem *web*. Sistem ini dapat mempermudah pegawai ketika ingin mendapatkan data yang diinginkan tanpa harus menunggu proses pencarian di bagian SDM kapan saja dan dimana saja. Selain itu, sistem ini dapat menyediakan data pegawai yang dibutuhkan secara *real-time* ketika proses perubahan data seperti mutasi dan cuti. Sistem akan dibangun menggunakan bahasa pemrograman *JavaScript*, dengan kerangka kerja pengembangan aplikasi *mobile* *react native* dan komponen *user interface nativebase*. Manfaat yang diharapkan dari tugas akhir ini adalah untuk memfasilitasi kebutuhan data pegawai mulai dari data pribadi, data lamaran, riwayat pendidikan, riwayat mutasi sampai dengan riwayat cuti.

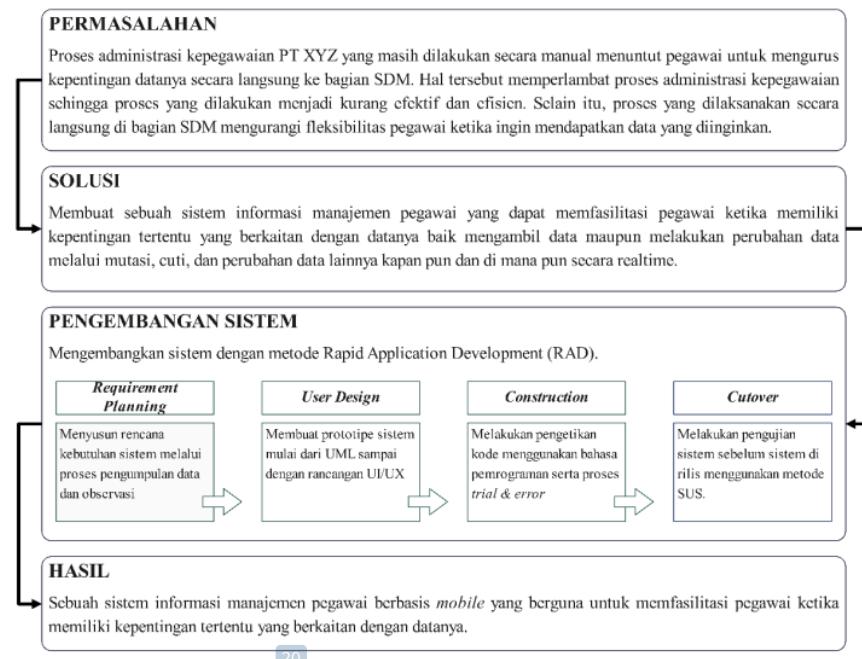
1.2. Tujuan

53

Tujuan tugas akhir ini adalah membangun sebuah sistem informasi manajemen pegawai berbasis *mobile* untuk diterapkan di PT XYZ. Sistem informasi ini bertujuan untuk memfasilitasi pegawai dalam proses pencarian data dan memudahkan pegawai ketika ingin mendapatkan data kepegawaian seperti data pribadi, data lamaran, riwayat pendidikan, riwayat mutasi dan riwayat cuti yang berkaitan dengan dirinya secara *real-time*.

1.3. Kerangka Pemikiran

Pengelolaan arsip kepegawaian pada PT XYZ saat ini masih dilakukan secara manual. Penyimpanan datanya masih disimpan di dalam lemari dan belum dicadangkan sehingga berisiko hilang. Hal ini dapat menurunkan kinerja proses administrasi instansi yang dilakukan oleh admin SDM ketika melayani pegawai yang membutuhkan datanya. Selain itu, proses pelayanan yang hanya dapat dilakukan di hari kerja dapat mengurangi fleksibilitas akses pegawai terhadap perubahan datanya. Untuk mengatasi permasalahan tersebut perlu adanya sebuah sistem informasi yang dapat membantu pegawai untuk mengakses data yang diinginkan dengan cepat kapan saja dan di mana saja secara *real-time*.⁶² Sebuah solusi yang muncul untuk menyelesaikan permasalahan tersebut adalah “Rancang Bangun Sistem Informasi Manajemen Pegawai Berbasis Mobile Pada PT XYZ Menggunakan Framework React Native” dengan menggunakan metode pengembangan RAD (*Rapid Application Development*). Tahapan kerangka pemikiran dalam pembuatan sistem ini disajikan pada Gambar 1.



20
Gambar 1. Kerangka pemikiran

1.4. Kontribusi

Sistem informasi manajemen pegawai berbasis *mobile* ini mampu memberikan kontribusi untuk beberapa pihak sebagai berikut:

1. Kontribusi bagi PT XYZ
 - a. Memfasilitasi proses administrasi pada PT XYZ menggunakan sebuah sistem informasi yang sebelumnya manual sehingga data yang disajikan telah sesuai dengan data yang terbaru secara *real-time*.
 - b. Mempermudah pegawai dalam mengakses datanya melalui perangkat *mobile* mereka tanpa harus mengurusnya langsung ke bagian SDM.
2. Kontribusi bagi Politeknik Negeri Lampung
 - a. Meningkatkan dan mengembangkan ilmu pengetahuan⁷³
 - b. Menjadi bahan rujukan dan referensi bagi mahasiswa yang akan atau sedang melakukan penelitian tugas akhir dengan tema sejenis

6 **BAB II. TINJAUAN PUSTAKA**

2.1. Sistem Informasi Manajemen

Sistem informasi manajemen merupakan suatu metode pengorganisasian yang saling berkaitan dan saling berinteraksi antar komponen dalam sebuah kesatuan untuk menghasilkan informasi yang dibutuhkan manajer. Sistem informasi manajemen memfasilitasi proses pengambilan keputusan dalam menerapkan fungsi manajemen seperti perencanaan, pengendalian, dan pengorganisasian suatu instansi supaya dapat dilaksanakan dengan cepat dan efektif. Sistem informasi manajemen bertujuan untuk mengatasi berbagai permasalahan terkait dengan pengaturan suatu instansi dalam menjalankan proses bisnisnya seperti permasalahan layanan, biaya operasional, strategi bisnis, dan proses operasional instansi lainnya. Terdapat 5 komponen utama yang saling berkaitan dalam satu kesatuan dalam sistem informasi manajemen yaitu manusia (*brainware*), prosedur bisnis, data, perangkat keras (*hardware*) dan perangkat lunak (*software*) (Sadikin & Wiranda, 2022). Berikut ini manfaat penggunaan sistem informasi manajemen (Hutahaean dkk., 2021).

1. Mempermudah dalam perencanaan sehingga lebih efektif dan efisien.
2. Menjadi sarana untuk menganalisis pelaksanaan dan keperluan.
3. Meningkatkan produktivitas dan kinerja instansi.
4. Mengurangi biaya operasional instansi.
5. Menghasilkan informasi yang aktual dan *real-time* bagi pengguna yang membutuhkan tanpa perantara.

2.2. Arsip

Arsip adalah semua rekaman yang terekam dalam berbagai media baik tertulis, berupa gambar, atau berupa rekaman (dalam bentuk audio atau video) yang dihasilkan oleh suatu instansi. Suatu dokumen dapat dianggap sebagai arsip jika dalam dokumen tersebut mengandung informasi yang penting bagi sebuah instansi pada masa lalu, ²³ masa kini, maupun ⁴⁸ di masa yang akan datang. Pengelolaan arsip di sebuah instansi harusnya dikelola dengan baik dan teratur untuk menghindari

kesalahpahaman informasi yang disajikan sehingga pengambilan keputusan dapat dilakukan dengan benar (Rosalin, 2017).

37

Menurut Undang-Undang Republik Indonesia No. 43 Tahun 2009 tentang

5

Kearsipan, Arsip dibagi menjadi beberapa jenis yaitu :

1. Arsip dinamis merupakan arsip yang dapat digunakan secara langsung dan dalam jangka waktu tertentu.
2. Arsip statis merupakan arsip yang memiliki nilai kesejarahan atau yang sudah habis masa retensinya.
27
3. Arsip vital merupakan arsip yang dipergunakan sebagai syarat dasar dalam proses operasional. Arsip ini tidak dapat diperbarui, dan keberadaannya tidak dapat digantikan apabila hilang atau rusak.
5
4. Arsip aktif merupakan arsip yang sering digunakan secara terus-menerus.
5. Arsip terjaga merupakan arsip yang berkaitan dengan negara yang keberadaannya mempengaruhi keberlangsungan hidup bangsa dan negara yang harus dilindungi, dijaga keamanannya, dan keutuhannya.
6. Arsip umum merupakan arsip yang berkategori selain arsip terjaga.

Berikut ini beberapa jenis arsip lainnya (Sugiarto & Wahyono, 2014).

61

1. Arsip berdasarkan subjeknya yaitu arsip kepegawaian, arsip pemasaran, arsip keuangan, arsip pendidikan, dan sebagainya.
2. Arsip berdasarkan bentuk medianya yaitu berkas surat, arsip digital, rekaman baik audio maupun video, berkas gambar, dan lain-lain.
3. Arsip berdasarkan nilai kegunaannya yaitu :
4. Arsip yang bernilai administrasi seperti prosedur kerja.
5. Arsip yang bernilai hukum seperti akta kelahiran, akta tanah, dll.
6. Arsip yang bernilai pendidikan seperti silabus, kurikulum, dll.
19
7. Arsip yang bernilai sejarah seperti laporan bulanan, tahunan, dll.
8. Arsip yang bernilai keuangan seperti kuitansi, nota pembayaran, dll.
19
9. Arsip yang bernilai informasi seperti pengumuman dan undangan.
10. Arsip yang bernilai ilmiah seperti laporan penelitian, skripsi, dll.
3
11. Arsip berdasarkan fungsinya yaitu arsip dinamis dan arsip statis.
12. Arsip berdasarkan tempat pengelolaannya yaitu arsip pusat dan arsip unit.

13. Arsip berdasarkan tingkatan keasliannya yaitu arsip asli, arsip tembusan, arsip salinan, dan arsip petikan.
14. Arsip berdasarkan kekuatan hukumnya yaitu arsip otentik, dan arsip tidak otentik.
15. Arsip berdasarkan kepentingannya yaitu :
 - a. Arsip yang tidak berguna seperti surat undangan.
 - b. Arsip yang berguna seperti surat cuti, surat izin, dan presensi pegawai.
 - c. Arsip yang penting seperti laporan keuangan, surat keputusan, dll.
 - d. Arsip yang vital seperti akta kelahiran, ijazah, sertifikat pelatihan, dll.

Salah satu jenis arsip berdasarkan subjeknya yaitu arsip kepegawaian. Arsip kepegawaian merupakan semua arsip yang ada hubungannya dengan masalah kepegawaian, seperti surat lamaran, daftar riwayat hidup, data pribadi, surat-surat keputusan pegawai, absensi pegawai, dan lain-lain (Suparman, 2020).

2.3. Aplikasi Mobile

Perangkat lunak (*software*) merupakan serangkaian detail instruksi yang ditulis menggunakan bahasa pemrograman untuk memberitahu komputer apa yang harus dilakukan guna mencapai sebuah tujuan tertentu. Keberadaan perangkat lunak berperan penting sebagai penghubung antara pengguna dengan perangkat keras (*hardware*) komputer, karena tanpa adanya perangkat lunak komputer hanyalah kumpulan perangkat keras yang tidak dapat dioperasikan (Fox, 2013).

Secara umum perangkat lunak dibagi menjadi dua jenis yaitu:

1. Perangkat lunak sistem

Perangkat lunak sistem merupakan kumpulan program atau instruksi yang dibuat untuk membentuk sebuah sistem operasi yang menghubungkan perangkat keras komputer dengan perangkat lunak aplikasi. Komponen inti dari sebuah perangkat lunak sistem disebut sebagai *kernel* yang dijalankan saat pertama kali komputer dihidupkan. *Kernel* memiliki beberapa tugas yaitu menangani manajemen proses, manajemen sumber daya, dan manajemen memori pada komputer. Tanpa adanya *kernel* perangkat lunak aplikasi tidak dapat dijalankan secara efisien atau bahkan tidak dapat dijalankan sama sekali.

Beberapa contoh sistem operasi yang umum digunakan adalah *linux*, *windows*, dan *macOS*.

2. Perangkat lunak aplikasi

Perangkat lunak aplikasi adalah kumpulan program yang dijalankan pengguna untuk menyelesaikan beberapa tugas atau fungsi-fungsi lainnya yang berjalan di atas perangkat lunak sistem.⁷⁸ Pengelompokan perangkat lunak aplikasi bergantung pada kegunaannya bagi pengguna.

- a. Aplikasi pendukung produktivitas yang digunakan semua orang mencakup aplikasi pengolah kata, aplikasi pengolah angka, aplikasi presentasi, sistem manajemen data, aplikasi kontak, aplikasi pengatur data dll.
- b. Aplikasi desain yang digunakan para desainer mencakup aplikasi menggambar, aplikasi edit gambar, aplikasi pengolah *vector*, aplikasi 3D, dan aplikasi pengolah video.
- c. Aplikasi pengolah suara yang digunakan para musisi.
- d. Aplikasi pengembangan yang digunakan para *programmer* mencakup aplikasi editor kode, aplikasi IDE, aplikasi pelacakan perubahan kode, *library* atau *framework*, aplikasi *debugging*, aplikasi visualisasi kode, dll.
- e. Aplikasi daring mencakup aplikasi yang menggunakan koneksi internet ketika penggunaannya seperti aplikasi email, browser web, aplikasi FTP, aplikasi *remote*, dsb.
- f. Aplikasi lainnya mencakup aplikasi di luar kategori aplikasi yang dijelaskan di atas seperti aplikasi hiburan (pemutar musik dan video), dan permainan.

