**PEMANFAATAN FRAMEWORK REACT NATIVE DALAM PENGEMBANGAN APLIKASI SISTEM INFORMASI MANAJEMEN PEGAWAI BERBASIS MOBILE**

**PADA PT. XYZ**

(Laporan Tugas Akhir Mahasiswa)

**Oleh**

**Irfandi Iqbal Abimanyu**

**NPM 19753027**



**POLITEKNIK NEGERI LAMPUNG**

**BANDAR LAMPUNG**

**2023**

# KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT karena dengan limpahan rahmat dan hidayah-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul “Pemanfaatan Framework React Native Dalam Pengembangan Aplikasi Manajemen Pegawai Berbasis Mobile Pada PT. XYZ”. Penulis mendapat bimbingan dan dukungan dari berbagai pihak, penulis menyampaikan rasa terima kasih kepada :

1. Arif Makhsun, S.E., M.S.Ak. Selaku ketua jurusan ekonomi dan bisnis Politeknik Negeri Lampung.
2. Dewi Kania Widyawati, S.Kom., M.Kom. Selaku ketua program studi manajemen informatika Politeknik Negeri Lampung sekaligus dosen penguji I yang telah memberikan arahan dan bimbingan dalam proses penulisan tugas akhir ini.
3. Imam Asrowardi, S.Kom., M.Kom. Selaku dosen pembimbing I yang telah memberikan bimbingan, saran, dan masukan kepada penulis sehingga tugas akhir ini dapat terselesaikan dengan baik.
4. Dr. Septafiansyah Dwi Putra, S.T., M.T. Selaku dosen pembimbing II yang telah memberikan ilmu, bimbingan, motivasi dan saran selama kuliah dan penyusunan tugas akhir ini.
5. Oki Arifin, S.Kom., M.Cs. selaku Dosen penguji II yang telah memberikan saran dalam penyusunan tugas akhir.
6. Seluruh dosen dan teknisi program studi manajemen informatika Politeknik Negeri Lampung yang telah memberikan ilmu dan motivasi selama kuliah.
7. Angga, Selaku pimpinan PT XYZ.
8. Fahma Abdurrahman Selaku pembimbing lapang PT XYZ.
9. Teman-teman angkatan 19 program studi manajemen informatika yang sudah banyak membantu dan saling memberikan dukungan.
10. Bapak Trimiadi dan Ibu Nur Handayani selaku orang tua penulis yang selalu memberikan dukungan dan doa demi keberhasilan anaknya.
11. Shafa dan Hanum selaku adik penulis yang selalu menjadi motivasi terbesar penulis dalam menyelesaikan tugas akhir ini.
12. Keluarga penulis yang lain yang selalu mendukung dan memberikan doanya kepada penulis.
13. Teman-teman penulis yang telah memberikan dukungan dalam menyelesaikan penulisan tugas akhir ini.

Pada penulisan tugas akhir ini penulis menyadari bahwa masih banyak kekurangan, oleh karena itu penulis mengharapkan kritik dan saran. Semoga penulisan tugas akhir ini bermanfaat bagi yang membacanya.

Bandar Lampung, 10 Januari 2023

Penulis

# DAFTAR ISI

**Halaman**

[**KATA** **PENGANTAR** 2](#_Toc124809948)

[**DAFTAR** **ISI** 4](#_Toc124809949)

[**DAFTAR** **TABEL** 6](#_Toc124809950)

[**DAFTAR** **GAMBAR** 7](#_Toc124809951)

[**BAB** **I.** **PENDAHULUAN** 8](#_Toc124809952)

[1.1. Latar Belakang 8](#_Toc124809953)

[1.2. Tujuan 9](#_Toc124809954)

[1.3. Kerangka Pemikiran 10](#_Toc124809955)

[1.4. Kontribusi 12](#_Toc124809956)

[**BAB** **II.** **TINJAUAN** **PUSTAKA** 13](#_Toc124809957)

[2.1. Sistem Informasi Manajemen 13](#_Toc124809958)

[2.2. Arsip 13](#_Toc124809959)

[2.3. Aplikasi Mobile 15](#_Toc124809960)

[2.3.1. JavaScript 17](#_Toc124809961)

[2.3.2. React Native 17](#_Toc124809962)

[2.3.3. NativeBase 18](#_Toc124809963)

[2.3.4. Application Programming Interface 18](#_Toc124809964)

[2.3.5. Representatinal State Transfer 19](#_Toc124809965)

[2.4. Perancangan Sistem 20](#_Toc124809966)

[2.4.1. Mapping Chart 21](#_Toc124809967)

[2.4.2. Unified Modelling Language 22](#_Toc124809968)

[2.5. Metode Pengembangan Sistem 24](#_Toc124809969)

[2.5.1. Scrum 24](#_Toc124809970)

[2.6. Pengujian Sistem 26](#_Toc124809971)

[2.6.1. System Usability Scale 26](#_Toc124809972)

[2.7. Jurnal terkait 26](#_Toc124809973)

[**BAB** **III.** **METODOLOGI** **PELAKSANAAN** 29](#_Toc124809974)

[3.1. Tempat dan Waktu 29](#_Toc124809976)

[3.2. Alat dan Bahan 29](#_Toc124809977)

[3.2.1. Perangkat Keras (*Hardware*) 29](#_Toc124809978)

[3.2.2. Perangkat Lunak (*Software*) 29](#_Toc124809979)

[3.3. Metode Pengumpulan Data 30](#_Toc124809980)

[3.4. Metode Pengembangan Sistem 30](#_Toc124809981)

[3.4.1. Product Backlog 30](#_Toc124809982)

[3.4.2. Sprint Backlog 30](#_Toc124809983)

[3.4.3. Sprint Planning 30](#_Toc124809984)

[3.4.4. Sprint 30](#_Toc124809985)

[3.4.5. Reviewdan Restrospective 31](#_Toc124809986)

[**DAFTAR** **PUSTAKA** 51](#_Toc124809987)

# DAFTAR TABEL

**Tabel Halaman**

[Tabel 1. Metode HTTP. 20](#_Toc126687692)

[Tabel 2. Simbol-simbol *mapping chart.* 21](#_Toc126687693)

[Tabel 3. Simbol-simbol diagram *use case.* 22](#_Toc126687694)

[Tabel 4. Lanjutan simbol-simbol *use case.* 23](#_Toc126687695)

[Tabel 5. Simbol-simbol *activity diagram.* 23](#_Toc126687696)

[Tabel 6. Lanjutan simbol-simbol *activity diagram.* 24](#_Toc126687697)

# DAFTAR GAMBAR

**Gambar Halaman**

[Gambar 1. Kerangka pemikiran 12](#_Toc126917008)

[Gambar 2. Struktur organisasi PT XYZ. 33](#_Toc126917009)

[Gambar 3. Daftar *item product backlog.* 34](#_Toc126917010)

[Gambar 4. Daftar *item sprint backlog.* 35](#_Toc126917011)

[Gambar 5. Daftar *item* *sprint planning*. 35](#_Toc126917012)

[Gambar 6. Rancangan *use case* diagram. 36](#_Toc126917013)

[Gambar 7. *Activity* diagram *login*. 37](#_Toc126917014)

[Gambar 8. *Activity* diagram data pribadi. 38](#_Toc126917015)

[Gambar 9. *Activity* diagram riwayat pendidikan. 39](#_Toc126917016)

[Gambar 10. *Activity* diagram riwayat mutasi. 40](#_Toc126917017)

[Gambar 11. *Activity* diagram riwayat cuti. 41](#_Toc126917018)

[Gambar 12. *Activity* diagram data lamaran. 42](#_Toc126917019)

[Gambar 13. *Activity* diagram *reset password.* 43](#_Toc126917020)

[Gambar 14. *Activity* diagram pengajuan cuti. 44](#_Toc126917021)

[Gambar 15. Daftar *item sprint* desain tampilan aplikasi (*wireframing*). 45](#_Toc126917022)

[Gambar 16. *Wireframe interface splash screen.* 46](#_Toc126917023)

[Gambar 17. *Wireframe interface login screen*. 46](#_Toc126917024)

[Gambar 18. *Wireframe interface home screen*. 47](#_Toc126917025)

[Gambar 19. *Wireframe interface* data pribadi *screen.* 48](#_Toc126917026)

[Gambar 20. *Wireframe interface* kontak & akun *screen*. 48](#_Toc126917027)

[Gambar 21. *Wireframe interface reset password screen*. 49](#_Toc126917028)

[Gambar 22. *Wireframe interface* riwayat pendidikan *screen*. 50](#_Toc126917029)

[Gambar 23. *Wireframe interface* riwayat mutasi *screen*. 50](#_Toc126917030)

[Gambar 24. *Wireframe interface* riwayat cuti *screen*. 51](#_Toc126917031)

[Gambar 25. *Wireframe interface* pengajuan cuti *screen*. 52](#_Toc126917032)

[Gambar 26. *Wireframe interface* data lamaran *screen*. 52](#_Toc126917033)

[Gambar 27. Daftar *item sprint* desain tampilan aplikasi (*user interface*). 53](#_Toc126917034)

[Gambar 28. Desain *user interface splash screen.* 54](#_Toc126917035)

[Gambar 29. Desain *user interface login screen.* 54](#_Toc126917036)

[Gambar 30. Desain *user interface home screen*. 55](#_Toc126917037)

[Gambar 31. Desain *user interface* data pribadi *screen*. 56](#_Toc126917038)

[Gambar 32. Desain *user interface* kontak & akun *screen*. 56](#_Toc126917039)

[Gambar 33. Desain *user interface reset password screen*. 57](#_Toc126917040)

[Gambar 34. Desain *user* *interface* riwayat pendidikan *screen*. 58](#_Toc126917041)

[Gambar 35. Desain *user interface* riwayat mutasi *screen* 58](#_Toc126917042)

[Gambar 36. Desain *user interface* riwayat cuti *screen*. 59](#_Toc126917043)

[Gambar 37. Desain *user interface* pengajuan cuti *screen*. 60](#_Toc126917044)

[Gambar 38. Desain *user interface* pengajuan cuti *screen*. 60](#_Toc126917045)

[Gambar 39. Daftar *item sprint* pengembangan aplikasi. 61](#_Toc126917046)

# BAB I. PENDAHULUAN

## 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 tepat dan akurat untuk membantu proses pengambilan keputusan dengan mudah (Sadikin & Wiranda, 2022). Penerapan sistem informasi manajemen pegawai pada setiap instansi bertujuan untuk mempermudah pengelolaan data dan pengendalian pegawai dengan cepat dan efektif.

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, salah satunya yaitu arsip kepegawaian. Arsip kepegawaian merupakan salah satu jenis arsip yang berisi kumpulan data pegawai seperti daftar riwayat hidup, surat lamaran, surat keputusan pengangkatan, dll. Untuk itu optimalisasi pengelolaan arsip kepegawaian menjadi penting dilakukan di sebuah instansi seperti PT XYZ.

PT XYZ merupakan Badan Usaha Milik Negara (BUMN) yang bertanggung jawab membangkitkan dan menyediakan listrik di Sumatera bagian tengah dan selatan. PT XYZ merupakan pusat administrasi pembangkitan, semua unit pembangkitan yang ada di provinsi Lampung direncanakan, dipelihara, dan diawasi oleh PT XYZ. Perusahaan ini memiliki beberapa divisi bagian yaitu : *Engineering*, Ophar (operasi dan pemelihara), dan KSA (logistik, akuntansi, SDM dan umum). Arsip kepegawaian PT XYZ dikelola di bagian SDM yang berisi data pribadi, keluarga, kenaikan pangkat, mutasi, cuti dll. Saat ini pengelolaan arsip kepegawaian pada PT XYZ masih dilakukan dengan cara manual, mulai dari pegawai diterima sampai dengan pensiun masih dikelola dengan pemberkasan yang disimpan di dalam ordner map dan dikelompokkan berdasarkan tahun masuk pegawai. Ketika ada perubahan data, pencarian berkas pegawai memerlukan waktu yang lama. Hal ini, menyebabkan proses administrasi pegawai yang dilakukan belum efektif dan efisien.