Perangkat lunak dapat berjalan di beberapa *platform* seperti komputer, perangkat *mobile*, perangkat jaringan, atau bahkan perangkat *embedded system*.¹¹ Perangkat lunak yang dijalankan menggunakan perangkat *mobile* disebut aplikasi *mobile*. Aplikasi *mobile* merupakan kumpulan beberapa perangkat lunak atau program yang berjalan pada perangkat *mobile* yang mampu menjalankan tugas-tugas tertentu berdasarkan keinginan pengguna. Aplikasi *mobile* merupakan terobosan terbaru akibat berkembang pesatnya teknologi informasi dan komunikasi. Kemudahan penggunaan, tampilan yang *user-friendly*, mudah didapat, mudah diunduh, dan dapat dijalankan di berbagai perangkat *mobile* merupakan kelebihan aplikasi *mobile*. Penggunaannya yang luas mulai dari sarana komunikasi, menjelajah internet, jejaring sosial, pendidikan, bisnis, sampai dengan hiburan

mampu terpenuhi oleh aplikasi *mobile*. Jumlah pasar aplikasi *mobile* yang besar berbanding lurus dengan bertambahnya jumlah pengembang, penerbit, dan penyedia aplikasi menjadikan aplikasi *mobile* sebuah teknologi baru yang menguntungkan penggunanya (Islam dkk., 2010).

Aplikasi *mobile* saat ini dapat dikembangkan dengan berbagai bahasa pemrograman seperti *Java*, *Kotlin*, *Dart*, *Objective-C*, *Swift*, dan *JavaScript*. Selain itu para pengembang telah berhasil mengembangkan berbagai *framework* yang dapat digunakan untuk mengembangkan aplikasi *mobile* baik Android, IOS, maupun lintas *platform* dengan stabil dan lebih cepat. Salah satu *framework* yang bisa digunakan untuk mengembangkan aplikasi *mobile* yaitu *React Native* dengan bahasa pemrograman *JavaScript*.

2.3.1. *JavaScript*

JavaScript merupakan bahasa pemrograman tingkat tinggi dinamis yang mampu dikembangkan dengan gaya pemrograman fungsional atau gaya pemrograman berorientasi objek. *JavaScript* merupakan bahasa pemrograman yang memerlukan interpreter untuk menjalankan kodennya. Dahulu *JavaScript* hanya dapat dijalankan di dalam lingkungan browser, namun satu dekade terakhir muncul *runtime-environment* yang memfasilitasi *JavaScript* supaya dapat berjalan di luar lingkungan browser yaitu *Node.js*. Keberhasilan *Node.js* membawa *JavaScript* keluar dari lingkungan browser membuat *JavaScript* sekarang menjadi bahasa pemrograman yang paling banyak digunakan di kalangan pengembang perangkat lunak (Flanagan, 2020). Selain pengembangan web, saat ini para pengembang *JavaScript* telah mengembangkan *framework* yang dapat digunakan untuk mengembangkan aplikasi *mobile*. *Framework* yang populer yaitu *React Native* dan *Ionic*.

2.3.2. *React Native*

React Native adalah salah satu kerangka kerja (*framework*) *JavaScript* besutan Facebook yang digunakan untuk mengembangkan aplikasi baik aplikasi *mobile* (Android dan IOS) maupun aplikasi web. Kerangka kerja *React Native* dibuat berdasarkan *library React*. Dengan kata lain, *React Native* memungkinkan pengembang web untuk membuat aplikasi *mobile* yang mirip seperti aplikasi *mobile* yang dibuat secara *native* tanpa harus beralih ke kerangka kerja dan bahasa

11
pemrograman yang lain. Mirip seperti *React*, *React Native* ditulis menggunakan campuran bahasa pemrograman *JavaScript* dan *markup XML-esque* atau lebih dikenal sebagai *JSX*. *React Native* menghubungkan kode *JSX* yang ditulis dengan API *rendering* asli milik masing-masing *platform* sehingga aplikasi yang dibuat akan ditampilkan menggunakan komponen UI masing-masing *platform* dan akan terasa seperti aplikasi *mobile* lainnya (Eisenman, 2015).

2.3.3. *NativeBase*

Library komponen UI/UX (*User Interface/User Experience*) merupakan kumpulan serangkaian kode komponen yang siap digunakan pengguna untuk kebutuhan tertentu. Dengan menggunakan komponen yang sudah disediakan memungkinkan pengguna untuk menyusun tata letak tampilan dan pengalaman pengguna dengan cepat tanpa membuang waktu untuk merancangnya dari awal. Selain itu keuntungan menggunakan *library* komponen sering kali menghasilkan hasil yang lebih stabil dan konsisten baik dari segi UI maupun UX. Salah satu *library* yang populer digunakan dalam pengembangan aplikasi *React Native* adalah *NativeBase* (Boduch dkk., 2022).

2.3.4. *Application Programming Interface*

59
Application Programming Interface (API) merupakan kumpulan aturan yang ditentukan oleh pengembang guna memfasilitasi aplikasi untuk dapat berkomunikasi dengan aplikasi lain. API bertindak sebagai perantara yang memproses pertukaran data antar sistem atau aplikasi. Hal ini memungkinkan perusahaan membuka akses dan fungsionalitas aplikasinya untuk dapat dikembangkan atau dihubungkan dengan pengembang pihak ketiga, mitra bisnis, atau divisi internal perusahaan. Manfaat mengembangkan API adalah untuk membantu proses bisnis perusahaan dengan menghubungkan banyak aplikasi berbeda yang digunakan sehingga menghemat waktu penggerjaan dan memudahkan pengembang berkolaborasi dan berinovasi (IBM Cloud Education, 2022).

API menggunakan beberapa protokol yang sering dijumpai dalam pengembangan aplikasi yaitu:

1. **SOAP (*Simple Object Access Protocol*)**

API SOAP dibangun dengan menggunakan XML yang memungkinkan *endpoint* untuk mengirim dan menerima data melalui SMTP dan HTTP.

2. **XML-RPC (*XML-Remote Procedure Call*)**

Protokol XML-RPC menggunakan format XML tertentu untuk mentransfer datanya. XML-RPC dibuat sebelum adanya SOAP, akan tetapi jauh lebih sederhana, dan relatif ringan karena menggunakan *bandwidth* yang kecil.

3. **JSON-RPC (*JSON-Remote Procedure Call*)**

JSON-RPC hampir sama dengan XML-RPC, pembedanya hanya penggunaan JSON (*JavaScript Object Notation*) untuk mentransfer data dan tidak lagi menggunakan XML.

4. **REST (*Representational State Transfer*)**

REST adalah seperangkat prinsip pengembangan arsitektur web API.

2.3.5. *Representational State Transfer*

Representational State Transfer atau REST merupakan arsitektur web yang menyediakan proses pertukaran data antara *server* dan *client* dengan menggunakan API yang terhubung secara *point-to-point*. Arsitektur REST menggunakan prinsip *client-server*. Dengan kata lain *server* REST bertanggung jawab untuk menyediakan sumber daya yang diminta oleh *client* REST ketika melakukan pembuatan, pengambilan, pengubahan, dan penghapusan sumber daya. Arsitektur REST menggunakan format XML dan JSON (*JavaScript Object Notation*) sebagai media pertukaran datanya (Chatterjee & Mamatha, 2020). Ada beberapa prinsip yang perlu diperhatikan ketika mengembangkan REST API yaitu:

1. ***Addressability***

Setiap sumber daya harus memiliki setidaknya satu URI (*Uniform Resource Identifier*) yang terkait. URI digunakan untuk menentukan sumber daya atau sekumpulan sumber daya.

2. *Statelessness*

Layanan REST adalah layanan yang independen. Setiap permintaan yang dikirimkan menggunakan protokol HTTP (*Hypertext Transfer Protocol*) tidak berhubungan dengan permintaan sebelumnya.

3. *Cacheable*

Sumber daya yang ditandai sebagai *cache* dapat disimpan sementara dalam sistem dan dapat digunakan kembali ketika permintaan menghasilkan hasil yang sama. Penandaan sumber daya dapat dilakukan ketika permintaan dimulai.

4. *Uniform Interface*

Layanan REST menggunakan kumpulan metode HTTP standar untuk menentukan permintaan yang dibuat. Berikut ini beberapa metode HTTP sederhana yang dapat digunakan dalam layanan REST. Berikut ini metode HTTP yang umum digunakan.

Tabel 1. Metode HTTP

Metode	Penjelasan
GET	Digunakan untuk mengambil data (<i>Read</i>).
POST	Digunakan untuk <i>input</i> data (<i>Create</i>).
PUT	Digunakan untuk <i>input</i> data, apabila data sudah ada maka data tersebut akan diperbarui (<i>Create/Update</i>).
PATCH	Digunakan untuk memperbarui data (<i>Update</i>)
DELETE	Digunakan untuk menghapus data (<i>Delete</i>)

Sumber: (MDN Mozilla, 2022)

60

2.4. Perancangan Sistem

Perancangan sistem merupakan proses merinci pembuatan sistem menggunakan berbagai teknik, mulai dari deskripsi arsitektur pendukung dan detail komponen sampai dengan kendala pengerjaan yang bertujuan untuk memberikan gambaran yang jelas dalam memenuhi kebutuhan pengguna. Perancangan sistem biasanya digambarkan dalam bentuk diagram alir yang menampilkan proses secara runtut dan diagram-diagram pendukung lainnya (Aziz, N., Pribadi, G., & Nurcahya, 2020).

2.4.1. *Mapping Chart*

Mapping chart merupakan diagram alir yang memvisualisasikan langkah-langkah dalam bentuk simbol-simbol grafis yang digunakan untuk membantu proses analisis, penelitian, atau untuk memecahkan masalah tertentu. *Mapping chart* berfungsi untuk menyederhanakan proses agar lebih mudah dilihat dan dipahami menggunakan media simbol-simbol (Mukodimah dkk., 2019).

7
Tabel 2. Simbol-simbol *mapping chart*

Simbol	Nama	Fungsi
(1)	(2)	(3)
	Dokumen	Menyatakan <i>input</i> dokumen yang dicetak
	Dokumen rangkap	Menyatakan dokumen yang lebih dari satu
	Connector	Menyatakan sambungan satu dengan yang lainnya pada proses berikutnya
	Proses	Proses pengolahan yang ditugaskan oleh komputer
	Proses manual	Proses pengolahan yang dilakukan secara manual
	Display	Menyatakan output yang digunakan
	Arsip	Mengarsipkan data di dalam program

7
Tabel 2. Lanjutan simbol-simbol *mapping chart*

Simbol	Nama	Fungsi
(1)	(2)	(3)
	Terminal	Awalan dan akhiran pada program
	Desicion	7 Keputusan ya atau tidak dalam proses pengolahan data
	Connecting Line	Menghubungkan simbol dengan simbol lainnya dengan menyatakan suatu alur proses

Sumber: (Fitriyana & Susianto, 2018)

12 2.4.2. *Unified Modelling Language*

Unified Modelling Language atau UML merupakan pemodelan visual yang digunakan untuk mendefinisikan, memvisualisasikan, dan mendokumentasikan rancangan analisis dan desain aplikasi, serta memvisualisasikan arsitektur dalam OOP (*Object Oriented Programming*). UML menjadi sarana yang digunakan untuk memberikan gambaran jelas terkait analisis sistem sebelum dibangun baik secara struktural maupun secara fungsional (Putra & Andriani, 2019). Dalam UML terdapat beberapa diagram yang sering digunakan yaitu:

1. *Use Case Diagram*

Use case diagram merupakan diagram yang memodelkan perlakuan (*behavior*) sistem yang akan dibuat kepada penggunaanya. *Use case* digunakan untuk mendeskripsikan interaksi antara satu atau lebih pengguna dengan sebuah sistem. Melalui *use case* pengembang mendapat gambaran bagaimana interaksi pengguna dengan sistem dan fungsi apa saja yang ada pada sistem tersebut. Notasi *use case* dibagi menjadi beberapa bagian yaitu pengguna (*actor*), sistem atau sub-sistem (*use case*), dan hubungan (*relationship*).

6
Tabel 3. Simbol-simbol diagram *use case*

Simbol	Nama	Fungsi
	Aktor	Menggambarkan pengguna yang berinteraksi dan berperan dalam sistem bisnis.
	Use Case	Menggambarkan fungsi yang dapat dilakukan sebuah sistem bisnis.
	Batas Sistem	Menggambarkan batasan sistem bisnis untuk kasus penggunaan khusus.
	<i>Association</i>	Relasi asosiasi antara <i>use case</i> dengan pengguna.
	<i>Includes</i>	Relasi ketika sebuah <i>use case</i> digambarkan menggunakan fungsionalitas dari <i>use case</i> yang lain.
	<i>Extends</i>	Relasi yang digunakan untuk menyertakan perilaku opsional dari <i>use case</i> yang diperluas.

Sumber: (Visual Paradigm, 2022b)

56

2. *Activity Diagram*

Activity diagram merupakan diagram yang memvisualisasikan alir kerja (*workflow*) atau aktivitas dari sebuah sistem yang menjalankan proses bisnis atau menu yang ada pada perangkat lunak. Dengan kata lain *activity diagram* hanya menggambarkan aktivitas yang berjalan pada sistem bukan aktivitas yang dilakukan pengguna (*actor*).