Maka diperlukan sebuah sistem yang mempercepat proses administrasi pada bagian pegawai yaitu “Pemanfaatan FrameworkReactNative Dalam Pengembangan Sistem Informasi Manajemen Pegawai Berbasis Mobile pada PT XYZ” untuk mempermudah pegawai ketika ingin mendapatkan data yang diinginkan tanpa harus menunggu proses pencarian di bagian SDM kapan saja dan dimana saja. Pada penulisan tugas akhir ini kami akan mengkaji data apa saja yang dibutuhkan untuk membangun sebuah sistem mengenai manajemen data kepegawaian. Sistem yang akan dibangun menggunakan bahasa pemrograman JavaScript, dengan *framework* ReactNative dan Nativebase *UI.* Manfaat yang diharapkan dari penulisan tugas akhir ini adalah untuk mempermudah pegawai dalam melakukan pencarian data yang dibutuhkan di PT XYZ serta membantu perusahaan agar memiliki media untuk melakukan pencadangan arsip kepegawaian.

## Tujuan

Tujuan dari penulisan tugas akhir ini adalah mendesain dan mengimplementasikan aplikasi sistem informasi manajemen pegawai berbasis *mobile* untuk diterapkan di PT XYZ. Sistem aplikasi ini berfungsi untuk mendigitalisasi proses pencarian data dan informasi kepegawaian agar mempermudah pegawai dalam memperoleh data yang diinginkan, seperti data pribadi pegawai, data riwayat pendidikan, surat keputusan mutasi, surat cuti, surat pensiun, dll.

## Kerangka Pemikiran

Pengolahan arsip kepegawaian pada PT XYZ saat ini belum efektif dan efisien. Penyimpanan data masih disimpan di dalam lemari dan belum dicadangkan ke *cloud* sehingga berisiko hilang dan lambannya proses pencarian data. Hal ini dapat menurunkan kinerja bagian SDM ketika melayani pegawai yang membutuhkan datanya, selain itu proses pelayanan yang hanya dapat dilakukan di hari kerja dapat mengurangi fleksibilitas akses pegawai terhadap datanya. Untuk mengatasi permasalahan tersebut perlu adanya sebuah sistem aplikasi yang dapat membantu pegawai untuk mengakses data yang diinginkan dengan cepat kapan saja dan di mana saja, tanpa harus memprosesnya ke bagian SDM. Maka muncul sebuah solusi “Pemanfaatan FrameworkReactNative Dalam Pengembangan Sistem Informasi Manajemen Pegawai Berbasis Mobile pada PT XYZ” dengan menggunakan metode pengembangan sistem s*crum.* Tahapan kerangka pemikiran dalam pembuatan aplikasi ini disajikan pada gambar 1.



Gambar 1. Kerangka pemikiran

## Kontribusi

Kontribusi pemanfaatan *framework react native* dalam pengembangan sistem informasi manajemen pegawai berbasis *mobile* pada PT XYZ diharapkan berhasil mengoptimalkan proses administrasi arsip kepegawaian pada bagian SDM di PT XYZ yang sebelumnya manual menjadi otomatis menggunakan sistem sehingga informasi yang disajikan untuk pegawai sudah sesuai dengan data yang terbaru. Selain itu, mempermudah pegawai dalam mengakses arsipnya melalui *smartphone* mereka tanpa harus mengurusnya langsung ke bagian SDM. Hal ini akan meningkatkan kinerja dan efisiensi proses administrasi perusahaan.

# BAB II. TINJAUAN PUSTAKA

## 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 bagi manajer demi 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 menurut 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.

## 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, masa kini, maupun 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).

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

Arsip juga dibagi menjadi beberapa jenis menurut Sugiarto dan Wahyono (2014) yaitu :

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.
7. Arsip yang bernilai sejarah seperti laporan bulanan, tahunan, dll.
8. Arsip yang bernilai keuangan seperti kuitansi, nota pembayaran, dll.
9. Arsip yang bernilai informasi seperti pengumuman dan undangan.
10. Arsip yang bernilai ilmiah seperti laporan penelitian, skripsi, dll.
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 :
16. Arsip yang tidak berguna seperti surat undangan.
17. Arsip yang berguna seperti surat cuti, surat izin, dan presensi pegawai.
18. Arsip yang penting seperti laporan keuangan, surat keputusan, dll.
19. Arsip yang vital seperti akta kelahiran, ijazah, sertifikat pelatihan, dll.

Salah satu jenis arsip berdasarkan subjeknya yaitu arsip kepegawaian, menurut Suparman (2020) arsip kepegawaian adalah semua arsip yang berhubungan dengan masalah kepegawaian, seperti surat lamaran, daftar riwayat hidup, data pribadi, surat-surat keputusan pegawai, absensi pegawai, dll.

## Aplikasi Mobile

Menurut Richard (2013) 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. 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.*

1. 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. Pengelompokan perangkat lunak aplikasi bergantung pada kegunaannya bagi pengguna.

1. 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.
2. Aplikasi desain yang digunakan para desainer mencakup aplikasi menggambar, aplikasi edit gambar, aplikasi pengolah *vector*, aplikasi 3D, dan aplikasi pengolah video.
3. Aplikasi pengolah suara yang digunakan para musisi.
4. 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.
5. Aplikasi daring mencakup aplikasi yang menggunakan konektivitas internet ketika penggunaannya seperti aplikasi email, browser web, aplikasi FTP, aplikasi *remote*, dsb.
6. 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*. 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.

### JavaScript

JavaScript merupakan bahasa pemrograman tingkat tinggi dinamis yang mampu dikembangkan dengan gaya pemrograman fungsional atau gaya pemrograman berorientasi objek. JavaScriptmerupakan bahasa pemrograman yang memerlukan interpreter untuk menjalankan kodenya. Dahulu JavaScripthanya 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 JavaScriptkeluar 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.

### React Native

ReactNative adalah salah satu kerangka kerja (*framework*) JavaScript besutan Facebookyang 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 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).

### NativeBase

*Library* komponen UI/UX (*User Interface/User Exprience*) 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).

### Application Programming Interface

*Application Programming Interface* (API) merupakan kumpulan aturan yang ditentukan oleh pengembang untuk memfasilitasi aplikasi untuk dapat berkomunikasi dengan aplikasi lain. API bertindak sebagai perantara yang memproses pertukaran data yang dilakukan 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 pengerjaan 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.

1. 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.

1. 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.

1. REST (*Representational State Transfer*)

REST adalah seperangkat prinsip pengembangan arsitektur web API.

### Representatinal State Transfer

Menurut Sirsha dan Mamatha (2020) *Representatinal State Transfer* atau RESTmerupakan 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. 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.

1. *Statelessness*

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

1. *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.

1. *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.

Tabel 1. Metode HTTP.

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

Sumber: (MDN Mozilla, 2022)

## 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 dan untuk memenuhi kebutuhan pengguna. Perancangan sistem bisanya digambarkan dalam bentuk diagram alir yang menampilkan proses secara runtut dan diagram-diagram pendukung lainnya (Aziz, N., Pribadi, G., & Nurcahya, 2020).

### 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).

Tabel 2. Simbol-simbol *mapping chart.*

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Fungsi |
|  | Dokumen | Menyatakan *input* dokumen yang dicetak |
|  | *Connector* | Menyatakan sambungan satu dengan yang lainnya pada proses berikutnya |
|  | Proses | Proses pengolahan yang ditugaskan oleh komputer |
|  | *Display* | Menyatakan output yang digunakan |
|  | Arsip | Mengarsipkan data di dalam program |
|  | Terminal | Awalan dan akhiran pada program |
|  | *Connecting* *Line* | Menghubungkan simbol dengan simbol lainnya dengan menyatakan suatu alur proses |

Sumber: (Fitriyana & Susianto, 2018)

### Unified Modelling Language

Menurut Putra dan Andriani (2019) *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. 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 penggunanya. *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*).

Tabel . Simbol-simbol diagram *use case.*

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Fungsi |
| (1) | (2) | (3) |
|  | 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. |
|  | *Association* | Relasi asosiasi antara *use case* dengan pengguna. |

Tabel . Lanjutan simbol-simbol *use case.*

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Fungsi |
| (1) | (2) | (3) |
|  | *Includes* | Relasi ketika sebuah *use case* digambarkan menggunakan fungsionalitas dari *use case* yang lain. |
|  | *Extends* | Relasi yang digunakan untuk menyertakan perilaku opsional dari *use case* yang diperluas. |

Sumber: (Visual Paradigm, 2022b)

1. **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 . Simbol-simbol *activity diagram.*

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Fungsi |
| (1) | (2) | (3) |
|  | *Initial* *Node* | Menggambarkan awal dari serangkaian tindakan atau kegiatan. |
|  | *Activity* | Menggambarkan serangkaian tindakan atau kegiatan. |
|  | *Decision Node* | Menggambarkan kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur. |
|  | *Fork Node* | Membagi perilaku menjadi serangkaian aktivitas (atau tindakan) paralel atau bersamaan. |

Tabel . Lanjutan simbol-simbol *activity diagram.*

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Fungsi |
| (1) | (2) | (3) |
|  | *Join Node* | Menyatukan kembali serangkaian aktivitas (atau tindakan) yang paralel atau bersamaan. |
|  | *Swimlane and Partition* | Mengelompokkan aktivitas yang dilakukan oleh aktor yang sama pada diagram aktivitas atau mengelompokkan aktivitas dalam satu utas. |
|  | *Activity Final Node* | Menghentikan semua aliran kontrol dan aliran objek dalam aktivitas (atau tindakan) |

Sumber: (Visual Paradigm, 2022a)

## Metode Pengembangan Sistem

### Scrum

*Agile* merupakan metodologi pengembangan aplikasi yang dalam prosesnya beradaptasi dengan cepat terhadap segala bentuk perubahan. Sedangkan *scrum* merupakan kerangka kerja yang mengimplementasikan metode pengembangan aplikasi *agile*. *Scrum* adalah sebuah kerangka kerja yang digunakan untuk mengelola pengembangan sebuah produk yang kompleks dengan tujuan menjadikan produk bernilai setinggi mungkin secara produktivitas dan kreativitas. *Scrum* tidak terpaku dengan sebuah proses, teknik, atau metodologi karena *scrum* bisa mencangkup berbagai proses, teknik, ataupun metodologi di dalamnya (Schwaber & Sutherland, 2020). Menurut Schwaber dan Sutherland (2020) *scrum* memiliki beberapa prinsip yaitu:

1. Transparansi

Semua proses pengerjaan harus dapat dilihat secara transparan oleh pengguna yang mengerjakan dan pengguna yang menerima hasil pekerjaannya. Dengan proses yang transparan pengambilan keputusan dapat dilakukan dengan baik dan dapat terhindar dari risiko.

1. Inspeksi

Pemeriksaan hasil pengerjaan harus sering dilakukan dengan seksama untuk mendeteksi fitur-fitur baru atau bahkan mendeteksi kesalahan yang tidak diinginkan.

1. Adaptasi

Perubahan harus dilakukan sesegera mungkin apabila produk yang dikerjakan tidak diterima, proses yang diterapkan atau bahan produksi yang belum sesuai dengan tujuan bersama. Biasanya adaptasi perubahan ini dilakukan setelah hasil di inspeksi.

Berikut ini langkah-langkah metode pengembangan sistem menggunakan *scrum* menurut Prastio dan Ani (2018) yaitu:

1. ***Product Backlog***

*Product Backlog* dibuat berdasarkan kebutuhan pengguna yang di dapat melalui proses pengumpulan data. *Requirements* pada *product* *backlog* bersifat dinamis karena penambahan kebutuhan dapat dilakukan saat mendapatkan *feedback* dari pengguna ketika mengulas kembali aplikasi.

1. ***Sprint* *Backlog***

*Sprint* *Backlog* merupakan rentang waktu pengerjaan *product* *backlog* yang telah dibagi menjadi beberapa bagian yang digunakan pada fase *sprint*. Waktu *sprint* berlangsung 1-4 minggu sesuai kesepakatan.

1. ***Sprint*** ***Planning***

*Sprint* *Planning* merupakan perencanaan dalam *product* *backlog* pada *sprint*. Pada tahap ini terdapat perancangan sistem yang dipetakan kebutuhannya secara fungsional dalam sebuah diagram. Pada proses perancangan terdapat *Unified Modelling Language* (UML).

1. ***Sprint***

*Sprint* terdiri dari beberapa bagian yang didefinisikan pada *backlog* serta harus diselesaikan sesuai dengan waktu yang ditetapkan. Pada tahap ini pembuatan aplikasi mulai dikerjakan sesuai dengan *sprint* *planning*.