Tabel 4. Simbol-simbol *activity diagram*

Simbol	Nama	Fungsi
●	<i>Initial Node</i>	Menggambarkan titik awal dari serangkaian tindakan atau kegiatan.
	<i>Activity</i>	Menggambarkan serangkaian tindakan atau aktivitas sistem.
	<i>Decision Node</i>	Menggambarkan seleksi kondisi untuk memastikan bahwa aliran kontrol atau aliran aktivitas hanya turun satu jalur.
	<i>Fork Node</i>	Membagi aktivitas sistem menjadi serangkaian aktivitas (atau tindakan) paralel atau bersamaan.
	<i>Join Node</i>	Menyatukan kembali aktivitas atau tindakan sistem yang paralel atau bersamaan.
	<i>Swimlane and Partition</i>	Mengelompokkan aktivitas yang dilakukan oleh aktor yang sama atau pengelompokan aktivitas dalam satu partisi.
○	<i>Activity Final Node</i>	Menghentikan semua aliran kontrol dan aliran aktivitas atau tindakan

Sumber: (Visual Paradigm, 2022a)

29

2.5. Rapid Application Development (RAD)

Rapid Application Development atau RAD merupakan salah satu metode pengembangan sistem perangkat lunak yang bersifat bertahap (*incremental*). Dalam penerapannya, metode RAD mementingkan waktu siklus pengembangan yang relatif singkat, cepat, pendek, dan berulang (*iterative*) (Kaban dkk., 2022). Metode RAD menggunakan pendekatan yang berfokus pada tahap perancangan dan pembuatan prototipe dengan tujuan mendapatkan umpan balik pengguna secara instan. Perulangan yang konstan dalam mengembangkan sistem berdasarkan umpan balik pengguna dan pembaruan yang cepat membantu mencapai hasil yang

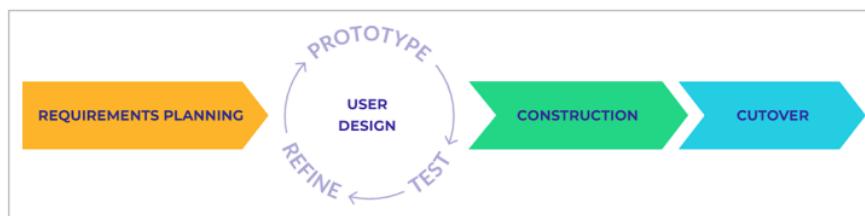
sesuai dengan keinginan pengguna. Berikut ini keuntungan menggunakan metode RAD (Glaschenko, 2021).

1. Menghasilkan kualitas sistem yang tinggi

Keterlibatan pengguna dalam tahap pembuatan prototipe, sistem yang dihasilkan akan lebih relevan dengan keinginan dan ekspektasi pengguna.

2. Meminimalisir biaya dan risiko pengembangan

Ketika menggunakan metode *waterfall*, pengguna hanya dapat melihat hasil dan memberikan umpan balik ketika proyek diluncurkan. Perubahan yang terjadi pada proses ini memakan biaya yang mahal dan memakan lebih banyak waktu. Dengan metode pengembangan aplikasi yang cepat, risiko penulisan ulang dari proyek yang telah diluncurkan menjadi lebih minim.



Gambar 2. Fase metode RAD

Metode RAD memiliki beberapa fase dalam pengerjaannya. Berikut ini fase yang harus dilalui ketika mengembangkan sebuah sistem menggunakan metode RAD.

1. *Requirements Planning*

Pada fase ini, pengguna dan tim proyek mengidentifikasi tujuan sistem yang diinginkan. Pengembangan nantinya berfokus untuk mencapai tujuan bisnis dengan kebutuhan yang telah ditetapkan. Kebutuhan yang sebelumnya sudah ditetapkan dapat diubah dan disesuaikan kembali pada tahap pembuatan prototipe.

2. *User Design*

User design merupakan fase penting dari metode RAD. Pada fase ini, pengembang dimulai dengan mengerjakan prototipe. Tujuan membuat prototipe adalah untuk mendemonstrasikan sebuah gagasan tentang sistem yang akan dibuat kepada klien dengan cepat. Tim mengumpulkan semua

umpan balik dari pengguna berdasarkan prototipe yang telah disampaikan. Pada proses inilah kebutuhan awal dapat disesuaikan kembali berdasarkan keinginan pengguna. Dengan umpan balik yang telah dikumpulkan, pengembang akan mengulang langkah pembuatan prototipe, hingga pengguna puas dengan hasilnya.

3. *Construction*

Pada fase ini, pengembangan dan pengujian dilakukan untuk mempersiapkan sistem menuju ke tahap produksi. Dalam pengerjaannya fokus utamanya adalah pada kualitas, skalabilitas, pemeliharaan, dll. Namun, pengguna terus berpartisipasi pada fase ini dan terus memberikan umpan balik saat fitur diimplementasikan. Sedikit penyempurnaan dimungkinkan pada fase ini.

4. *Cutover*

Cutover merupakan fase terakhir. Dalam fase ini mencakup *acceptance testing*, peluncuran (*rollout*) dan pelatihan pengguna.

2.6. *System Usability Scale (SUS)*

Pengujian sistem merupakan proses mencari kesalahan pada setiap komponen sistem, melakukan pencatatan, mengevaluasi semua aspek dan fitur-fitur sistem yang sedang dikembangkan. Berikut ini beberapa aturan dalam pengujian sistem (W. Wibisono & Baskoro, 2002).

1. Pengujian dilakukan dengan mengeksekusi program dengan tujuan menemukan kesalahan
2. Sebuah kasus pengujian dikatakan baik jika dalam proses pengujinya berpotensi menemukan kesalahan lebih tinggi.
3. Pengujian dikatakan berhasil jika menemukan kesalahan.

System Usability Scale merupakan metode evaluasi sebuah aplikasi untuk menilai dan mengukur tingkat kegunaannya dengan menggunakan sebuah kuesioner sederhana yang memiliki sepuluh pernyataan. Pernyataan metode SUS yang bermotor ganjil menggunakan kalimat positif dan pernyataan yang bermotor genap menggunakan kalimat negatif. Selanjutnya untuk menilai hasil kuesioner tersebut menggunakan skala *likert* yang telah ditentukan (Muhammad Nur Fauzi dkk., 2022).

Kelebihan menggunakan SUS adalah:

1. SUS mudah digunakan karena tidak menggunakan perhitungan yang rumit.
2. Rentang skor SUS bernilai 0-100.
3. Akurat walau menggunakan sampel yang kecil.
4. Gratis tanpa biaya.

2.7. Jurnal terkait

25

BAB III. METODOLOGI PELAKSANAAN

3.1. Tempat dan Waktu

8

Tugas akhir yang berjudul “Rancang Bangun Sistem Informasi Manajemen Pegawai Berbasis Mobile Pada PT XYZ Menggunakan Framework React Native” ini dilaksanakan pada bulan Juli 2022 hingga bulan Maret 2023 di Politeknik Negeri Lampung. Tugas akhir ini mengambil permasalahan dari tempat Praktik Kerja Lapang (PKL) pada semester sebelumnya di PT XYZ.

52

3.2. Alat dan Bahan

Alat dan bahan merupakan kebutuhan wajib yang perlu disiapkan untuk membangun atau membuat sebuah produk. Pada saat penulisan tugas akhir ini alat dan bahan yang diperlukan untuk membangun sebuah sistem informasi manajemen pegawai berbasis ⁴⁰ mobile adalah sebagai berikut:

3.2.1. Alat

66 Alat yang dibutuhkan dalam penulisan dan pengembangan tugas akhir ini dibagi menjadi dua yaitu:

Tabel 5. Alat

Perangkat Keras (<i>Hardware</i>)	Perangkat Lunak (<i>Software</i>)
Laptop/komputer dan kabel <i>charger</i>	Sistem operasi (<i>Windows 11</i>)
Tetikus (<i>mouse</i>)	<i>Microsoft Word 2021</i>
Papan ketik (<i>keyboard</i>)	<i>Figma/FigJam/EdrawMax</i> <i>Mendeley</i> <i>Google Chrome</i> <i>Visual Studio Code</i> <i>React Native</i> <i>NativeBase</i> <i>Postman/Insomnia</i>

3.2.2. Bahan

Tugas akhir ini memerlukan bahan pendukung yang dibutuhkan dalam penyelesaiannya. Bahan yang diperlukan dapat berupa hasil proses pengumpulan data melalui metode observasi dan wawancara secara langsung dengan pihak PT XYZ mengenai sistem yang saat ini sedang berjalan. Selain itu jurnal ilmiah

pendukung lainnya pun perlu disiapkan sebagai referensi dan sumber rujukan penulisan tugas akhir ini.³⁵

3.3. Metode Pengumpulan Data

Pengumpulan data yang dilakukan penulis untuk “Rancang Bangun Sistem Informasi Manajemen Pegawai Berbasis Mobile Pada PT XYZ Menggunakan Framework React Native” menggunakan metode wawancara secara langsung kepada pihak PT XYZ yang bersangkutan yaitu Ibu Dita selaku pegawai divisi SDM, Bapak Angga selaku manajer dan berkonsultasi dengan Bapak Fahma selaku pegawai divisi Engineering sekaligus pembimbing lapangan penulis ketika PKL. Informasi yang didapatkan sangat berguna dan membantu dalam pengembangan sistem tugas akhir ini.

3.4. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan untuk “Rancang Bangun Sistem Informasi Manajemen Pegawai Berbasis Mobile Pada PT XYZ Menggunakan Framework React Native” menggunakan metode RAD. Berikut ini runtutan fase pengembangan menggunakan metode RAD:

3.4.1. Requirements Planning

Saat merancang sebuah sistem menggunakan metode RAD, hal pertama yang harus dilakukan adalah menyusun rencana kebutuhan sistem melalui pengumpulan data dan observasi pada proses bisnis yang sedang berjalan. Data atau informasi yang telah didapatkan dari pengguna akan digunakan sebagai referensi awal kebutuhan sistem.¹⁴

3.4.2. User Design

Pengembang akan membuat sebuah prototipe terkait sistem yang akan dibuat. Pembuatan prototipe dimulai dari UML (*use case* dan *activity diagram*), sampai dengan rancangan UI/UX (*user interface/user experience*). Prototipe ini akan terus dikonsultasikan dengan pengguna untuk menyesuaikan kebutuhan sampai pengguna merasa cukup.

3.4.3. Construction

Pengembang akan membangun sistem yang diinginkan pengguna berdasarkan hasil *user design*. Sistem dikembangkan dengan melakukan pengetikan kode menggunakan bahasa pemrograman (*coding*), pengujian fitur-fitur, dan perbaikan terkait *error* atau *bugs* yang terjadi.

3.4.4. Cutover

Setelah sistem selesai dikembangkan pada fase *construction*, sistem akan diulas kembali oleh PT XYZ menggunakan SUS. Apabila masih ada kekurangan maka kekurangan tersebut akan diperbaiki lagi pada fase *user design* dan tahapan akan diulang kembali sampai aplikasi sesuai dengan keinginan pengguna dan lolos pengujian.
43

BAB IV. HASIL DAN PEMBAHASAN

4.1. Hasil dan Pembahasan

Pengerjaan sistem manajemen pegawai berbasis *mobile* yang dibangun menggunakan metode RAD melalui beberapa fase yang harus dilalui sebelum aplikasi dinyatakan jadi dan dapat digunakan.

4.1.1. Requirements Planning

Semua kebutuhan dalam pengembangan sistem secara teknis dikumpulkan dan dipelajari sebagai bahan pertimbangan pengembangan sistem yang diinginkan pengguna, mulai dari proses bisnis yang saat ini sedang berjalan sampai dengan usulan proses bisnis yang baru sesuai dengan kebutuhan pengguna. Kebutuhan pengembangan sistem didapat melalui proses wawancara secara langsung kepada pihak PT XYZ. Detail wawancara dapat dilihat pada halaman lampiran hasil wawancara.
24

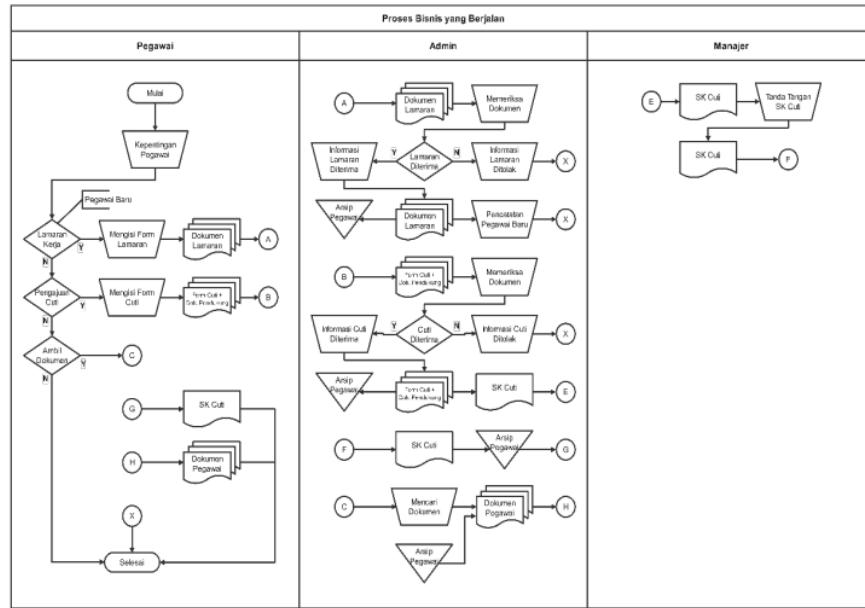
4.1.1.1. Proses Bisnis Yang Sedang Berjalan

Tahap *requirements planning* dalam mengembangkan sistem manajemen pegawai berbasis *mobile*, dimulai dengan mempelajari proses bisnis yang saat ini sedang berjalan.² Dengan mempelajari proses bisnis tersebut, pengembang dapat menilai bagian mana saja yang dapat dipermudah dan digantikan dengan suatu sistem. Sehingga proses bisnis menjadi lebih cepat, efektif, dan efisien.

Proses bisnis yang sedang berjalan diawali ketika pegawai memiliki kepentingan tertentu. Jika pegawai ingin mengambil dokumen kepegawaianya, pegawai masih harus menemui admin di bagian sumber daya manusia. Proses pencarian dokumen pegawai oleh admin masih dilakukan secara manual sehingga memerlukan proses yang cukup lama. Setelah dokumen ditemukan maka dokumen akan diberikan ke pegawai. Jika pegawai ingin mengajukan cuti, pegawai masih harus menemui admin di bagian sumber daya manusia untuk mengisi formulir pengajuan cuti dan formulir tersebut akan diberikan kembali ke admin untuk diperiksa apakah cuti disetujui atau tidak. Apabila cuti disetujui, admin akan menyerahkan SK cuti kepada manajer untuk di tandatangani. Setelah itu admin akan menyediakan dan menyimpan dokumen cuti, lalu dokumen cuti tersebut diberikan kembali ke pegawai. Apabila cuti tidak disetujui maka pegawai akan

46

mendapat pemberitahuan. Berikut ini *mapping chart* proses bisnis yang sedang berjalan.