1. ***Review* dan *Retrospective***

Pada tahap ini pengguna sudah dapat me-*review* aplikasi untuk tujuan pengujian terakhir. Pengujian ini dilakukan untuk mencari kesalahan sistem yang ada pada aplikasi informasi manajemen pegawai dengan menggunakan metode *System Usability Scale* (SUS). Kemudian dilakukan *retrospective* untuk mengetahui apakah terdapat masukan berupa *feedback* dari kebutuhan fitur yang sudah di *review.*

## Pengujian Sistem

Menurut Wibisono dan Baskoro (2002) 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 yaitu:

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

### System Usability Scale

*System Usability Scale* merupakan metode evaluasi sebuah aplikasi untuk menilai dan mengukur tingkat kegunaannya dengan menggunakan sebuah kuesioner sederhana yang memiliki sepuluh buah pertanyaan. Pertanyaan pada metode SUS pada nomor ganjil memiliki kalimat positif dan pada nomor genap memiliki kalimat negatif. Selanjutnya untuk menilai hasil kuesioner tersebut cukup 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.

## Jurnal terkait

Berikut ini beberapa jurnal terkait yang menjadi bahan pertimbangan dan menjadi bahan referensi pengembangan sistem.

1. Adam dkk. (2022). Dengan judul “Aplikasi Jasa Titip Belanja Berbasis Mobile di Minahasa Utara”. Karya ilmiah ini bertujuan untuk membangun sebuah aplikasi jasa titip belanja supermarket di daerah Minahasa Utara. Aplikasi yang akan dibangun akan tersedia di platform Android untuk Pelanggan dan Kurir serta *website* untuk admin dan juga toko, dengan sisi *front-end* dikembangkan dengan bahasa pemrograman Javascript menggunakan *framework* React Native, React JS, dan sisi *back-end* dikembangkan lewat bahasa pemrograman Javascript dengan memanfaatkan teknologi firebase *real-time database*.
2. Karim & Adriansyah (2022). Dengan judul “Analisis dan Perancangan Aplikasi Mobile untuk Donasi menggunakan Metode Hybrid berbasis React Native”. Karya ilmiah ini dibuat untuk memudahkan pengguna untuk berdonasi barang, dimana pun dan kapan pun. Aplikasi dibangun menggunakan *framework* React Native dengan RESTFul API dan MySQL sebagai bahasa pemrograman *database*. Pengujian yang dilakukan yaitu pengujian fungsional sistem dengan menggunakan teknik *black-box testing* dan *User Acceptance Test* (UAT) serta pengujian dengan menggunakan kuesioner.
3. Nursaid dkk. (2020). Dengan judul “Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri)”. Karya ilmiah ini dibuat untuk mempermudah pedagang yang memiliki toko Uda Fajri tanpa harus mencatat barang masuk dan keluar dengan menggunakan kertas lagi, selain itu juga dapat melakukan transaksi barang dengan menggunakan telepon genggam yang sudah terpasang aplikasi sistem pengelolaan barang.
4. Mukodimah dkk. (2019). Dengan Judul “Aplikasi Penentuan Bengkel TSM Berkualitas Untuk UKK Siswa SMK Kabupaten Pringsewu Berbasis Mobile”. *Website* untuk mengukur kelayakan lab/bengkel uji kompetensi kejuruan TSM dapat dijadikan sebagai solusi dalam menyelesaikan permasalahan pengukuran kelayakan bengkel TSM SMK secara tepat dan akurat. Dengan menerapkan 5 kriteria yang ada seperti area kerja mesin otomotif, area kerja kelistrikan, area kerja *chasis* dan pemindah tenaga, ruang penyimpanan dan instruktur dan kelengkapan peralatan. Proses pengukuran kelayakan dapat dilakukan dengan lebih tepat, akurat dan efisien.
5. Prastio & Ani (2018). Dengan judul “Aplikasi Self Service Menu Menggunakan Metode Scrum Berbasis Android (Case Study : Warkobar Café Cikarang)”. Sebuah aplikasi yang diharapkan membantu pelanggan *café*, pelayan, kasir, dapur, dan pemilik *café*. Perancangan sistem akan dibuat dengan tampilan sederhana dan mudah dipahami oleh pelanggan. Dengan begitu pemesanan menu bisa berjalan dengan cepat dan praktis, tanpa melalui proses panjang serta mengurangi kemungkinan terjadinya kesalahan komunikasi antara pelayan dan pelanggan. Sistem ini juga akan meminimalisir adanya antrean tempat duduk dalam *café*, dengan adanya fungsi reservasi dengan pemilihan tempat dan waktu yang sudah ditentukan oleh pelanggan. Sistem ini juga akan menggunakan OS Mobile populer saat ini, yaitu Android.

# BAB III. METODOLOGI PELAKSANAAN



## Tempat dan Waktu

Tugas akhir yang berjudul “Pemanfaatan FrameworkReactNative Dalam Pengembangan Sistem Informasi Manajemen Pegawai Berbasis Mobile pada PT XYZ” ini dilaksanakan pada bulan Juni 2022 hingga bulan Februari 2023 di Politeknik Negeri Lampung. Tugas akhir ini mengambil permasalahan dari tempat Praktik Kerja Lapang (PKL) pada semester sebelumnya di PT XYZ.

## Alat dan Bahan

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

### Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan dalam penulisan dan pengembangan tugas akhir ini yaitu:

1. Laptop/komputer dan kabel *charger*.
2. Tetikus (*mouse*).
3. Papan ketik (*keyboard*).

### Perangkat Lunak (*Software*)

Sedangkan perangkat lunak yang dibutuhkan dalam penulisan dan pengembangan tugas akhir ini yaitu:

1. Sistem operasi (Windows11).
2. Aplikasi pengolah kata (Microsoft Word 2021).
3. Aplikasi pembuat diagram (Figma/FigJam/EdrawMax)
4. Aplikasi sitasidaring (Mendeley).
5. Aplikasi browser web (GoogleChrome).
6. Aplikasi manajemen proyek (ClickUp).
7. Aplikasi editor kode (VisualStudioCode).
8. Kerangka kerja aplikasi (React Native).
9. Pustaka komponen UI (NativeBase).
10. Aplikasi REST *endpoint testing* (Postman/Insomnia).

## Metode Pengumpulan Data

Pengumpulan data yang dilakukan ketika penulisan tugas akhir untuk membangun “Pemanfaatan FrameworkReactNative Dalam Pengembangan Sistem Informasi Manajemen Pegawai Berbasis Mobile pada PT XYZ” menggunakan metode wawancara secara langsung kepada pihak PT XYZ yang bersangkutan yaitu Ibu Dita selaku pegawai divisi SDM PT XYZ, Bapak Angga selaku manajer PT XYZ dan berkonsultasi dengan Bapak Fahma selaku pegawai divisi Engineering sekaligus pembimbing lapangan kami ketika PKL. Informasi yang didapatkan sangat berguna dan membantu dalam pengembangan sistem tugas akhir ini.

## Metode Pengembangan Sistem

Pengembangan sistem yang dilakukan untuk membangun “Pemanfaatan FrameworkReactNative Dalam Pengembangan Sistem Informasi Manajemen Pegawai Berbasis Mobile pada PT XYZ” menggunakan metode *scrum*. Berikut ini runtutan tahapan pengembangan menggunakan metode *scrum*:

### Product Backlog

Saat merancang sebuah sistem menggunakan metode *scrum*, hal pertama yang harus dilakukan adalah menyusun *product backlog* berdasarkan kebutuhan pengguna melalui pengumpulan data dan observasi*.* Tahap *product backlog* dilaksanakan selama satu minggu mulai tanggal 7 November 2022 sampai dengan tanggal 14 November 2022.

### Sprint Backlog

*Sprint backlog* merupakan daftar *product backlog* yang telah dipilih dan terurut berdasarkan tugasnya untuk dikerjakan dalam *sprint* melalui perencanaan mengenai fungsinya dalam jangka waktu yang telah ditentukan agar menghasilkan produk yang sesuai dengan keinginan pengguna. Tahap *sprint backlog* dilaksanakan selama satu minggu mulai tanggal 14 November 2022 sampai dengan tanggal 21 November 2022.

### Sprint Planning

Melakukan perancangan dan analisis sistem yang mencakup proses bisnis yang sedang berjalan dan proses bisnis yang ditawarkan menggunakan *mapping chart* dan UML. Tahap *spint planning* dilaksanakan selama satu minggu mulai dari tanggal 21 November 2022 sampai dengan tanggal 28 November 2022.

### Sprint

Dalam pengembangan sistem ini pengerjaan aplikasi dikerjakan berdasarkan penjabaran tugas yang sudah ditentukan pada *sprint* *planning*. Dalam pengerjaan *sprint*, setiap *progress sprint* yang sudah selesai akan melalui proses pengujian fungsionalitas fiturnya terlebih dahulu lalu setiap *progress*-nya akan langsung dicatat dan dilaporkan kepada pihak PT XYZ.

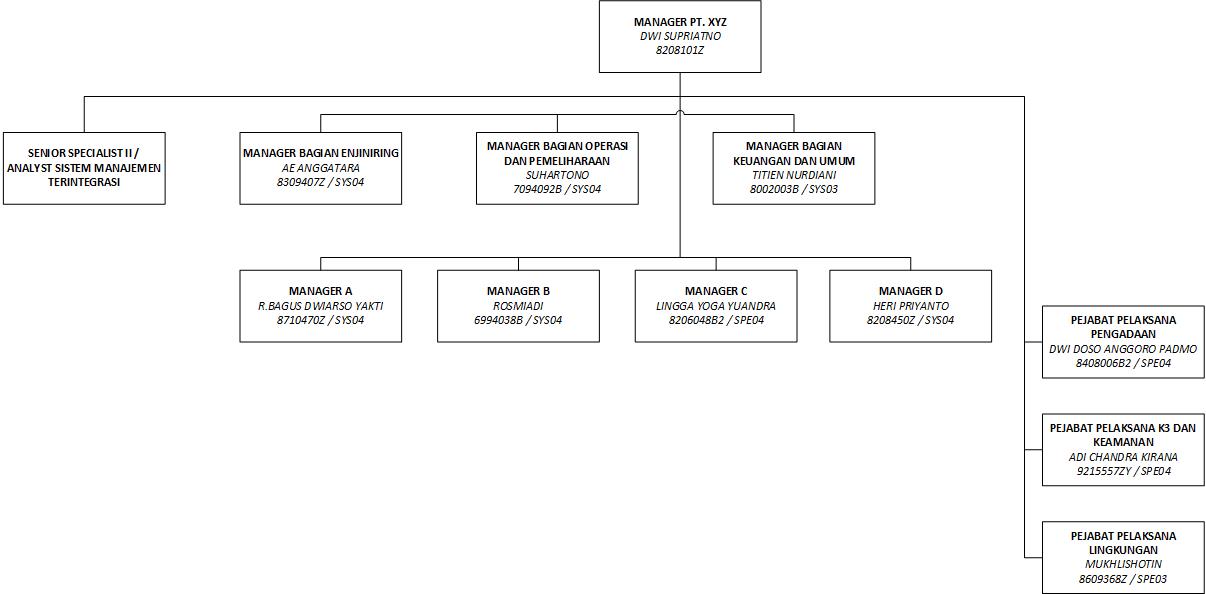
### Reviewdan Retrospective

Setelah *sprint* selesai, aplikasi diulas kembali dan di-*review* oleh pihak PT XYZ menggunakan SUS. Apabila masih ada kekurangan maka kekurangan tersebut akan ditambahkan kembali di *product backlog* dan tahapan akan diulang kembali sampai aplikasi sesuai dengan keinginan pengguna dan lolos pengujian.

# BAB IV. HASIL DAN PEMBAHASAN

## Struktur Organisasi

Berdasarkan hasil pengamatan dan pengumpulan data yang dilakukan, PT XYZ memiliki struktur organisasi seperti berikut ini.