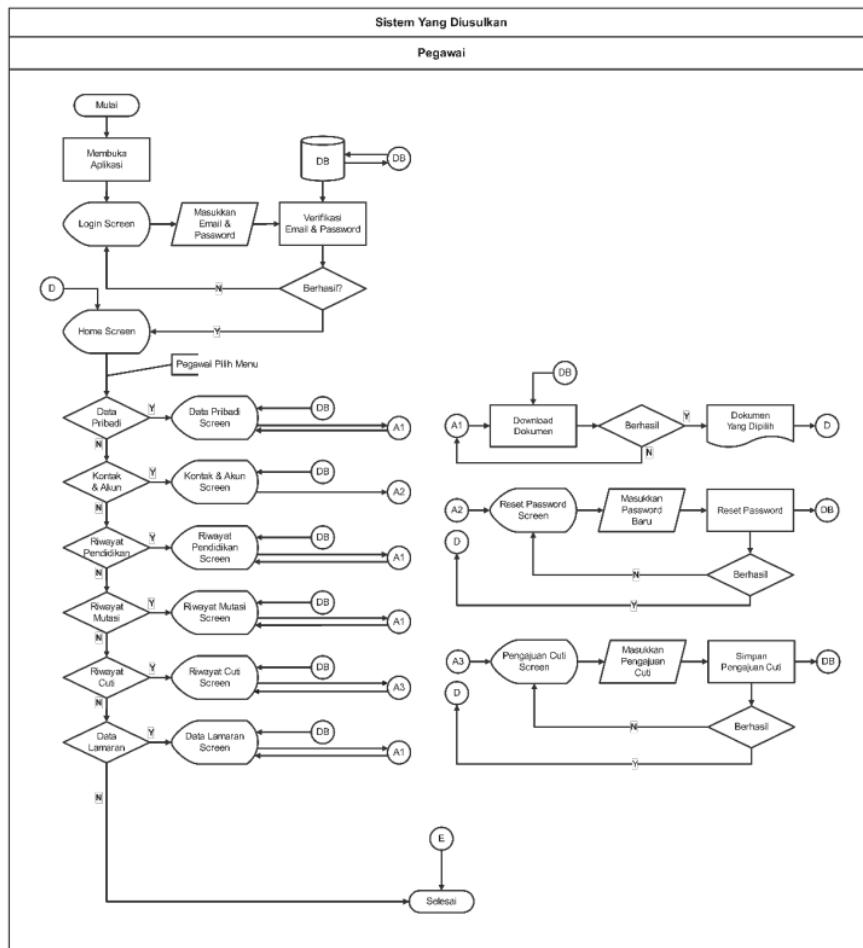
Gambar 3. *Mapping chart* proses bisnis yang sedang berjalan

24

4.1.1.2. Proses Bisnis Yang Diusulkan

Proses bisnis yang baru menggunakan sebuah sistem manajemen pegawai berbasis *mobile* sehingga proses bisnis menjadi lebih cepat, efektif, efisien, dan dapat dilakukan dimana saja dan kapan saja. Proses bisnis yang diusulkan memuat semua kebutuhan pegawai mulai dari *login*, data pribadi, kontak & akun, riwayat pendidikan, riwayat mutasi, riwayat cuti, pengajuan cuti, *reset password*, dan data lamaran. Semua dokumen yang berkaitan dengan kepegawaian dapat diunduh oleh pegawai berdasarkan akun yang telah mereka miliki. Selain itu, untuk proses pengajuan cuti pegawai mampu menambahkan pengajuan cuti secara mandiri melalui sistem. Ketika pengajuan cuti ditambahkan, maka pengajuan tersebut akan diperiksa oleh admin melalui sistem web. Apabila pengajuan cuti diterima, admin akan mengunggah dokumen yang berkaitan dengan pengajuan cuti tersebut lalu dokumen dan riwayat cuti akan ditampilkan di sisi pegawai. Apabila pengajuan cuti tidak diterima, pengajuan cuti tersebut akan dihapus oleh sistem secara otomatis. Berikut ini *mapping chart* proses bisnis yang diusulkan.

10



Gambar 4. *Mapping chart* proses bisnis yang diusulkan

4.1.2. User Design

Prototipe akan dikembangkan berdasarkan analisa pada proses bisnis yang sedang berjalan. Prototipe dikembangkan mulai dari UML sampai rancangan desain UI/UX.

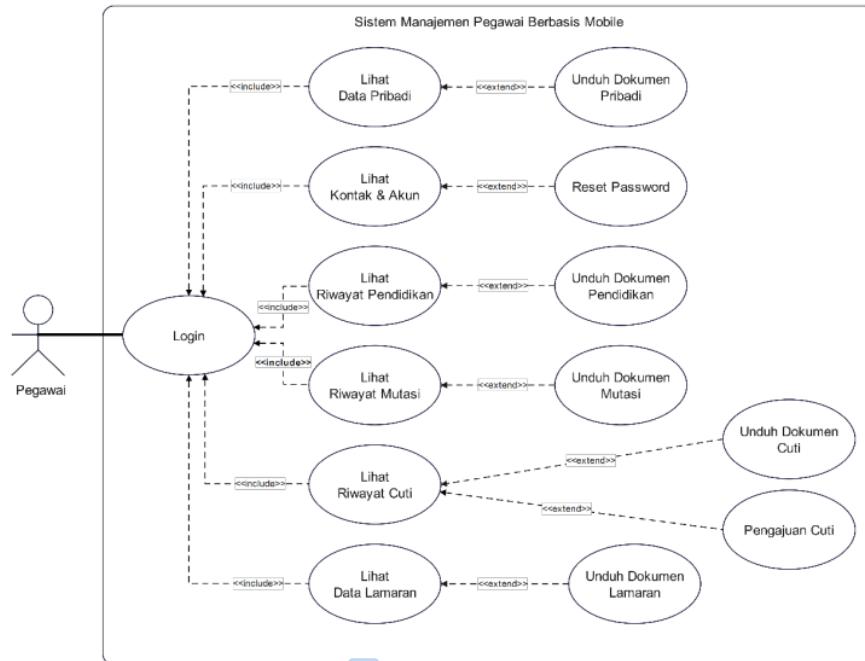
4.1.2.1. Use Case Diagram

Perancangan diagram *use case* dimaksudkan untuk memberikan gambaran hubungan antara satu atau lebih aktor dengan sistem yang dibangun. Sehingga pengembang dapat memahami fungsi yang ada di dalam sistem dan memahami aktor mana saja yang dapat mengakses fungsi tersebut. Dalam pengembangan

55

41

sistem manajemen pegawai berbasis *mobile* ini hanya terdapat aktor pegawai yang berinteraksi dengan sistem. Berikut ini *use case diagram* yang dibuat.



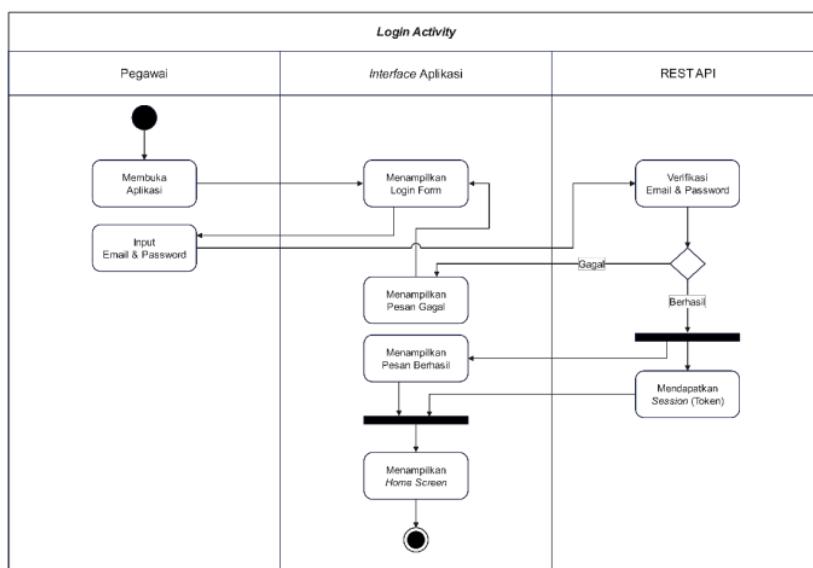
Gambar 5. Rancangan *use case diagram*

Dalam rancangan *use case diagram* pada Gambar 5, hanya aktor pegawai yang dapat berinteraksi dengan sistem. Semua jabatan pegawai dapat berinteraksi dengan sistem mulai dari pegawai biasa sampai dengan manajer. Pegawai dapat melakukan semua fungsi yang ada di dalam sistem manajemen pegawai berbasis *mobile*. Pegawai dapat melihat data pribadi, kontak & akun, riwayat pendidikan, riwayat mutasi, riwayat cuti, dan data lamaran. Selain itu, pegawai juga dapat melakukan *reset password* pada akun mereka masing-masing dan dapat mengajukan permohonan cuti, serta dapat mengunduh semua dokumen kepegawaian yang berkaitan. Semua fungsi tersebut dapat diakses pegawai jika pegawai berhasil melakukan *login* ke dalam aplikasi menggunakan akun mereka sendiri.

22 4.1.2.2. Activity Diagram

Rancangan *activity diagram* dimaksudkan untuk menggambarkan kegiatan dan alir kerja sistem untuk mempermudah pengembang dalam mengembangkan fungsi-fungsi yang ada di dalam sistem manajemen pegawai berbasis *mobile* sesuai dengan alir kerja sistem yang diinginkan. Pada rancangan *activity diagram* ini dibuat dalam tiga *swimlane* yaitu pegawai, *interface* aplikasi, dan REST API.
76
16 Berikut ini beberapa *activity diagram* berdasarkan fungsi-fungsi yang ada.

1. Activity diagram login



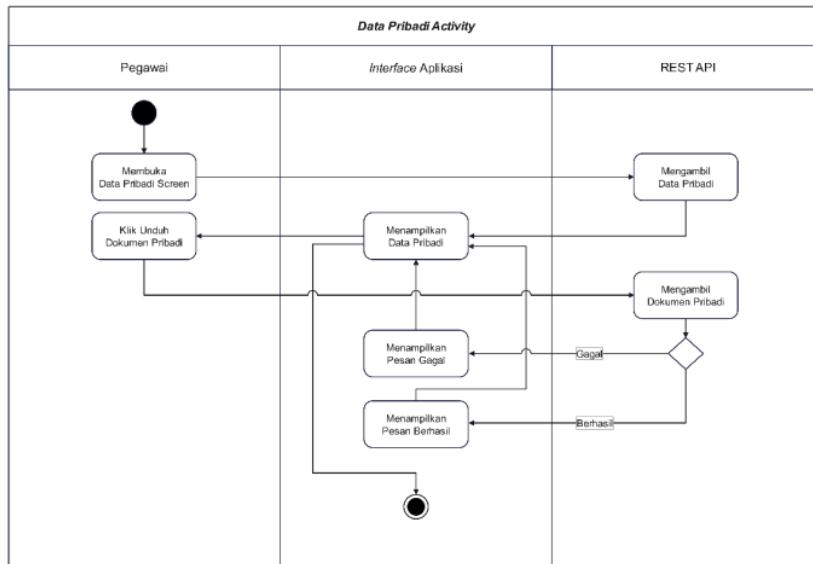
Gambar 6. *Activity diagram login*

Pada *activity diagram* ini, proses *login* dimulai ketika pegawai membuka aplikasi dan dihadapkan dengan tampilan *login form*. Pegawai diminta mengisi *login form* dengan *email* dan *password* akun mereka masing-masing. Setelah itu, data yang berisi *email* dan *password* tersebut dikirim menuju REST API untuk diverifikasi. Apabila *email* atau *password* yang dimasukkan pegawai gagal diverifikasi, maka REST API akan mengirimkan respons gagal ke *interface* aplikasi untuk ditampilkan ke hadapan pegawai dan pegawai akan diminta memasukkan kembali data *login* yang benar. Apabila *email* dan *password* berhasil diverifikasi, maka REST API akan mengirimkan respons berhasil bersamaan dengan *session* ke *interface* aplikasi. Dengan *session*

tersebut *interface* aplikasi akan mengalihkan pegawai menuju ke halaman *home* dan proses *login* pun berhasil.

18

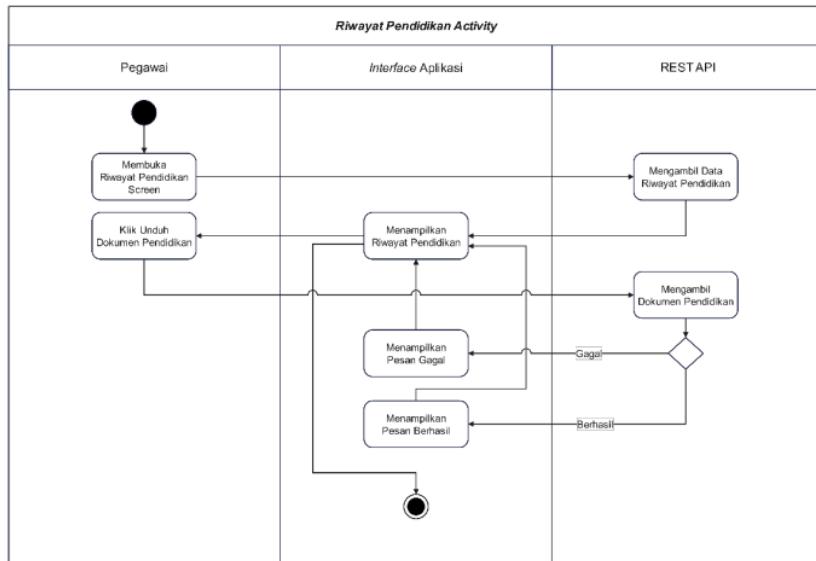
2. Activity diagram data pribadi



Gambar 7. Activity diagram data pribadi

Pada *activity diagram* ini, proses akan dimulai ketika pegawai memuka halaman data pribadi. Sebelum data ditampilkan pada *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data yang dibutuhkan. Data yang telah disediakan REST API selanjutnya akan diteruskan dan ditampilkan *interface* aplikasi ke hadapan pegawai. Jika pegawai ingin mengunduh dokumen pribadi dengan menekan *button* unduh, maka permintaan tersebut diteruskan ke REST API. Berdasarkan permintaan pegawai tersebut REST API akan menyediakan dokumen pribadi. Jika dokumen yang diminta gagal disiapkan, maka respons gagal akan diteruskan ke *interface* aplikasi untuk ditampilkan ke hadapan pegawai. Jika dokumen yang diminta berhasil disiapkan, maka respons berhasil akan diteruskan ke *interface* aplikasi bersamaan dengan dokumen yang diminta sehingga pegawai dapat menyimpan dokumen tersebut lalu proses pun berakhir.

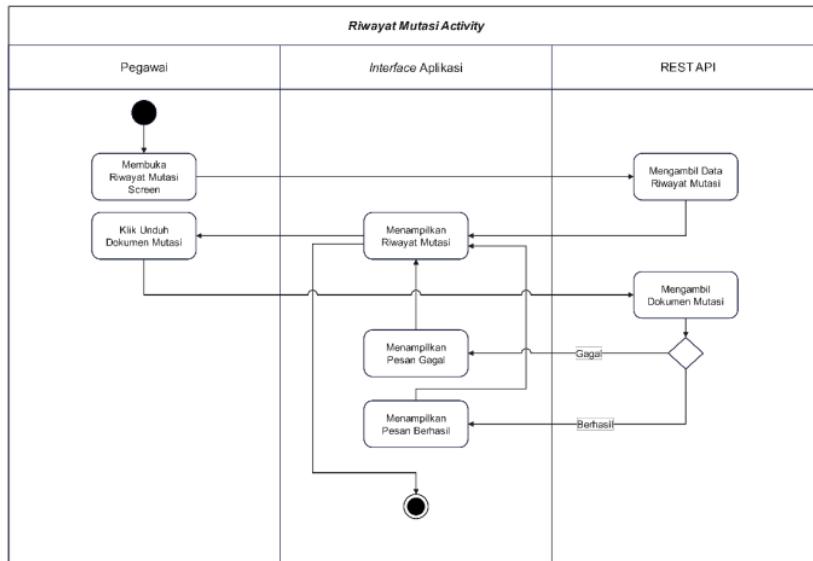
3. Activity diagram riwayat pendidikan



Gambar 8. Activity diagram riwayat pendidikan

Pada *activity* diagram ini, proses akan dimulai ketika pegawai memuka halaman riwayat pendidikan. Sebelum data ditampilkan pada *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data yang dibutuhkan. Data yang telah disediakan REST API selanjutnya akan diteruskan dan ditampilkan *interface* aplikasi ke hadapan pegawai. Jika pegawai ingin mengunduh dokumen pendidikan dengan menekan *button* unduh, maka permintaan tersebut diteruskan ke REST API. Berdasarkan permintaan pegawai tersebut REST API akan menyediakan dokumen pendidikan. Jika dokumen yang diminta gagal disiapkan, maka respons gagal akan diteruskan ke *interface* aplikasi untuk ditampilkan ke hadapan pegawai. Jika dokumen yang diminta berhasil disiapkan, maka respons berhasil akan diteruskan ke *interface* aplikasi bersamaan dengan dokumen yang diminta sehingga pegawai dapat menyimpan dokumen tersebut lalu proses pun berakhir.

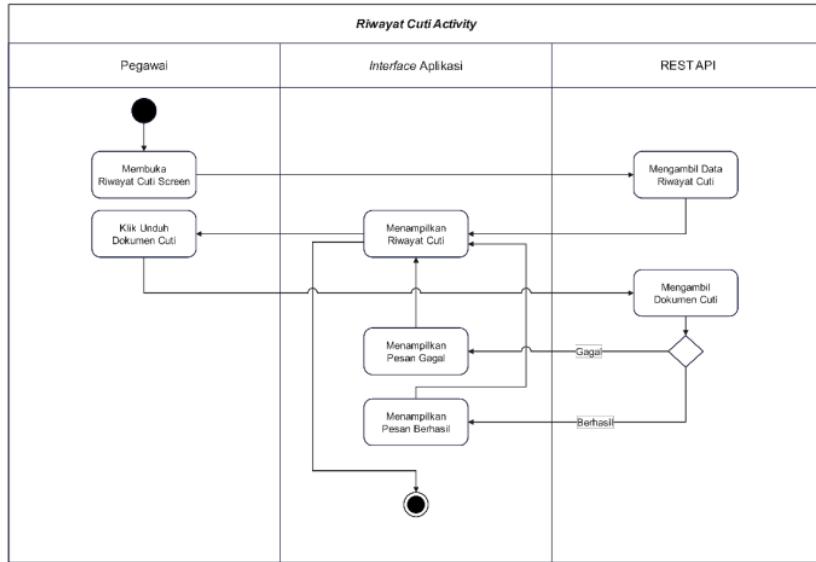
4. Activity diagram riwayat mutasi



Gambar 9. Activity diagram riwayat mutasi

Pada *activity* diagram ini, proses akan dimulai ketika pegawai memuka halaman riwayat mutasi. Sebelum data ditampilkan pada *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data yang dibutuhkan. Data yang telah disediakan REST API selanjutnya akan diteruskan dan ditampilkan *interface* aplikasi ke hadapan pegawai. Jika pegawai ingin mengunduh dokumen mutasi dengan menekan *button* unduh, maka permintaan tersebut diteruskan ke REST API. Berdasarkan permintaan pegawai tersebut REST API akan menyediakan dokumen mutasi. Jika dokumen yang diminta gagal disiapkan, maka respons gagal akan diteruskan ke *interface* aplikasi untuk ditampilkan ke hadapan pegawai. Jika dokumen yang diminta berhasil disiapkan, maka respons berhasil akan diteruskan ke *interface* aplikasi bersamaan dengan dokumen yang diminta sehingga pegawai dapat menyimpan dokumen tersebut lalu proses pun berakhir.

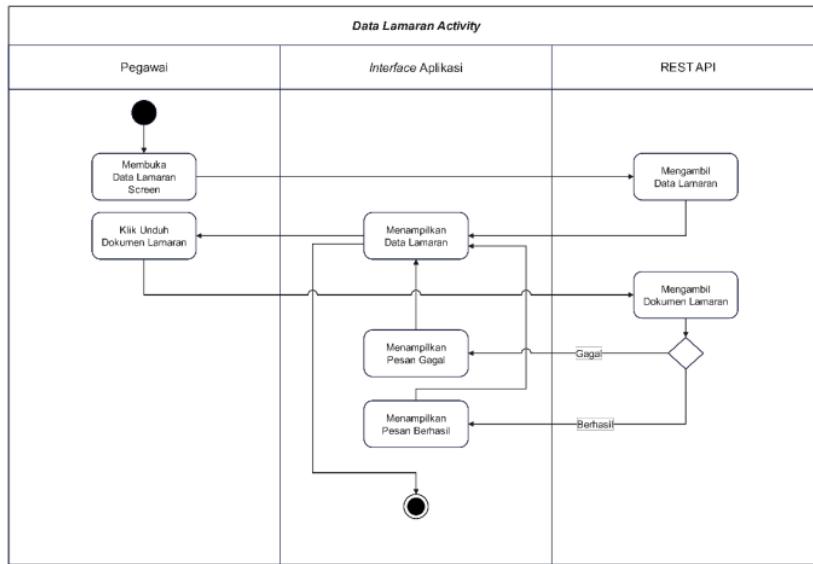
5. Activity diagram riwayat cuti



Gambar 10. Activity diagram riwayat cuti

Pada *activity* diagram ini, proses akan dimulai ketika pegawai memuka halaman riwayat cuti. Sebelum data ditampilkan pada *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data yang dibutuhkan. Data yang telah disediakan REST API selanjutnya akan diteruskan dan ditampilkan *interface* aplikasi ke hadapan pegawai. Jika pegawai ingin mengunduh dokumen cuti dengan menekan *button* unduh, maka permintaan tersebut diteruskan ke REST API. Berdasarkan permintaan pegawai tersebut REST API akan menyediakan dokumen cuti. Jika dokumen yang diminta gagal disiapkan, maka respons gagal akan diteruskan ke *interface* aplikasi untuk ditampilkan ke hadapan pegawai. Jika dokumen yang diminta berhasil disiapkan, maka respons berhasil akan diteruskan ke *interface* aplikasi bersamaan dengan dokumen yang diminta sehingga pegawai dapat menyimpan dokumen tersebut lalu proses pun berakhir.

18
6. Activity diagram data lamaran

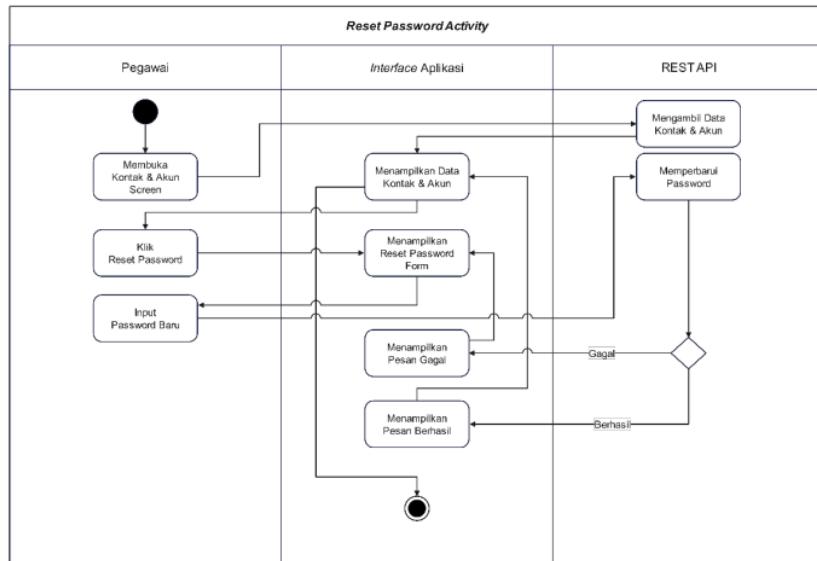


Gambar 11. Activity diagram data lamaran

Pada *activity diagram* ini, proses akan dimulai ketika pegawai memuka halaman data lamaran. Sebelum data ditampilkan pada *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data yang dibutuhkan. Data yang telah disediakan REST API selanjutnya akan diteruskan dan ditampilkan *interface* aplikasi ke hadapan pegawai. Jika pegawai ingin mengunduh dokumen lamaran dengan menekan *button* unduh, maka permintaan tersebut diteruskan ke REST API. Berdasarkan permintaan pegawai tersebut REST API akan menyediakan dokumen lamaran. Jika dokumen yang diminta gagal disiapkan, maka respons gagal akan diteruskan ke *interface* aplikasi untuk ditampilkan ke hadapan pegawai. Jika dokumen yang diminta berhasil disiapkan, maka respons berhasil akan diteruskan ke *interface* aplikasi bersamaan dengan dokumen yang diminta sehingga pegawai dapat menyimpan dokumen tersebut lalu proses pun berakhir.

16

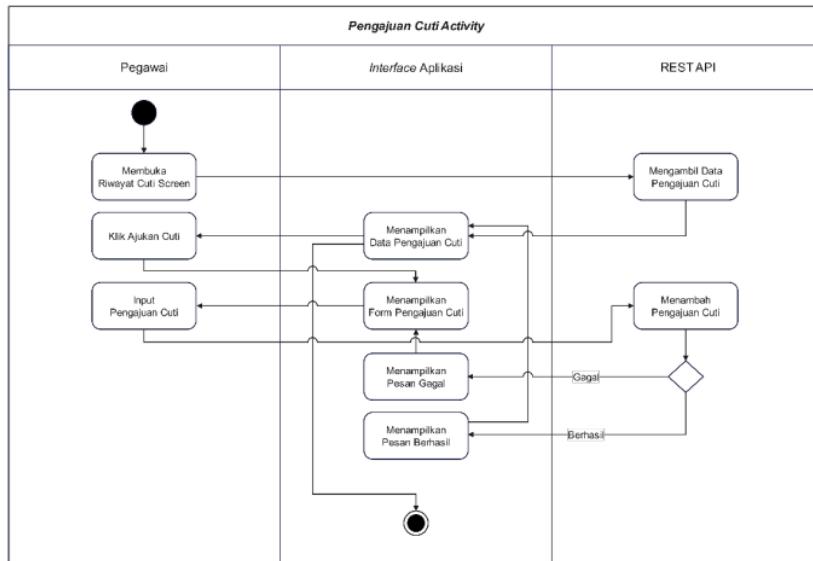
7. Activity diagram reset password



Gambar 12. Activity diagram reset password

Pada *activity diagram* ini, proses akan dimulai ketika pegawai memuka halaman kontak & akun. Sebelum data ditampilkan pada *interface aplikasi*, REST API mendapat permintaan untuk menyediakan data yang dibutuhkan. Data yang telah disediakan REST API selanjutnya akan diteruskan dan ditampilkan *interface aplikasi* ke hadapan pegawai. Jika ingin melakukan *reset password* dengan menekan *button reset password*, maka *interface aplikasi* akan menampilkan *reset password form* dan pegawai diminta untuk memasukkan *password* yang baru. Data *password* baru tersebut diteruskan ke REST API untuk diproses. Jika *password* baru gagal diproses, maka REST API akan meneruskan respons gagal ke *interface aplikasi* untuk ditampilkan ke hadapan pegawai dan pegawai akan diminta memasukkan kembali data *password* baru. Jika *password* baru tersebut berhasil diproses, maka REST API akan meneruskan respons berhasil ke *interface aplikasi* untuk ditampilkan ke hadapan pegawai dan proses pun berakhir.