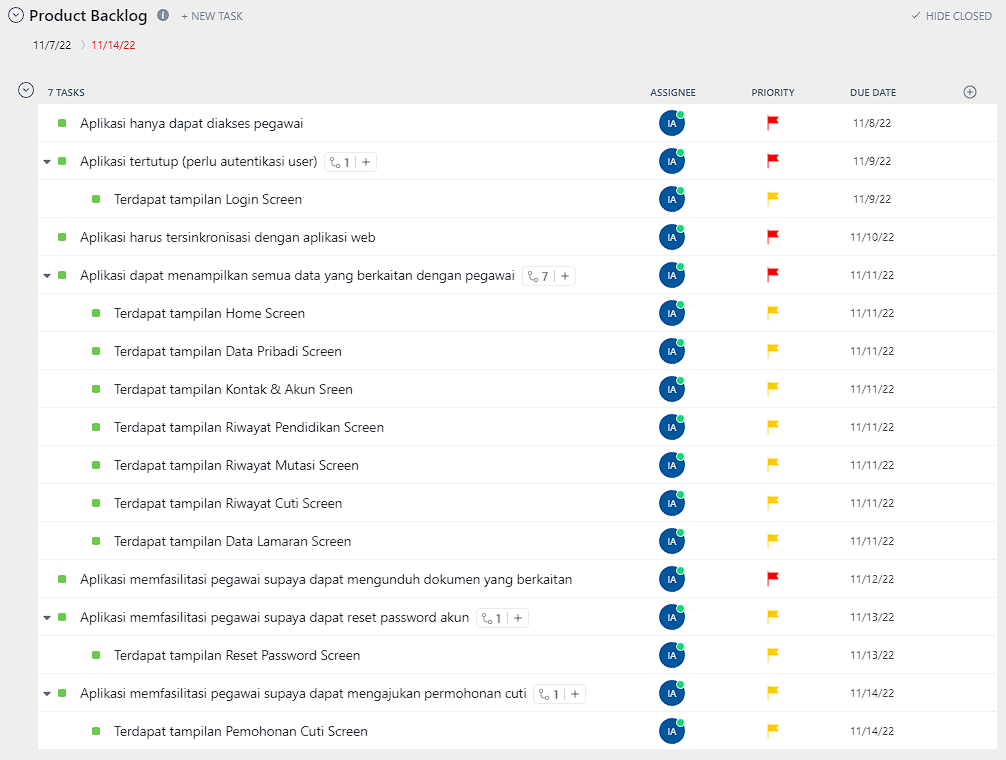
Gambar . Struktur organisasi PT XYZ.

## Hasil dan Pembahasan

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

### Product Backlog

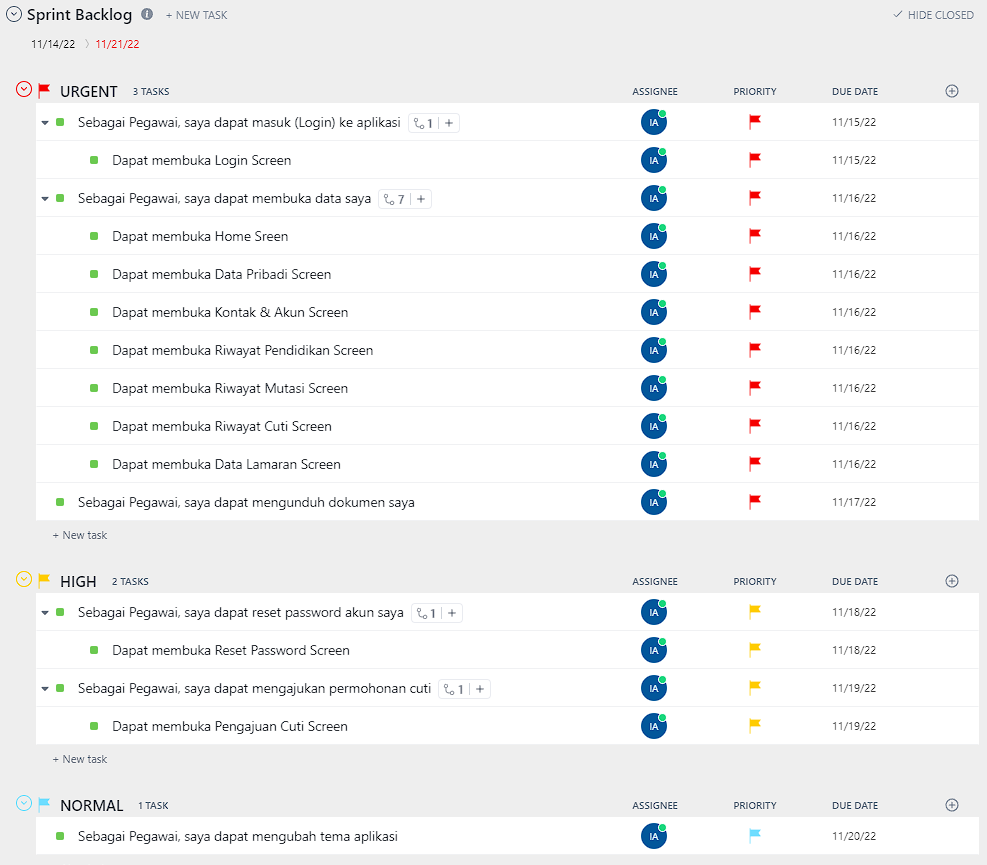
Tahap pertama yang wajib dilakukan adalah menyusun *product backlog*. *Product backlog* berisi daftar kebutuhan yang diinginkan pengguna sebelum aplikasi dibangun. Berikut ini daftar *product backlog* yang diperoleh dari eksplorasi kebutuhan pengguna melalui proses pengumpulan data.



Gambar . Daftar *item product backlog.*

### Sprint Backlog

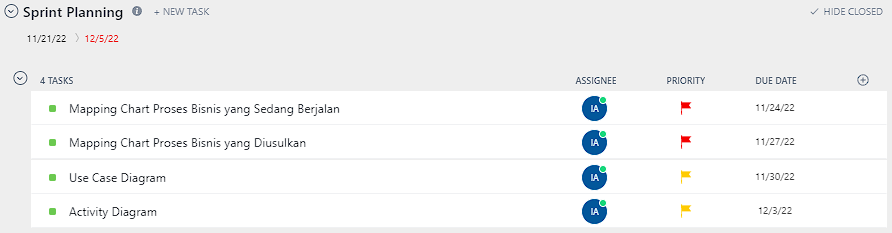
*Sprint backlog* merupakan tahap dimana setiap *item* dalam *product backlog* diurutkan berdasarkan pertimbangan prioritas pengerjaan dan perkiraan urgensi fungsionalitasnya terlebih dahulu lalu dimasukkan ke dalam daftar pengerjaan pada tahap *sprint*. Berikut ini daftar yang berisi *item sprint backlog* yang telah diurutkan*.*



Gambar . Daftar *item sprint backlog.*

### Sprint Planning

Pada tahap *sprint 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 menggunakan *mapping chart* dan UML. Berikut ini daftar *item* perencanaan dalam *sprint planning*.



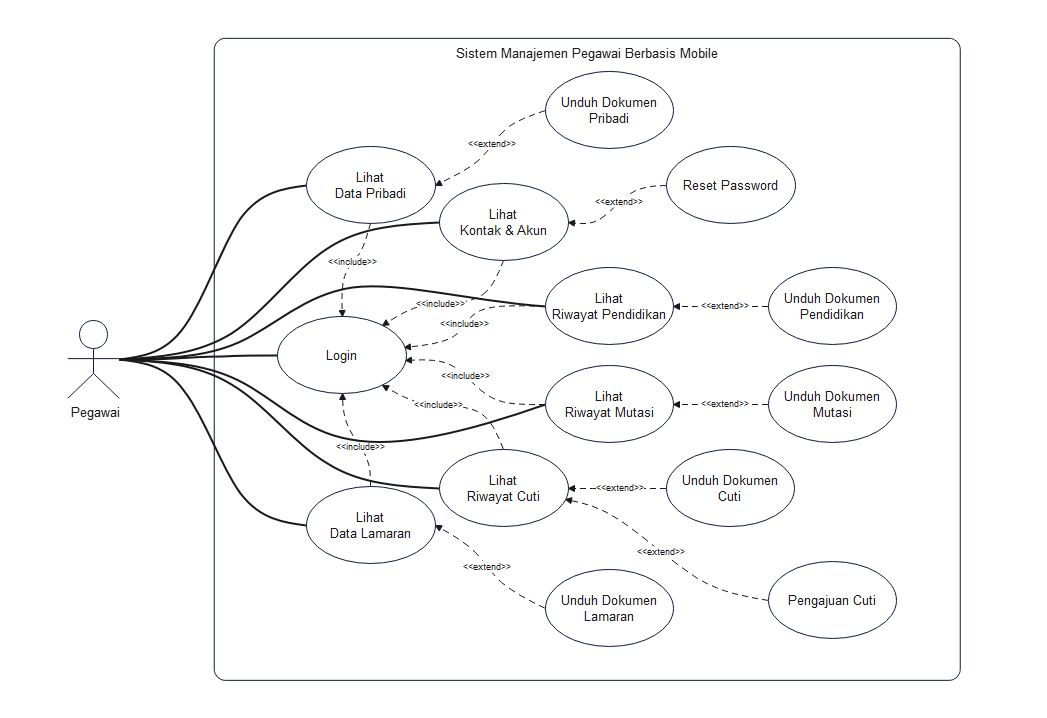
Gambar . Daftar *item* *sprint planning*.

#### Proses Bisnis Yang Sedang Berjalan

#### Proses Bisnis Yang Diusulkan

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



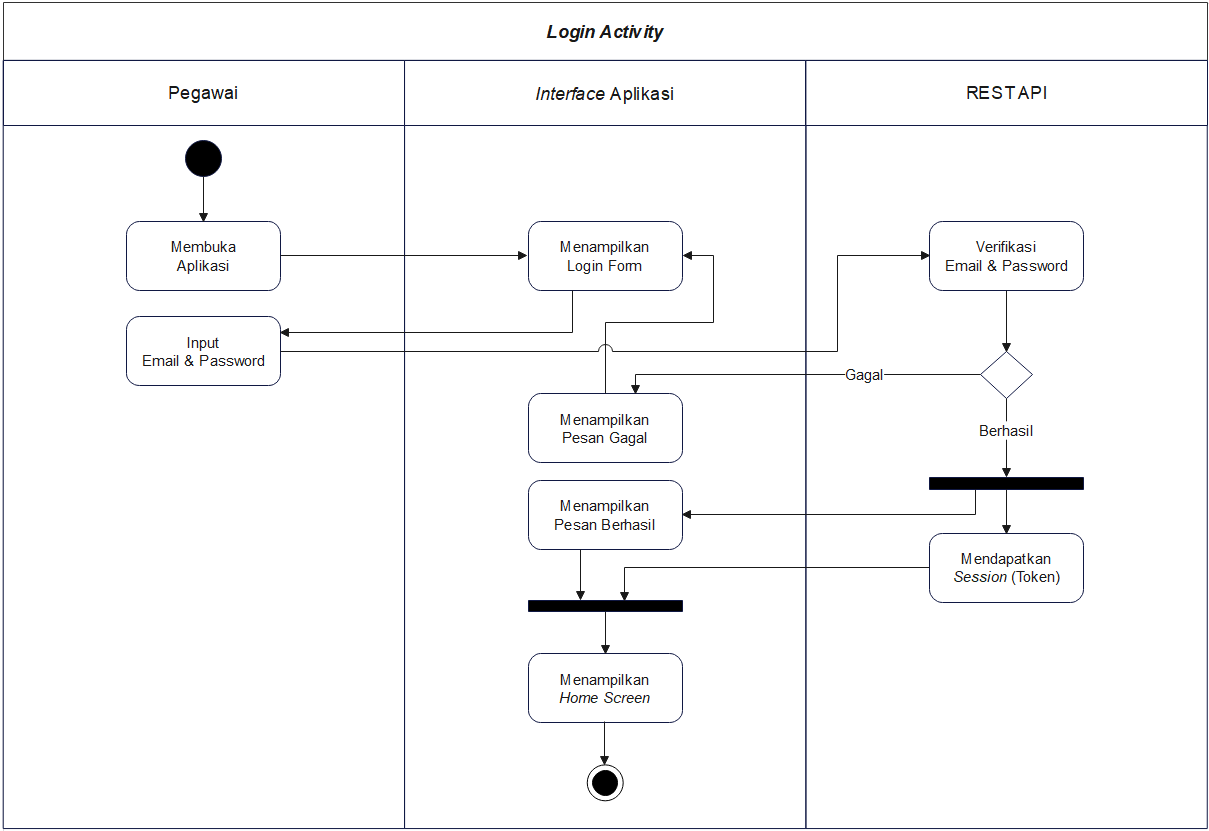
Gambar . Rancangan *use case* diagram.

Dalam rancangan *use case* diagram pada gambar 4 hanya aktor pegawai yang dapat berinteraksi dengan sistem. Pegawai dapat melakukan semua fungsi yang ada di dalam sistem manajemen pegawai berbasis *mobile*. Pegawai dapat melihat *login*, 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.

#### 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 *swimline* yaitu pegawai, *interface* aplikasi, dan REST API. Berikut ini beberapa *activity* diagram berdasarkan fungsi-fungsi yang ada.

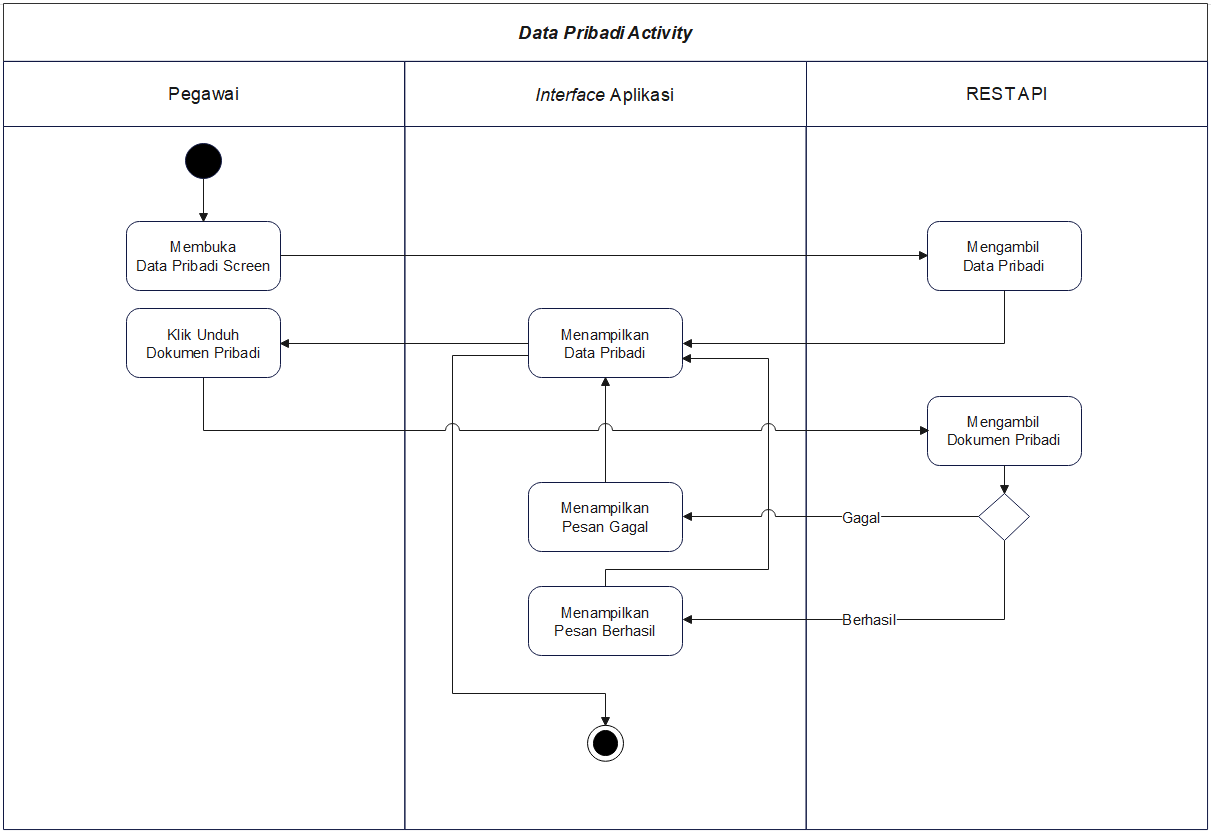
1. **Activity diagram login**



Gambar . *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 *login* 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.

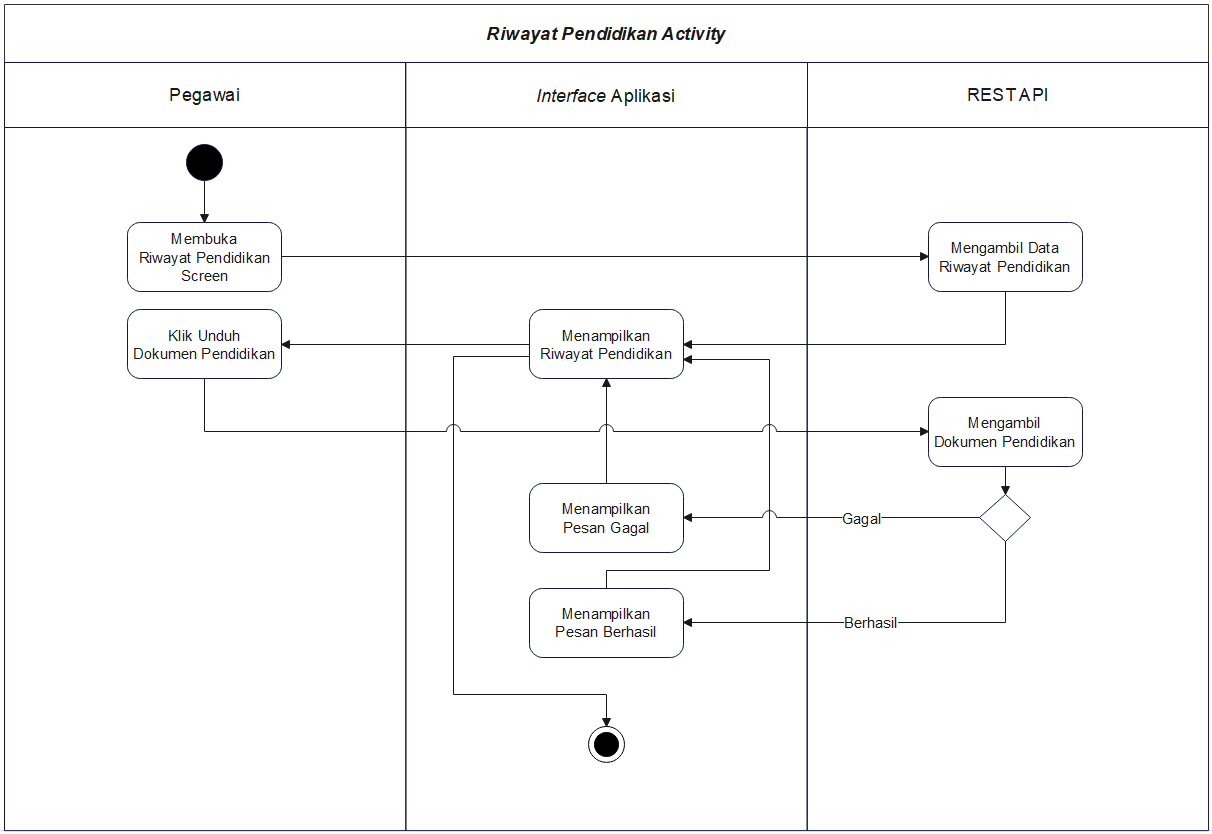
1. **Activity diagram data pribadi**



Gambar . *Activity* diagram data pribadi.

Pada *activity* diagram ini, proses akan dimulai ketika pegawai memuka halaman data pribadi. Sebelum data pribadi ditampilkan *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data pribadi berdasarkan pegawai, data yang 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.

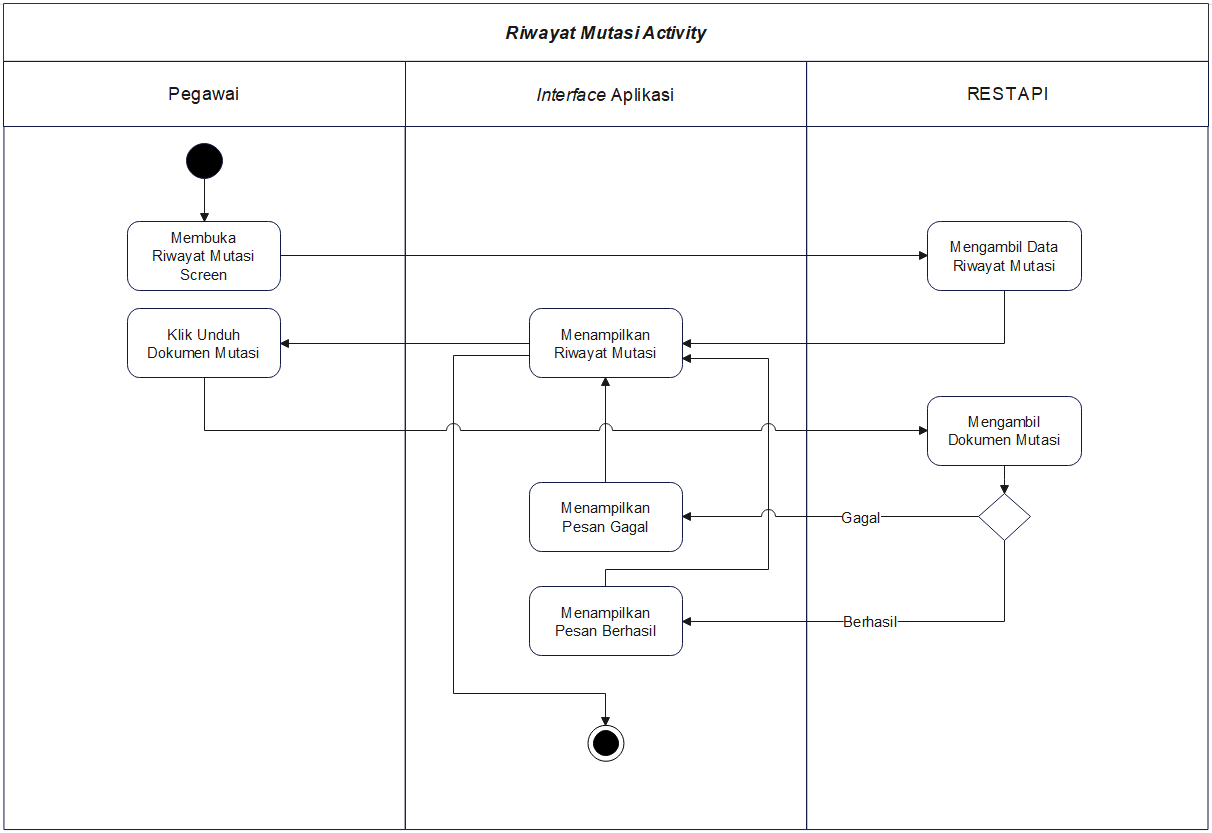
1. **Activity diagram riwayat pendidikan**



Gambar . *Activity* diagram riwayat pendidikan.

Pada *activity* diagram ini, proses akan dimulai ketika pegawai memuka halaman riwayat pendidikan. Sebelum data riwayat pendidikan ditampilkan *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data riwayat pendidikan berdasarkan pegawai, data yang 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.

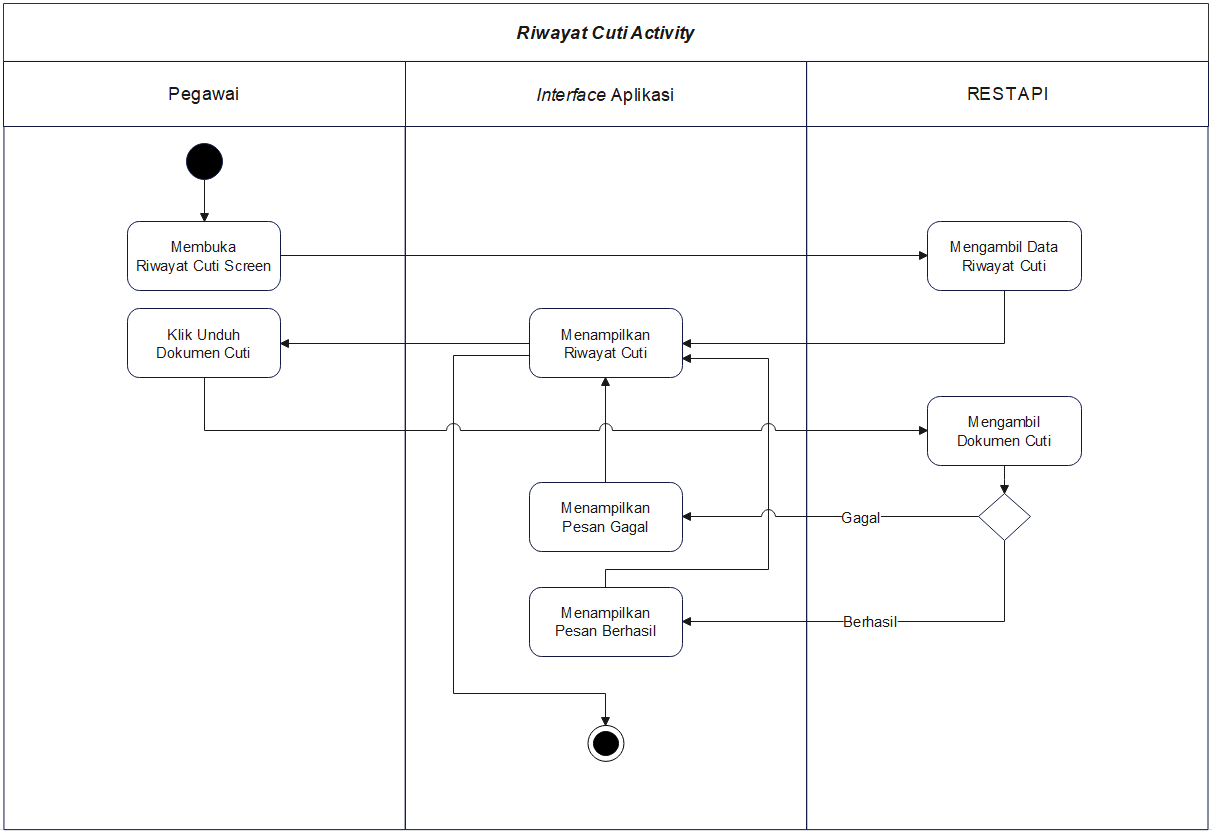
1. **Activity diagram riwayat mutasi**



Gambar . *Activity* diagram riwayat mutasi.