42
8. Activity diagram pengajuan cuti



Gambar 13. Activity diagram pengajuan cuti

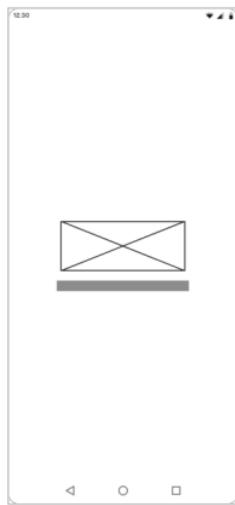
Pada *activity diagram* ini, proses akan dimulai ketika pegawai memuka halaman riwayat cuti. Sebelum data ditampilkan pada *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data yang dibutuhkan. Data yang telah disediakan REST API selanjutnya akan diteruskan dan ditampilkan *interface* aplikasi ke hadapan pegawai. Jika ingin melakukan pengajuan cuti baru dengan menekan *button* ajukan cuti, maka *interface* aplikasi akan menampilkan pengajuan cuti *form* dan pegawai diminta untuk memasukkan data pengajuan cuti baru. Data pengajuan cuti baru tersebut diteruskan ke REST API untuk diproses. Jika data pengajuan cuti baru gagal diproses, maka REST API akan meneruskan respons gagal ke *interface* aplikasi untuk ditampilkan ke hadapan pegawai dan pegawai akan diminta memasukkan kembali data pengajuan cuti baru. Jika data pengajuan cuti baru berhasil diproses, maka REST API akan meneruskan respons berhasil ke *interface* aplikasi untuk ditampilkan ke hadapan pegawai dan proses pun berakhir.

4.1.2.3. Desain Tampilan Aplikasi

Wireframe aplikasi didesain dengan mementingkan elemen-elemen dasar seperti *card*, *button*, *text*, *textfield*, dsb. tanpa mementingkan pewarnaan dan gaya rumit lainnya. Berikut ini desain *wireframe* aplikasi.

1. Wireframe interface splash screen

Halaman ini ditampilkan pertama kali saat aplikasi baru dibuka dan sebelum menampilkan halaman *login* atau halaman *home*. Pada halaman ini ditampilkan *brand logo* sebagai identitas aplikasi. Berikut ini tampilan *wireframe interface splash screen*.



Gambar 14. Wireframe interface splash screen

2. Wireframe interface login screen

Halaman ini ditampilkan setelah halaman *splash* dan ditampilkan apabila pegawai belum memiliki *login session*. Pada halaman ini terdapat *brand logo*, *textfield email* dan *password*, serta *button login*. Berikut ini tampilan *wireframe interface login screen*.



Gambar 15. Wireframe interface login screen

3. Wireframe interface home screen

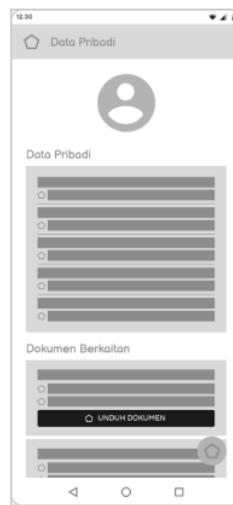
Halaman ini ditampilkan setelah halaman *splash* apabila pegawai memiliki *login session* dan ditampilkan setelah halaman *login* apabila pegawai belum memiliki *login session*. Pada halaman ini terdapat beberapa komponen utama seperti *card* profil, *list* menu, *card* mutasi terbaru, *card* cuti terbaru, *card* dokumen terbaru, dan *fab*. Berikut ini tampilan *wireframe interface home screen*.



Gambar 16. Wireframe interface home screen

4. Wireframe interface data pribadi screen

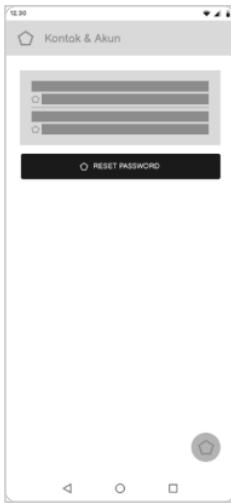
Halaman ini ditampilkan apabila pegawai menekan menu data pribadi pada komponen *list* menu yang ada di halaman *home*. Pada halaman terdapat beberapa komponen yaitu foto profil, *card* detail data pribadi, dan *list* dokumen data pribadi. Berikut ini tampilan *wireframe interface* data pribadi *screen*.



Gambar 17. *Wireframe interface* data pribadi *screen*

5. Wireframe interface kontak & akun screen

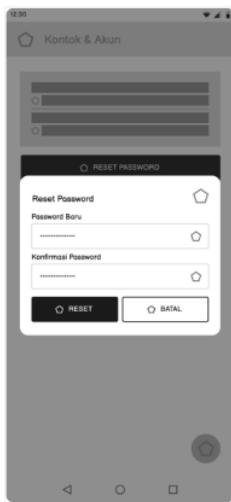
Halaman ini ditampilkan apabila pegawai menekan menu kontak & akun pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat beberapa komponen yaitu *card* detail kontak & akun, dan *button reset password*. Berikut ini tampilan *wireframe interface* kontak & akun *screen*.



Gambar 18. *Wireframe interface kontak & akun screen*

6. *Wireframe interface reset password screen*

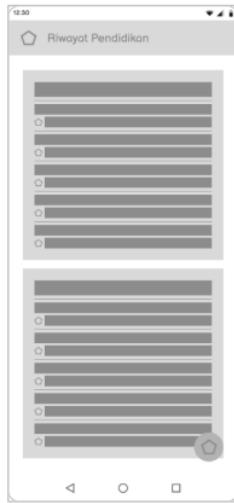
Halaman ini ditampilkan apabila pegawai menekan *button reset password* yang ada di halaman kontak & akun. Pada halaman ini komponen akan ditampilkan di atas *modal card* yang berisi *textfield password baru* dan *konfirmasi password*, serta *button reset* dan batal. Berikut ini tampilan *wireframe interface reset password screen*.



Gambar 19. *Wireframe interface reset password screen*

7. *Wireframe interface riwayat pendidikan screen*

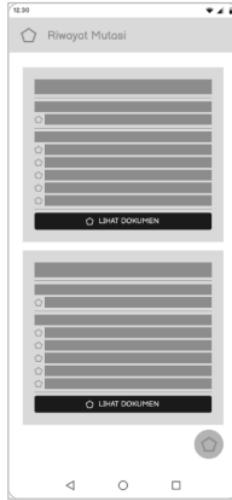
Halaman ini ditampilkan apabila pegawai menekan menu riwayat pendidikan pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat komponen *card* detail riwayat pendidikan. Berikut ini tampilan *wireframe interface* riwayat pendidikan *screen*.



Gambar 20. *Wireframe interface* riwayat pendidikan *screen*

8. *Wireframe interface riwayat mutasi screen*

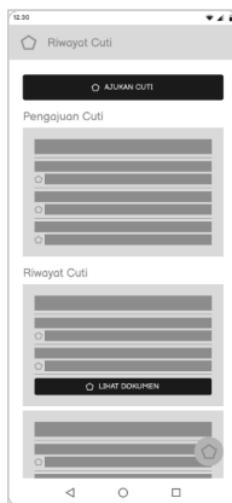
Halaman ini ditampilkan apabila pegawai menekan menu riwayat mutasi pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat komponen *card* detail riwayat mutasi. Berikut ini tampilan *wireframe interface* riwayat mutasi *screen*.



Gambar 21. Wireframe interface riwayat mutasi screen

9. Wireframe interface riwayat cuti screen

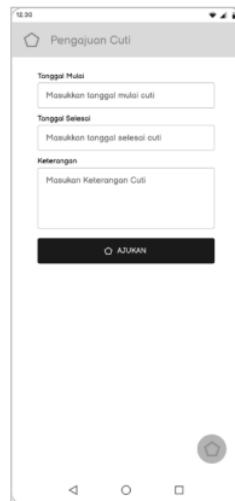
Halaman ini ditampilkan apabila pegawai menekan menu riwayat cuti pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat beberapa komponen yaitu *button* ajukan cuti, *card* detail pengajuan cuti, *card*, dan detail riwayat cuti. Berikut ini tampilan *wireframe interface* riwayat cuti screen.



Gambar 22. Wireframe interface riwayat cuti screen

10. Wireframe interface pengajuan cuti screen

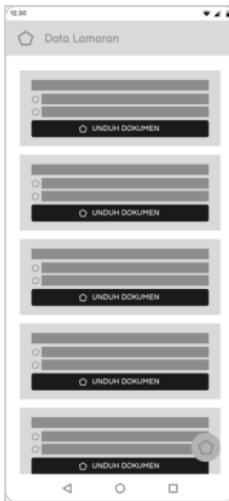
Halaman ini ditampilkan apabila pegawai menekan *button* ajukan cuti yang ada di halaman riwayat cuti. Pada halaman ini terdapat beberapa komponen yaitu *datefield* tanggal mulai dan tanggal selesai, *textarea* keterangan, dan *button* ajukan. Berikut ini tampilan *wireframe interface* pengajuan cuti screen.



Gambar 23. *Wireframe interface* pengajuan cuti screen

11. Wireframe interface data lamaran screen

Halaman ini ditampilkan apabila pegawai menekan menu data lamaran pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat komponen *card* detail data lamaran. Berikut ini tampilan *wireframe interface* data lamaran screen.



Gambar 24. *Wireframe interface* data lamaran screen

4.1.3. *Construction*

Pada fase ini, pengembang mulai mengembangkan aplikasi. Proses pengembangan tersebut dilakukan dengan penulisan kode (*coding*) dan pengujian fitur yang baru saja dikembangkan. Dalam fase ini, *error* atau *bug* sering terjadi. Oleh karena itu, proses ini memakan waktu yang cukup lama dan memerlukan tingkat ketelitian yang tinggi. Pengembangan aplikasi menggunakan beberapa *library* yaitu *react native*, *nativebase* dan beberapa *library* pendukung lainnya. Dokumentasi kode dapat dilihat pada halaman lampiran kode program.

Proses penulisan kode diawali dengan menghubungkan aplikasi dengan *back-end* sehingga fungsi-fungsi yang disediakan dapat dikonsumsi oleh aplikasi yang dikembangkan. Penulisan kode dapat dilihat pada lampiran kode program nomor 1. Proses penulisan kode selanjutnya dilakukan berdasarkan halaman yang sudah di desain sebelumnya. Berikut ini hasil pengembangan aplikasi pada fase *construction* berdasarkan tampilan halaman.

1. Halaman *splash*

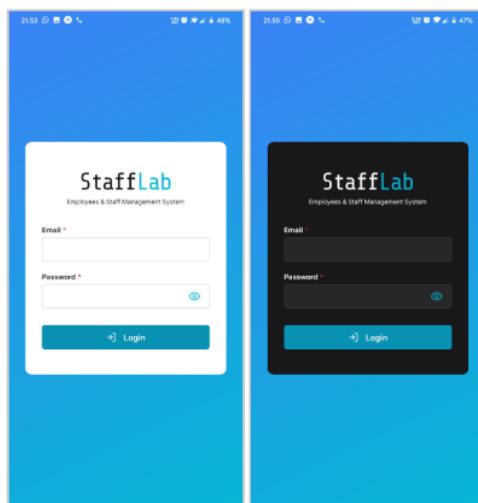
Halaman ini ditampilkan pertama kali saat aplikasi baru dibuka dan sebelum menampilkan halaman *login* atau halaman *home*. Pada halaman ini ditampilkan *brand logo* sebagai identitas aplikasi. Berikut ini tampilan halaman *splash*.



Gambar 25. Halaman *splash*

2. Halaman *login*

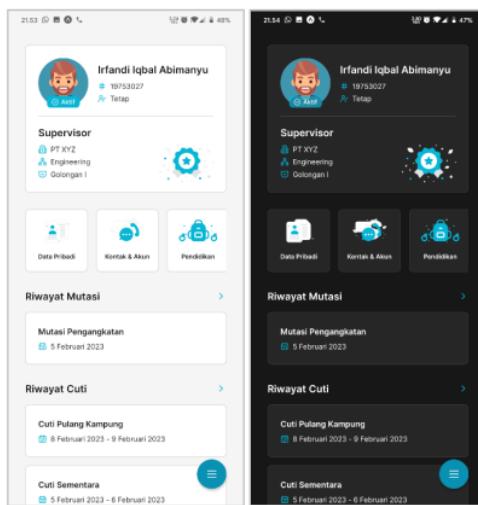
Halaman ini ditampilkan setelah halaman *splash* dan ditampilkan apabila pegawai belum memiliki *login session*. Pada halaman ini terdapat *brand logo*, *textfield email* dan *password*, serta *button login*. Dokumentasi kode dapat dilihat pada halaman lampiran kode program nomor 2. Berikut ini tampilan halaman *login* dalam dua tema aplikasi.



Gambar 26. Halaman *login*

3. Halaman *home*

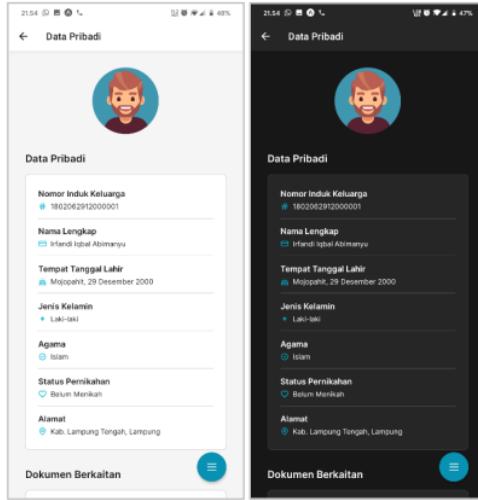
Halaman ini ditampilkan setelah halaman *splash* apabila pegawai memiliki *login session* dan ditampilkan setelah halaman *login* apabila pegawai belum memiliki *login session*. Pada halaman ini terdapat beberapa komponen utama seperti *card* profil, *list* menu, *card* mutasi terbaru, *card* cuti terbaru, *card* dokumen terbaru, dan *fab*. Dokumentasi kode dapat dilihat pada halaman lampiran kode program nomor 3. Berikut ini tampilan halaman *home* dalam dua tema aplikasi.



Gambar 27. Halaman *home*

4. Halaman data pribadi

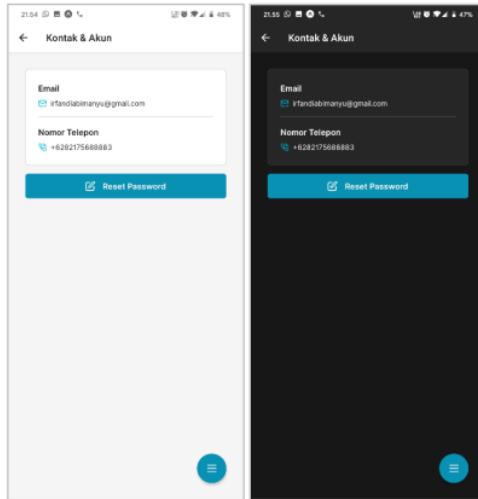
Halaman ini ditampilkan apabila pegawai menekan menu data pribadi pada komponen *list* menu yang ada di halaman *home*. Pada halaman terdapat beberapa komponen yaitu foto profil, *card* detail data pribadi, dan *list* dokumen data pribadi. Dokumentasi kode dapat dilihat pada halaman lampiran kode program nomor 4. Berikut ini tampilan halaman data pribadi dalam dua tema aplikasi.



Gambar 28. Halaman data pribadi

5. Halaman kontak & akun

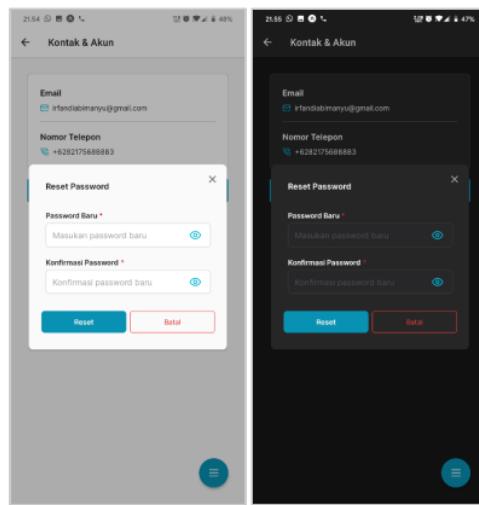
Halaman ini ditampilkan apabila pegawai menekan menu kontak & akun pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat beberapa komponen yaitu *card* detail kontak & akun, dan *button* *reset password*. Dokumentasi kode dapat dilihat pada halaman lampiran kode program nomor 5. Berikut ini tampilan halaman kontak & akun dalam dua tema aplikasi.



Gambar 29. Halaman kontak & akun

6. Halaman *reset password*

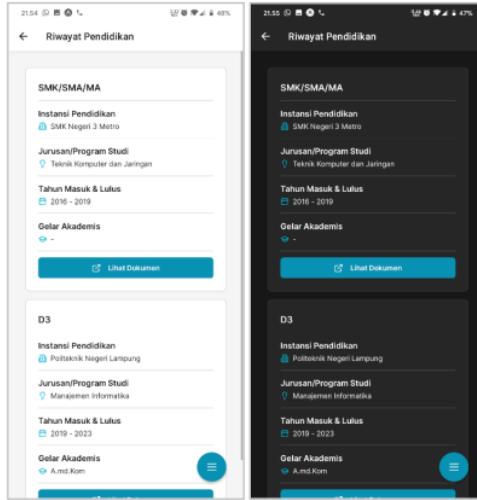
Halaman ini ditampilkan apabila pegawai menekan *button reset password* yang ada di halaman kontak & akun. Pada halaman ini komponen akan ditampilkan di atas *modal card* yang berisi *textfield password* baru dan konfirmasi *password*, serta *button reset* dan batal. Dokumentasi kode dapat dilihat pada halaman lampiran kode program nomor 6. Berikut ini tampilan halaman *reset password* dalam dua tema aplikasi.



Gambar 30. Halaman *reset password*

7. Halaman riwayat pendidikan

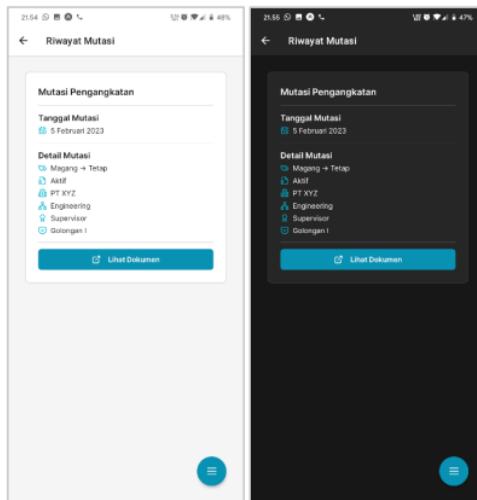
Halaman ini ditampilkan apabila pegawai menekan menu riwayat pendidikan pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat komponen *card* detail riwayat pendidikan. Dokumentasi kode dapat dilihat pada halaman lampiran kode program nomor 7. Berikut ini tampilan halaman riwayat pendidikan dalam dua tema aplikasi.



Gambar 31. Halaman riwayat pendidikan

8. Halaman riwayat mutasi

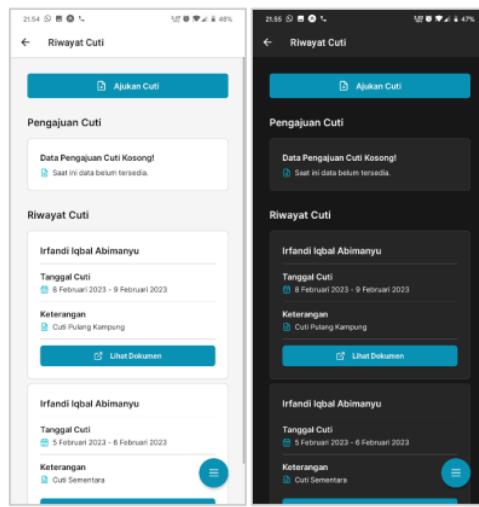
Halaman ini ditampilkan apabila pegawai menekan menu riwayat mutasi pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat komponen *card* detail riwayat mutasi. Dokumentasi kode dapat dilihat pada halaman lampiran kode program nomor 8. Berikut ini tampilan halaman riwayat mutasi dalam dua tema aplikasi.



Gambar 32. Halaman riwayat mutasi

9. Halaman riwayat cuti

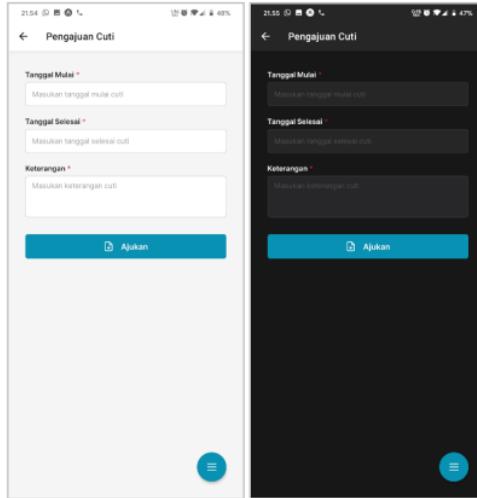
Halaman ini ditampilkan apabila pegawai menekan menu riwayat cuti pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat beberapa komponen yaitu *button* ajukan cuti, *card* detail pengajuan cuti, *card*, dan detail riwayat cuti. Dokumentasi kode dapat dilihat pada halaman lampiran kode program nomor 9. Berikut ini tampilan halaman riwayat cuti dalam dua tema aplikasi.



Gambar 33. Halaman riwayat cuti

10. Halaman pengajuan cuti

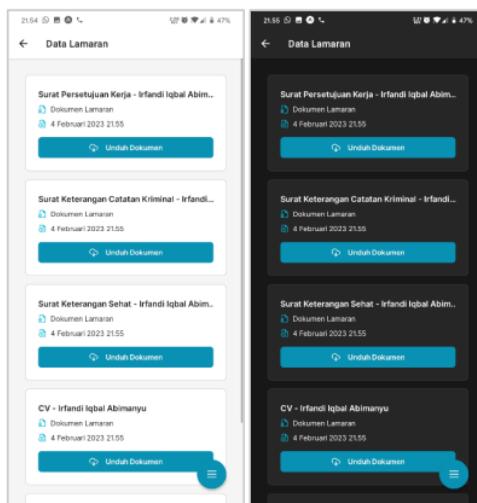
Halaman ini ditampilkan apabila pegawai menekan *button* ajukan cuti yang ada di halaman riwayat cuti. Pada halaman ini terdapat beberapa komponen yaitu *datefield* tanggal mulai dan tanggal selesai, *textarea* keterangan, dan *button* ajukan. Dokumentasi kode dapat dilihat pada halaman lampiran kode program nomor 10. Berikut ini tampilan halaman pengajuan cuti dalam dua tema aplikasi.



Gambar 34. Halaman pengajuan cuti

11. Halaman data lamaran

Halaman ini ditampilkan apabila pegawai menekan menu data lamaran pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat komponen *card* detail data lamaran. Dokumentasi kode dapat dilihat pada halaman lampiran kode program nomor 11. Berikut ini tampilan halaman data lamaran dalam dua tema aplikasi.



Gambar 35. Halaman pengajuan cuti

4.1.4. Cutover

Setelah pengembangan sistem manajemen pegawai berbasis *mobile* melalui fase *construction* dan berhasil melalui pengetesan fitur, selanjutnya sistem akan melalui fase *cutover*. Pada tahap ini ulasan dan *feedback* pengguna dikumpulkan dan dipertimbangkan apakah sistem layak digunakan atau perlu penambahan fitur baru. Pengumpulan ulasan dilakukan menggunakan metode SUS dengan jumlah responen 23 orang dengan 10 pernyataan yang bernilai ³⁰ 1-5 yaitu sangat tidak setuju, tidak setuju, ragu-ragu, setuju, dan sangat setuju.

4.1.4.1. Analisis Aspek Ketertarikan

Berdasarkan hasil analisis aspek ketertarikan, terdapat 15 responen dengan persentase 65,22% menyatakan sangat setuju dan 8 responen lainnya dengan persentase 34,78% menyatakan setuju. Hal ini mengindikasikan bahwa responden tertarik dalam menggunakan aplikasi ini. Berikut ini grafik persentase analisis aspek ketertarikan.



Gambar 36. Grafik persentase analisis aspek ketertarikan

4.1.4.2. Analisis Kompleksitas Aplikasi

Berdasarkan analisis kompleksitas aplikasi, terdapat 13 responen dengan persentase 56,52% menyatakan tidak setuju dan 10 responen lainnya dengan persentase 43,48% menyatakan sangat tidak setuju. Hal ini mengindikasikan aplikasi ini tidak terlalu rumit untuk digunakan. Berikut ini grafik persentase analisis kompleksitas aplikasi.



Gambar 37. Grafik persentase analisis kompleksitas aplikasi

4.1.4.3. Analisis Aspek Efektivitas Aplikasi

Berdasarkan analisis aspek efektivitas aplikasi, terdapat 13 responden dengan persentase 56,52% menyatakan sangat setuju dan 10 responden lainnya dengan persentase 43,48% menyatakan setuju. Hal ini mengindikasikan aplikasi yang ini sudah efektif dan mudah digunakan. Berikut ini grafik persentase analisis aspek efektivitas aplikasi.



Gambar 38. Grafik persentase analisis aspek efektivitas aplikasi

4.1.4.4. Analisis Aspek Teknis Aplikasi

Berdasarkan analisis aspek teknis aplikasi, terdapat 13 responden dengan persentase 56,52% memilih tidak setuju, 6 responden dengan persentase 26,09% menyatakan sangat tidak setuju, dan 4 responden lainnya dengan persentase 17,39% menyatakan ragu-ragu, Dengan demikian sebagian responden tidak memerlukan

bantuan teknisi atau orang lain ketika menjalankan aplikasi ini. Berikut ini grafik persentase analisis aspek teknik aplikasi.



Gambar 39. Grafik persentase analisis aspek teknik aplikasi

4.1.4.5. Analisis Aspek Fitur-fitur Aplikasi

Berdasarkan analisis aspek fitur-fitur aplikasi, terdapat 13 responden dengan persentase 56,52% menyatakan sangat setuju dan 10 responden lainnya dengan persentase 43,48% menyatakan setuju. Hal ini mengindikasikan bahwa fitur-fitur yang tersedia dalam aplikasi ini berjalan semestinya. Berikut ini grafik analisis aspek fitur-fitur aplikasi.



Gambar 40. Grafik analisis aspek fitur-fitur aplikasi

4.1.4.6. Analisis Aspek Penggunaan Keseharian Aplikasi

Berdasarkan analisis aspek penggunaan keseharian aplikasi, terdapat 12 responden dengan persentase 52,17% menyatakan tidak setuju dan 11 responden

lainnya dengan persentase 47,83% menyatakan sangat tidak setuju. Hal ini mengindikasikan aplikasi ini sudah serasi dan konsisten ketika digunakan. Berikut ini grafik analisis aspek keseharian aplikasi.



Gambar 41. Grafik analisis aspek keseharian aplikasi

4.1.4.7. Analisis Aspek Pemahaman Pengguna

Berdasarkan analisis aspek pemahaman pengguna, terdapat 13 responden dengan persentase 56,52% menyatakan sangat setuju dan 10 responden lainnya dengan persentase 43,48% menyatakan setuju. Hal ini mengindikasikan bahwa pengguna mudah memahami cara penggunaan aplikasi ini dengan cepat. Berikut ini grafik analisis aspek pemahaman aplikasi.



Gambar 42. Grafik analisis aspek pemahaman aplikasi

4.1.4.8. Analisis Aspek Kejelasan Sistem

Berdasarkan analisis aspek kejelasan sistem, terdapat 13 responden dengan persentase 56,52% menyatakan sangat tidak setuju dan 10 responden lainnya dengan persentase 43,48% menyatakan tidak setuju. Hal ini mengindikasikan aplikasi ini sangat jelas dan tidak membingungkan pengguna. Berikut ini grafik analisis aspek kejelasan sistem.