Pada *activity* diagram ini, proses akan dimulai ketika pegawai memuka halaman riwayat mutasi. Sebelum data riwayat mutasi ditampilkan *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data riwayat mutasi berdasarkan pegawai, data yang 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.

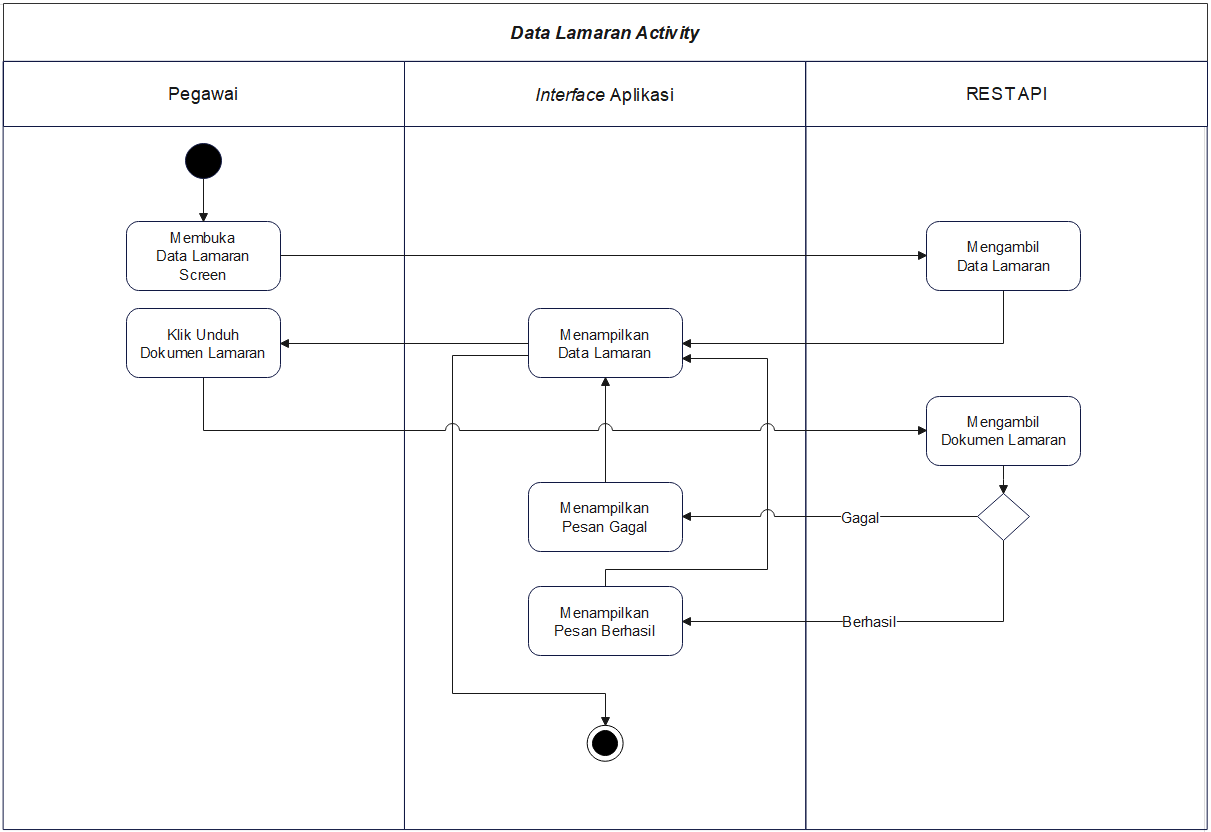
1. **Activity diagram riwayat cuti**



Gambar . *Activity* diagram riwayat cuti.

Pada *activity* diagram ini, proses akan dimulai ketika pegawai memuka halaman riwayat cuti. Sebelum data riwayat cuti ditampilkan *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data riwayat cuti berdasarkan pegawai, data yang 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.

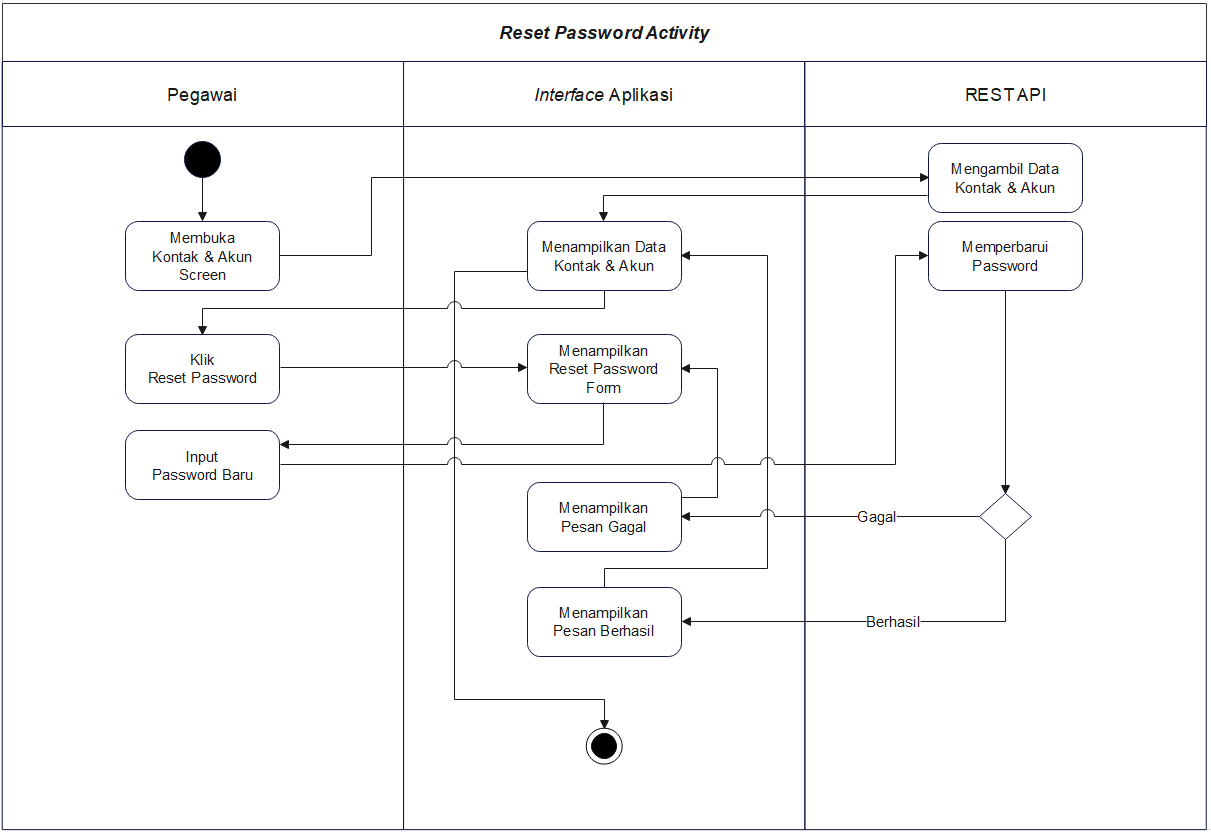
1. **Activity diagram data lamaran**



Gambar . *Activity* diagram data lamaran.

Pada *activity* diagram ini, proses akan dimulai ketika pegawai memuka halaman data lamaran. Sebelum data lamaran ditampilkan *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data lamaran berdasarkan pegawai, data yang 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.

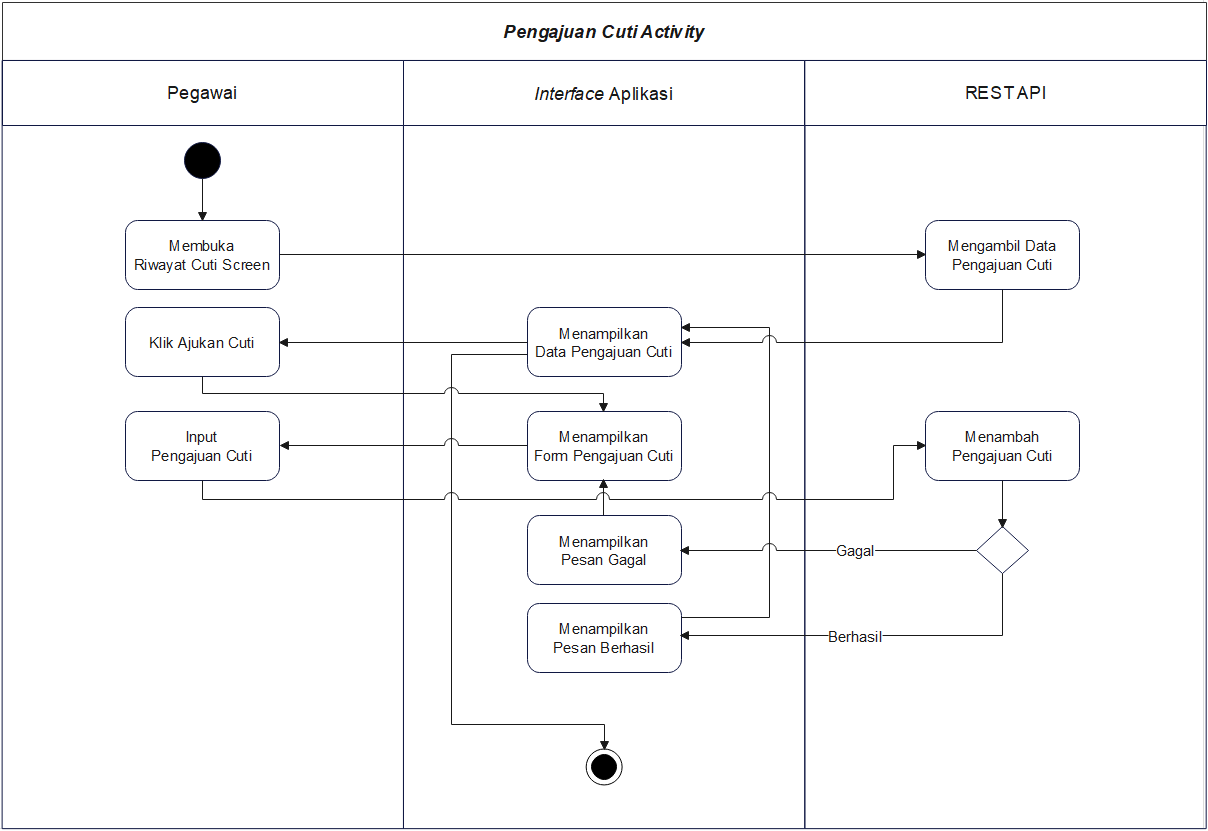
1. **Activity diagram reset password**



Gambar . *Activity* diagram *reset password.*

Pada *activity* diagram ini, proses akan dimulai ketika pegawai memuka halaman kontak & akun. Sebelum data kontak & akun ditampilkan *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data kontak & akun berdasarkan pegawai, data yang 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.

1. **Activity diagram pengajuan cuti**



Gambar . *Activity* diagram pengajuan cuti.

Pada *activity* diagram ini, proses akan dimulai ketika pegawai memuka halaman riwayat cuti. Sebelum data pengajuan cuti ditampilkan *interface* aplikasi, REST API mendapat permintaan untuk menyediakan data pengajuan cuti berdasarkan pegawai, data yang 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 cutibaru 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.