Gambar 43. Grafik analisis aspek kejelasan sistem

4.1.4.9. Analisis Aspek Kelancaran Aplikasi

Berdasarkan analisis aspek kelancaran aplikasi, terdapat 14 responden dengan persentase 60,87% menyatakan sangat setuju dan 9 responden lainnya dengan persentase 39,13% menyatakan setuju karena dalam menjalankan aplikasi tidak ada hambatan dalam penggunaannya. Hal ini mengindikasikan bahwa dalam penggunaannya, aplikasi ini sudah berjalan dengan lancar tanpa adanya hambatan. Berikut ini grafik analisis aspek kelancaran aplikasi.



Gambar 44. Grafik analisis aspek kelancaran aplikasi

4.1.4.10. Analisis Aspek Penguasaan Aplikasi

Berdasarkan analisis aspek penguasaan aplikasi, terdapat 10 responden dengan persentase 43,48% menyatakan sangat tidak setuju, 10 responden dengan persentase 43,48% menyatakan tidak setuju, dan 3 responden lainnya dengan persentase 13,04% menyatakan ragu-ragu. Hal ini mengindikasikan bahwa aplikasi ini mudah digunakan sehingga pengguna tidak memerlukan waktu yang lama untuk membiasakan diri. Berikut ini grafik analisis aspek penguasaan aplikasi.



Gambar 45. Grafik analisis aspek penguasaan aplikasi

4.1.4.11. Hasil Analisis Keseluruhan

Berdasarkan analisis yang dilakukan oleh 23 responden yang berpartisipasi, Langkah terakhir yang perlu dilakukan untuk mengetahui hasil analisis keseluruhan yaitu dengan melakukan perhitungan setiap nilai respons kuesioner di setiap aspek dengan aturan perhitungan metode SUS. Berikut ini aturan perhitungan skor kuesioner dalam metode SUS.

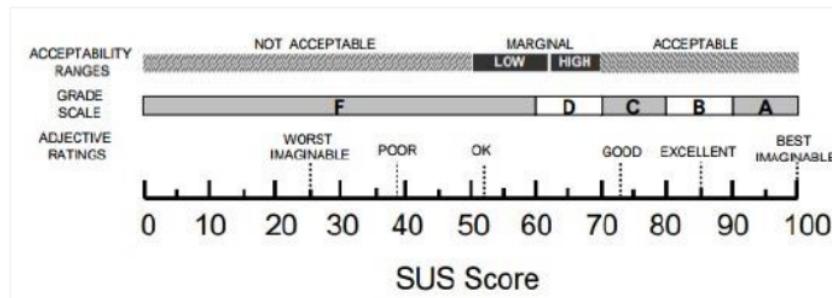
1. Skor dari respons pernyataan dengan nomor ganjil akan dikurangi 1 ($x-1$).
2. Skor dari respons pernyataan dengan nomor genap akan digunakan untuk mengurangkan 5 ($5-x$).
3. Skor yang diperoleh setiap pernyataan akan dijumlahkan seluruhnya dan akan dikalikan 2,5.
4. Skor akhir SUS didapatkan dengan cara mencari nilai rata-rata dari hasil perhitungan keseluruhan pada aturan nomor 3.

Berikut merupakan hasil perhitungan akhir analisis keseluruhan kuesioner dari 23 responden dapat dilihat pada Gambar 46.

User	P1	SUS#	P2	SUS#	P3	SUS#	P4	SUS#	P5	SUS#	P6	SUS#	P7	SUS#	P8	SUS#	P9	SUS#	P10	SUS#	Skor SUS
1	4	3	2	3	4	3	2	3	4	3	2	3	4	3	2	3	4	3	2	3	75,0
2	4	3	2	3	5	4	1	4	5	4	2	3	4	3	1	4	5	4	3	2	85,0
3	4	3	1	4	4	3	1	4	4	3	2	3	5	4	2	3	4	3	1	4	85,0
4	4	3	1	4	4	3	3	2	5	4	2	3	5	4	2	3	4	3	3	2	77,5
5	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	100,0
6	5	4	2	3	5	4	3	2	5	4	2	3	5	4	2	3	5	4	3	2	82,5
7	5	4	2	3	4	3	2	3	4	3	2	3	5	4	2	3	5	4	1	4	85,0
8	5	4	1	4	5	4	2	3	5	4	1	4	5	4	1	4	5	4	2	3	95,0
9	5	4	1	4	4	3	2	3	4	3	1	4	4	3	1	4	4	3	2	3	85,0
10	5	4	1	4	5	4	1	4	4	3	1	4	4	3	1	4	5	4	1	4	95,0
11	5	4	2	3	4	3	3	2	5	4	1	4	4	3	2	3	5	4	1	4	85,0
12	5	4	2	3	5	4	2	3	4	3	1	4	4	3	1	4	5	4	2	3	87,5
13	5	4	2	3	5	4	2	3	4	3	1	4	4	3	1	4	5	4	2	3	87,5
14	5	4	1	4	5	4	2	3	4	3	1	4	5	4	1	4	4	3	1	4	92,5
15	5	4	2	3	4	3	2	3	5	4	2	3	5	4	1	4	4	3	1	4	87,5
16	4	3	1	4	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	97,5
17	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	5	4	1	4	100,0
18	4	3	2	3	5	4	2	3	5	4	2	3	4	3	2	3	4	3	2	3	80,0
19	5	4	2	3	4	3	3	2	5	4	2	3	5	4	2	3	5	4	2	3	82,5
20	5	4	1	4	5	4	2	3	4	3	1	4	5	4	1	4	5	4	1	4	95,0
21	4	3	2	3	4	3	2	3	5	4	2	3	5	4	2	3	5	4	2	3	82,5
22	5	4	2	3	5	4	2	3	5	4	2	3	4	3	1	4	4	3	2	3	85,0
23	4	3	1	2	3	4	3	2	3	4	3	2	3	4	3	2	3	4	3	2	3
Rata-rata																					87,1

Gambar 46. Hasil analisis keseluruhan

Cara untuk menentukan peringkat skala dari hasil perhitungan dengan metode SUS dapat dilihat pada Gambar 47. Hasil perhitungan di atas menunjukkan skor akhir 87,1. Sistem ini mendapatkan predikat *Acceptable* dengan *grade B* dan mendapatkan *adjective rating excellent*. Dengan demikian sistem ini sudah layak untuk digunakan.



Gambar 47. Skala SUS

BAB V. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang bisa didapat dari tugas akhir ini yaitu telah berhasil membangun sebuah sistem informasi manajemen pegawai berbasis *mobile* untuk diterapkan di PT XYZ. Sistem informasi ini bertujuan untuk memfasilitasi pegawai dalam proses pencarian data dan memudahkan pegawai ketika ingin mendapatkan data kepegawaian seperti data pribadi, data lamaran, riwayat pendidikan, riwayat mutasi dan riwayat cuti yang berkaitan dengan dirinya secara *real-time*. Hal ini akan meningkatkan kinerja, efektivitas dan efisiensi proses administrasi perusahaan.

5.2. Saran

Saran ini ditujukan untuk pengembang selanjutnya supaya mengembangkan sistem informasi manajemen pegawai ini menjadi sistem yang utuh dan lengkap dengan menambahkan fitur-fitur manajemen pegawai lainnya seperti pengaturan gaji, pemantauan kinerja, dan lain-lain.

TA - Irfandi (turnitin)

ORIGINALITY REPORT



PRIMARY SOURCES

1	Submitted to Universitas Islam Syekh-Yusuf Tangerang	1 %
2	repository.ub.ac.id	1 %
3	Submitted to Universitas Negeri Jakarta	<1 %
4	Submitted to Politeknik Negeri Bandung	<1 %
5	Submitted to Sriwijaya University	<1 %
6	docplayer.info	<1 %
7	repository.polinela.ac.id	<1 %
8	www.slideshare.net	<1 %
9	Submitted to Ajou University Graduate School	<1 %

10	adoc.pub Internet Source	<1 %
11	Submitted to Universitas Brawijaya Student Paper	<1 %
12	repo.darmajaya.ac.id Internet Source	<1 %
13	repository.its.ac.id Internet Source	<1 %
14	de.scribd.com Internet Source	<1 %
15	repository.pelitabangsa.ac.id Internet Source	<1 %
16	eprints.uny.ac.id Internet Source	<1 %
17	yusupramlan1007.blogspot.com Internet Source	<1 %
18	ojs.stmikpringsewu.ac.id Internet Source	<1 %
19	Submitted to UIN Syarif Hidayatullah Jakarta Student Paper	<1 %
20	pindosaputra16.blogspot.com Internet Source	<1 %
21	repo.unand.ac.id Internet Source	<1 %

22	eprints.akakom.ac.id Internet Source	<1 %
23	www.scribd.com Internet Source	<1 %
24	elib.unikom.ac.id Internet Source	<1 %
25	es.scribd.com Internet Source	<1 %
26	santimaya123.blogspot.com Internet Source	<1 %
27	Admin Teika, Govin Sony, Yulianus Palopak. "Perancangan Aplikasi Pengarsipan (E-Archive) Dokumen Kepegawaian Pada Universitas Advent Indonesia", TelKa, 2018 Publication	<1 %
28	doku.pub Internet Source	<1 %
29	repository.atmaluhur.ac.id Internet Source	<1 %
30	journal.universitassuryadarma.ac.id Internet Source	<1 %
31	jurnal.dharmawangsa.ac.id Internet Source	<1 %
32	Submitted to Padjadjaran University Student Paper	<1 %

33	eprints.binadarma.ac.id Internet Source	<1 %
34	repository.uinjkt.ac.id Internet Source	<1 %
35	Submitted to Universitas Dian Nuswantoro Student Paper	<1 %
36	biz.kompas.com Internet Source	<1 %
37	conference.binadarma.ac.id Internet Source	<1 %
38	e-jurnal.uajy.ac.id Internet Source	<1 %
39	repositori.uin-alauddin.ac.id Internet Source	<1 %
40	text-id.123dok.com Internet Source	<1 %
41	Mustakim, Nur Fitrianingsih, Ita Fitriati. "Pengembangan Aplikasi E-Raport Berbasis Graphical User Interface (GUI) dengan Menggunakan VB.Net 2010 di SMKN 10 Bima", JURNAL PENDIDIKAN MIPA, 2019 Publication	<1 %
42	Submitted to Universitas Pamulang Student Paper	<1 %

43	dspace.uii.ac.id Internet Source	<1 %
44	ejournal.bsi.ac.id Internet Source	<1 %
45	Submitted to iGroup Student Paper	<1 %
46	Agus Sulaksono, Julius Nursyamsi, Haryono. "PERANCANGAN ENTERPRISE RESOURCE PLANNING (ERP) PADA SISTEM PENJUALAN BATU BATA DENGAN APLIKASI ODOO DI IKM BATU BATA MERAH MERSI", Jurnal Ekonomi dan Manajemen, 2022 Publication	<1 %
47	Arfa Fakaubun. "KURANGNYA PEMAHAMAN APARATUR SIPIL NEGARA TERHADAP PENANGANAN KEARSIPAN YANG BAIK PADA UNIT PELAKSANA TEKNIS SEKOLAH USAHA PERIKANAN MENENGAH SORONG LINGKUP KEMENTERIAN KELAUTAN DAN PERIKANAN", Jurnal Pari, 2019 Publication	<1 %
48	dokumen.tips Internet Source	<1 %
49	ecampus.sttind.ac.id Internet Source	<1 %
50	eprints.ums.ac.id	

Internet Source

<1 %

51 etd.repository.ugm.ac.id <1 %
Internet Source

52 pt.scribd.com <1 %
Internet Source

53 repository.usahidsolo.ac.id <1 %
Internet Source

54 repository.usd.ac.id <1 %
Internet Source

55 Bayu Priyatna. "PENERAPAN METODE USER CENTERED DESIGN (UCD) PADA SISTEM PEMESANAN MENU KULINER NUSANTARA BERBASIS MOBILE ANDROID", AIMS: Jurnal Accounting Information System, 2019
Publication

56 Kevin Alexander Harjanto, Magdalena Ariance Ineke Pakereng. "Perancangan Sistem Pengajuan Cuti Izin Sakit (CIS) Menggunakan Platform Pega System", Jurnal Informatika, 2020
Publication

57 aryhm76.blogspot.com <1 %
Internet Source

58 core.ac.uk <1 %
Internet Source

59	docplayer.net Internet Source	<1 %
60	eprints.uty.ac.id Internet Source	<1 %
61	fr.scribd.com Internet Source	<1 %
62	jurnalmahasiswa.unesa.ac.id Internet Source	<1 %
63	lib.unnes.ac.id Internet Source	<1 %
64	repository.mercubuana.ac.id Internet Source	<1 %
65	adetyaprtw.blogspot.com Internet Source	<1 %
66	eprints.upj.ac.id Internet Source	<1 %
67	id.scribd.com Internet Source	<1 %
68	jurnal.fmipa.unila.ac.id Internet Source	<1 %
69	jurnal.ulb.ac.id Internet Source	<1 %
70	jurusan.tik.pnj.ac.id Internet Source	<1 %

71	juti.if.its.ac.id Internet Source	<1 %
72	repository.uma.ac.id Internet Source	<1 %
73	stie-pertiwi.ac.id Internet Source	<1 %
74	tutorialdesaindreamweaver.wordpress.com Internet Source	<1 %
75	portal-ilmu.com Internet Source	<1 %
76	repository.uin-suska.ac.id Internet Source	<1 %
77	Ida Fitriani. "SISTEM INFORMASI DATA USULAN PROYEK PADA BADAN PERENCANAAN PEMBANGUNAN DAERAH (BAPPEDA) CIANJUR", Media Jurnal Informatika, 2020 Publication	<1 %
78	syifamss.wordpress.com Internet Source	<1 %

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off