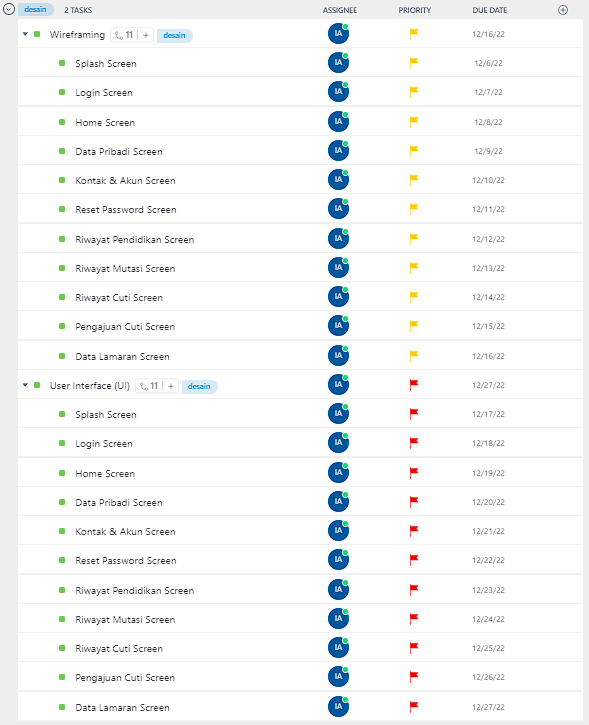
### Sprint

Pada tahap *sprint* semua proses perencanaan dalam tahap-tahap sebelumnya direalisasikan dan dikerjakan. Pada pengembangan sistem manajemen pegawai berbasis *mobile* ini, tahap *sprint* dilakukan sebanyak tiga kali dimulai dari mendesain tampilan aplikasi, pengembangan aplikasi, sampai dengan pengujian fitur aplikasi sebelum aplikasi di-*review­* pada tahap selanjutnya.

#### Desain Tampilan Aplikasi

1. **Wireframing**

Aplikasi didesain dengan mementingkan elemen-elemen dasar seperti *card, button, text, textfield, dsb.* tanpa mementingkan pewarnaan dan gaya rumit lainnya. Berikut ini daftar *item* pengerjaan dalam *sprint* desain tampilan aplikasi (*wireframing*) beserta lampiran hasilnya.



Gambar . Daftar *item sprint* desain tampilan aplikasi (*wireframing*).

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 w*ireframe* *interface splash screen.*



Gambar . *Wireframe interface splash screen.*

1. *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 . *Wireframe interface login screen*.

1. *Wireframe* *interface home screen*

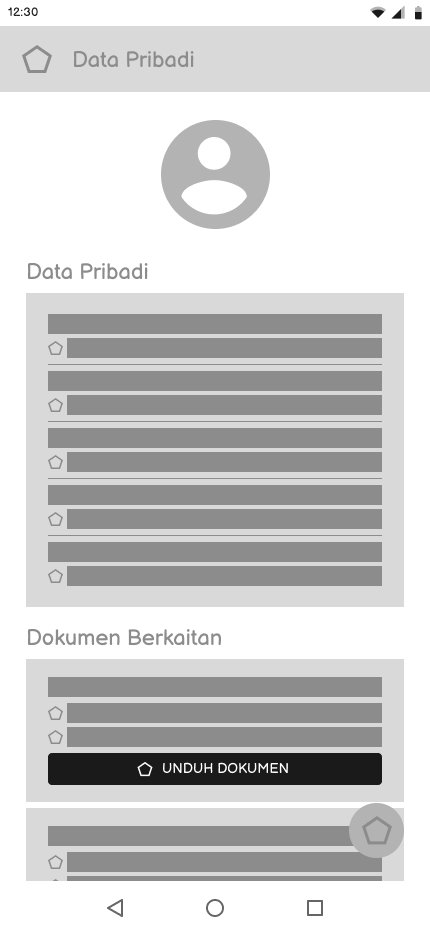
Halaman ini ditampilkan setelah halaman *splash* apabila pegawai memiliki *login session* dan ditampilkan setelah halaman *login* apabila pegawai belum memiliki *login session*. Pada halaman ini terdapat beberapa komponen utama seperti *card* profil, *list* menu, *card* mutasi terbaru*, card* cuti terbaru,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



Gambar . *Wireframe interface home screen*.

1. *Wireframe* *interface* data pribadi *screen*

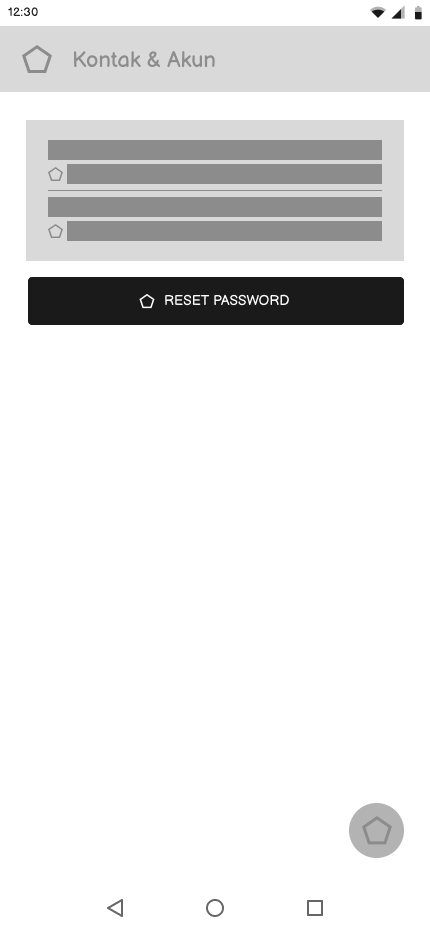
Halaman ini ditampilkan apabila pegawai menekan menu data pribadi pada komponen *list* menu yang ada di halaman *home*. Pada halaman terdapat beberapa komponen yaitu foto profil, *card* detail data pribadi, dan *list* dokumen data pribadi. Berikut ini tampilan *wireframe* *interface* data pribadi *screen.*



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

1. *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 . *Wireframe interface* kontak & akun *screen*.

1. *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* barudan konfirmasi *password*, serta *button reset* dan batal. Berikut ini tampilan *wireframe* *interface reset password screen*.



Gambar . *Wireframe interface reset password screen*.

1. *Wireframe* *interface* riwayat pendidikan *screen*

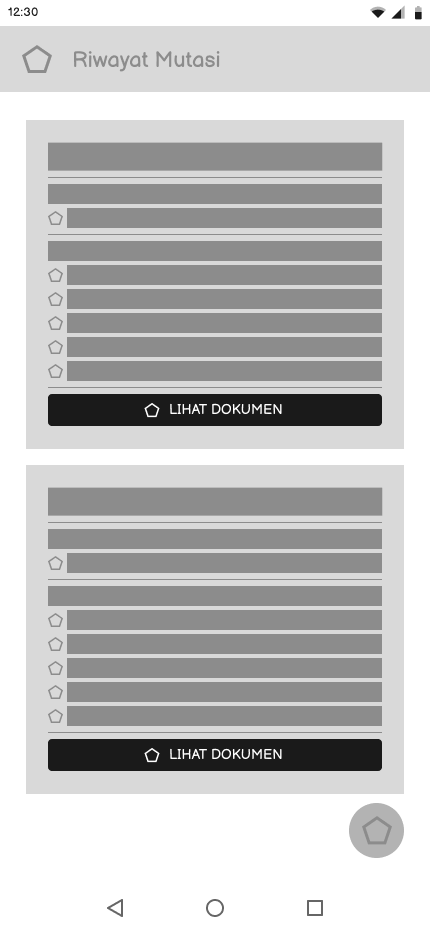
Halaman ini ditampilkan apabila pegawai menekan menu riwayat pendidikan pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat komponen *card* detail riwayat pendidikan. Berikut ini tampilan *wireframe* *interface* riwayat pendidikan *screen.*



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

1. *Wireframe* *interface* riwayat mutasi *screen*

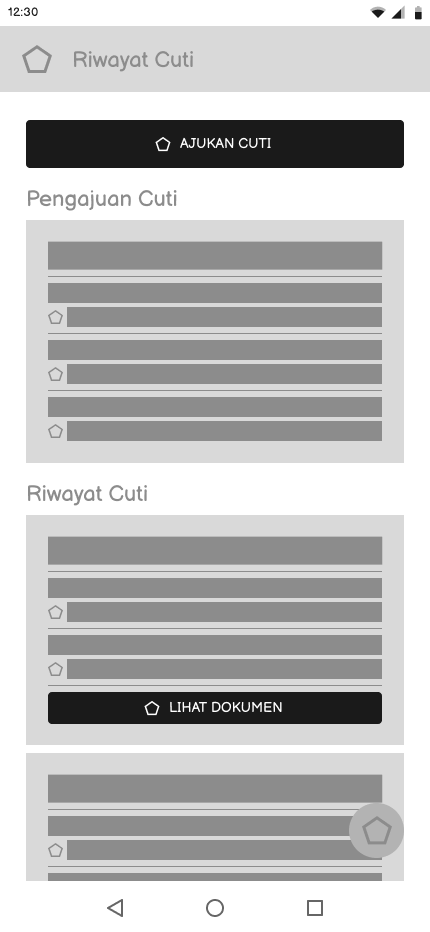
Halaman ini ditampilkan apabila pegawai menekan menu riwayat mutasi pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat komponen *card* detail riwayat mutasi. Berikut ini tampilan *wireframe* *interface* riwayat mutasi *screen.*



Gambar . *Wireframe interface* riwayat mutasi *screen*.

1. *Wireframe* *interface* riwayat cuti *screen*

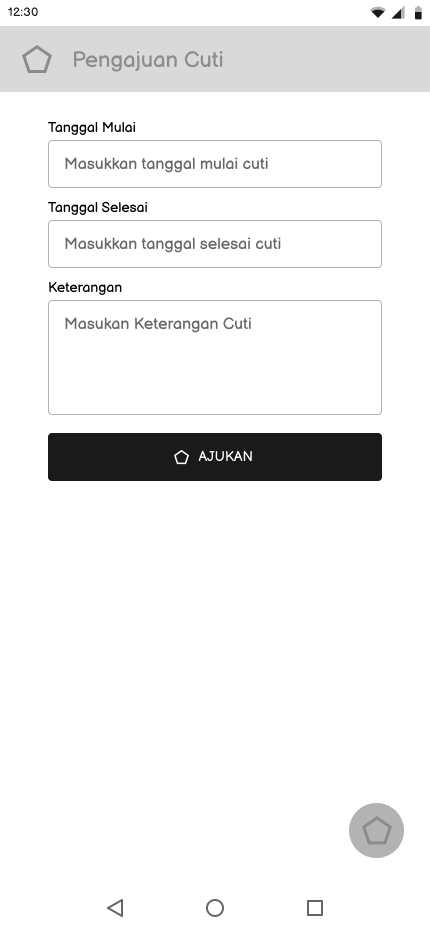
Halaman ini ditampilkan apabila pegawai menekan menu riwayat cuti pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat beberapa komponen yaitu *button* ajukan cuti, *card* detail pengajuan cuti, *card*, dan detail riwayat cuti. Berikut ini tampilan *wireframe* *interface* riwayat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



Gambar . *Wireframe interface* riwayat cuti *screen*.

1. *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 . *Wireframe interface* pengajuan cuti *screen*.

1. *Wireframe* *interface* data lamaran *screen*

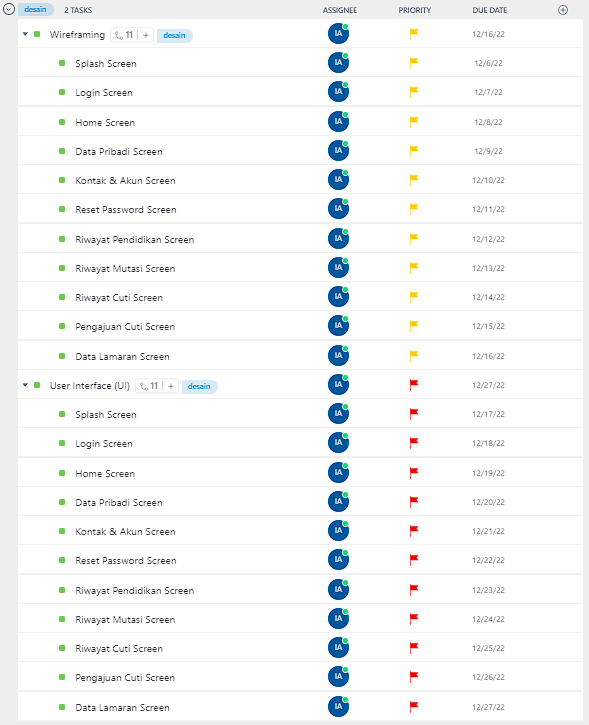
Halaman ini ditampilkan apabila pegawai menekan menu data lamaran pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat komponen *card* detail data lamaran. Berikut ini tampilan *wireframe* *interface* data lamaran *screen.*



Gambar . *Wireframe interface* data lamaran *screen*.

1. **Desain user interface (UI)**

Desain *user interface* aplikasi dilakukan dengan panduan *wireframing* yang sebelumnya sudah dibuat. Pada tahap ini warna, *icon, typography*, ilustrasi, ukuran, dan gaya rumit lainnya sangat diperhatikan sehingga menghasilkan tampilan yang menarik ketika dipandang pegawai. Berikut ini daftar *item* pengerjaan dalam *sprint* desain tampilan aplikasi (*user interface*) beserta lampiran hasilnya.



Gambar . Daftar *item sprint* desain tampilan aplikasi (*user interface*).

1. Desain *user 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 desain *user* *interface splash screen.*



Gambar . Desain *user interface splash screen.*

1. Desain *user 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 desain *user* *interface login* *screen* dalam dua tema aplikasi*.*



Gambar . Desain *user interface login screen.*

1. Desain *user 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 desain *user* *interface home screen* dalam dua tema aplikasi.



Gambar . Desain *user interface home screen*.

1. Desain *user 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 desain *user* *interface* data pribadi *screen* dalam dua tema aplikasi*.*



Gambar . Desain *user interface* data pribadi *screen*.

1. Desain *user 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 desain *user interface* kontak & akun *screen* dalam dua tema aplikasi.



Gambar . Desain *user interface* kontak & akun *screen*.

1. Desain *user 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* barudan konfirmasi *password*, serta *button reset* dan batal. Berikut ini tampilan desain *user interface reset password screen* dalam dua tema aplikasi.



Gambar . Desain *user interface reset password screen*.

1. Desain *user 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 desain *user* *interface* riwayat pendidikan *screen* dalam dua tema aplikasi*.*



Gambar . Desain *user* *interface* riwayat pendidikan *screen*.

1. Desain *user 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 desain *user* *interface* riwayat mutasi *screen* dalam dua tema aplikasi*.*



Gambar . Desain *user interface* riwayat mutasi *screen*

1. Desain *user 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 desain *user* *interface* riwayat cuti *screen* dalam dua tema aplikasi*.*



Gambar . Desain *user interface* riwayat cuti *screen*.

1. Desain *user 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 desain *user* *interface* pengajuan cuti *screen* dalam dua tema aplikasi*.*



Gambar . Desain *user interface* pengajuan cuti *screen*.

1. Desain *user interface* data lamaran *screen*

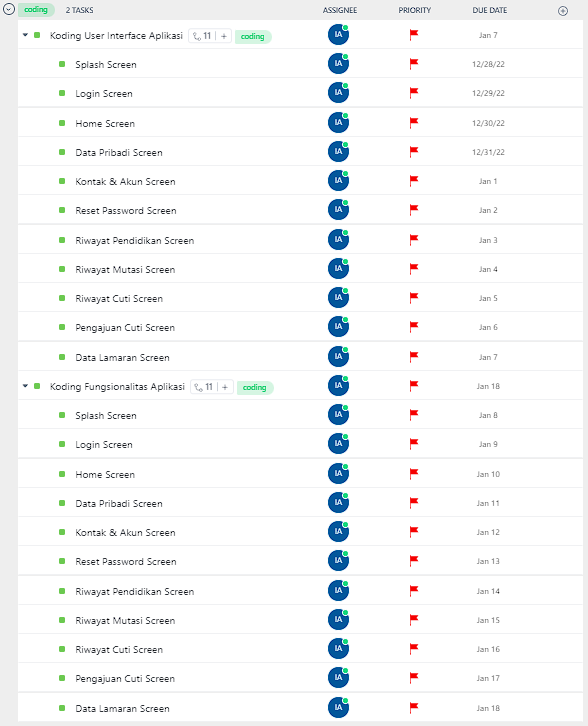
Halaman ini ditampilkan apabila pegawai menekan menu data lamaran pada komponen *list* menu yang ada di halaman *home*. Pada halaman ini terdapat komponen *card* detail data lamaran. Berikut ini tampilan desain *user* *interface* data lamaran *screen* dalam dua tema aplikasi*.*



Gambar . Desain *user interface* pengajuan cuti *screen*.

#### Pengembangan Aplikasi

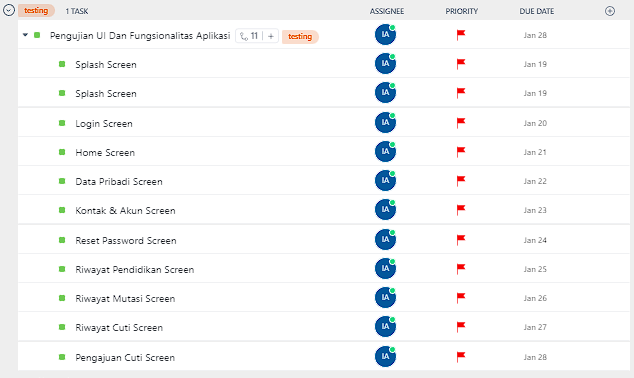
Pengembangan aplikasi dilakukan dengan mengimplementasikan desain menjadi sebuah aplikasi dan menyediakan fitur-fitur yang diinginkan pengguna. Proses pengembangan dilakukan dengan mengetikkan kode menggunakan kode editor (*coding*) lalu hasil sementara aplikasi di tampilkan sebagai bentuk pengujian kode yang baru saja diketik. Berikut ini daftar *item* pengerjaan dalam *sprint* pengembangan aplikasi.



Gambar . Daftar *item sprint* pengembangan aplikasi.

#### Pengujian Fitur Aplikasi

Pengujian fitur aplikasi dilakukan guna menguji apakah hasil pengembangan aplikasi telah berfungsi secara tampilan maupun fungsionalitasnya. Dalam *sprint* pengujian fitur aplikasi, setiap kekurangan baik dalam segi tampilan maupun fungsionalitas dilihat dan langsung diperbaiki kembali sebelum aplikasi di ulas oleh PT XYZ. Berikut ini daftar *item* pengerjaan dalam *sprint* pengujian fitur aplikasi.



Gambar . Daftar *item sprint* pengujian fitur aplikasi.

### Review dan Retrospective

#### Analisis Aspek Ketertarikan

#### Analisis Kompleksitas Aplikasi

#### Analisis Aspek Efektivitas Aplikasi

#### Analisis Aspek Teknis Aplikasi

#### Analisis Aspek Fitur-fitur Aplikasi

#### Analisis Aspek Penggunaan Keseharian Aplikasi

#### Analisis Aspek Pemahaman Pengguna

#### Analisis Aspek Kejelasan Sistem

#### Analisis Aspek Kelancaran Aplikasi

#### Analisis Aspek Penguasaan Aplikasi

#### Hasil Analisis Keseluruhan

#### Building for Production

# DAFTAR PUSTAKA

Adam, S. I., Mononutu, M. J., & Damping, G. (2022). Aplikasi Jasa Titip Belanja Berbasis Mobile di Minahasa Utara. *CogITo Smart Journal*, *8*(2), 434–445. https://doi.org/10.31154/cogito.v8i2.422.434-445

Aziz, N., Pribadi, G., & Nurcahya, M. S. (2020). Analisa dan Perancangan Aplikasi Pembelajaran Bahasa Inggris Dasar Berbasis Android. *Jurnal IKRAITH-INFORMATIKA*, *1*(3).

Boduch, A., Derks, R., & Sakhniuk, M. (2022). *React and React Native* (H. Edwards, Ed.; 4 ed.). Packt Publishing.

Chatterjee, S., & T, M. (2020). A comparative study on SOAP and RESTful web services. *IRJET*, *7*(5).

Eisenman, B. (2015). *Learning React Native: Building Mobile Applications with JavaScript* (M. Foley, Ed.; 1 ed.). O’Reilly Media, Inc.

Fitriyana, I., & Susianto, D. (2018). Aplikasi Akuntansi Piutang Jasa Service Pada PT. AUX Indonesia Bandar Lampung. *Jurnal JUSINTA*, *1*(1).

Flanagan, D. (2020). *JavaScript: The Definitive Guide* (J. Pollock, Ed.; 7 ed.). O’Reilly Media, Inc.

Fox, R. (2013). *Information Technology: An Introduction for Today’s Digital World* (1 ed.). Chapman and Hall/CRC.

Hendriyani, M. (2021). Pemberkasan Arsip Dinamis Aktif di Subbagian Persuratan Dan Arsip Aktif pada Arsip Nasional Republik Indonesia (ANRI). *Kompleksitas*, *10*(1). https://doi.org/10.56486/kompleksitas.vol10no01.80

Hutahaean, J., Purba, R. A., Siagian, Y., Heriyani, N., Amina, H. U. st., Syah, A. Z., Ardiana, D. P. Y., & Simarmata, J. (2021). *Pengantar Sistem Informasi Manajemen* (A. Rikki, Ed.; 1 ed.). Yayasan Kita Menulis.

IBM Cloud Education. (2022). *What is an Application Programming Interface (API)*. IBM. https://www.ibm.com/topics/api

Islam, Md. R., Islam, Md. R., & Mazumder, T. A. (2010). Mobile Application and Its Global Impact. *IJET-IJENS*, *10*(06).

Karim, M. A., & Adriansyah, A. R. (2022). Analisis dan Perancangan Aplikasi Mobile untuk Donasi menggunakan Metode Hybrid berbasis React Native. *Jurnal Informatika Terpadu*, *8*(1), 26–34. https://doi.org/10.54914/jit.v8i1.394

MDN Mozilla. (2022). *HTTP request methods*. Mozilla Foundation. https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods

Muhammad Nur Fauzi, A., Triayudi, A., & Diana Sholihati, I. (2022). MENGUKUR TINGKAT KEPUASAN PENGGUNA APLIKASI KEARSIPAN MENGGUNAKAN SYSTEM USABILITY SCALE DAN PIECES FRAMEWORK. *JIPI*, *7*(1), 231–239.

Mukodimah, S., Muslihudin, M., & Trisnawati. (2019). APLIKASI PENENTUAN BENGKEL TSM BERKUALITAS UNTUK UKK SISWA SMK KABUPATEN PRINGSEWU BERBASIS MOBILE. *SINTAK*, *3*.

Nursaid, F. F., Hendra Brata, A., & Kharisma, A. P. (2020). Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri). *J-Ptiik.Ub.Ac.Id*, *4*(1).

Prastio, C. E., & Ani, N. (2018). Aplikasi Self Service Menu Menggunakan Metode Scrum Berbasis Android (Case Study : Warkobar Café Cikarang). *PETIR*, *11*(2), 203–220. https://doi.org/10.33322/petir.v11i2.255

Putra, D. W. T., & Andriani, R. (2019). Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD. *Jurnal TeknoIf*, *7*(1), 32. https://doi.org/10.21063/jtif.2019.V7.1.32-39

Rosalin, S. (2017). *Manajemen Arsip Dinamis* (1 ed.). UB Press.

Sadikin, A., & Wiranda, N. (2022). *Sistem Informasi Manajemen* (I. Mirsa, Ed.; 1 ed.). K-Media. http://digilib.iain-palangkaraya.ac.id/3890/

Schwaber, K., & Sutherland, J. (2020, November). *The 2020 Scrum Guide*. Scrum Guides. https://scrumguides.org/scrum-guide.html

Sugiarto, A., & Wahyono, T. (2014). *Manajemen Kearsipan Elektronik* (1 ed.). Gava Media.

Suparman. (2020). Pelaksanaan Manajemen Arsip dalam Meningkatkan Efektivitas Kerja Pegawai pada Kantor Camat Kecamatan Tanjung Lago, Kabupaten Banyuasin. *Jurnal Studia Administrasi*, *2*(2), 42–57. https://doi.org/10.47995/jian.v2i2.13

Visual Paradigm. (2022a). *What is Activity Diagram?* https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/

Visual Paradigm. (2022b). *What is Use Case Diagram?* https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/

Wibisono, W., & Baskoro, F. (2002). PENGUJIAN PERANGKAT LUNAK DENGAN MENGGUNAKAN MODEL BEHAVIOUR UML. *JUTI: Jurnal Ilmiah Teknologi Informasi*, *1*(1). https://doi.org/10.12962/j24068535.v1i1.a95