

PERANCANGAN APLIKASI PRESENSI BERBASIS MOBILE PADA PDAM MUARO JAMBI

TUGAS AKHIR



Disusun oleh :

Rama Rifki Fadli

8020200003

Untuk Persyaratan Penelitian dan Penulisan Tugas Akhir

Sebagai akhir proses studi Strata 1

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DINAMIKA BANGSA JAMBI

2025

LEMBAR PENGESAHAN PEMBIMBING

Judul Tugas Akhir : Perancangan Aplikasi Presensi Berbasis Mobile Pada
PDAM Muaro Jambi
Nama : Rama Rifki Fadli
NIM : 8020200003
Tanggal Sidang :

Mengesahkan :

Pembimbing I

Pembimbing II

(Dr.Nurhadi, S.Kom, M.Cs)
NIK : YDB.02.77.006

(Eni Rohaini , S.Kom, M.Kom)
NIK : YDB.11.87.076

Ka. Prodi Teknik Informatika

(Marrylinteri Istoningtyas, ST, M.Kom)
NIK : YDB.16.82.108

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini :

Nama : Rama Rifki Fadli
NIM : 8020200003
Judul Tugas Akhir : Perancangan Aplikasi Presensi Berbasis Mobile
Pada PDAM Muaro Jambi

Menyatakan dengan sesungguhnya bahwa Tugas Akhir ini merupakan hasil penelitian, pemikiran dan pemaparan asli saya sendiri. Saya tidak mencantumkan tanpa pengakuan bahan-bahan yang telah dipublikasikan sebelumnya atau ditulis orang lain, atau sebagai bahan yang pernah diajukan untuk gelar atau ijazah pada Universitas Dinamika Bangsa.

Apabila kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai dengan peraturan yang berlaku di Universitas Dinamika Bangsa.

Demikian pernyataan ini saya buat.

Jambi, Februari 2025

Yang membuat pernyataan,

Rama Rifki Fadli

ABSTRAK

Rama Rifki Fadli, 8020200003, Teknik Informatika
Perancangan Aplikasi Presensi Berbasis Mobile Pada PDAM Muaro Jambi, 2025

Kata Kunci : Perancangan, Aplikasi, Presensi, Pegawai, Mobile.

(xii+98+Lampiran)

PDAM Muaro Jambi dalam Pengelolaan presensi secara konvensional dengan mencatat pada lembar absensi, akan membutuhkan waktu yang lama karena harus satu per satu data dimasukkan. Rentan juga adanya manipulasi data atau terjadi data yang ganda, sehingga tidak mendapatkan hasil yang akurat. Cara ini sangatlah rawan bagi suatu instansi karena tingkat kedisiplinan yang tidak dapat di kontrol dan dapat disalah gunakan oleh orang yang tidak bertanggung jawab. Tujuan penelitian ini Mempelajari dan menganalisa pendataan Presensi pada PDAM Muaro Jambi serta merancang sebuah aplikasi Presensi berbasis mobile pada PDAM Muaro Jambi yang akan memudahkan pegawai untuk melakukan presensi dengan memanfaatkan aplikasi berbasis mobile. Peneliti melakukan pengembangan sistem dengan metode Waterfall. Hasil dari penelitian ini pegawai PDAM Muaro Jambi tidak perlu lagi mengisi absensi secara manual karena semua data absensi tercatat secara otomatis dan terintegrasi. Dapat mempermudah dalam melakukan presensi menggunakan mobile dan melihat presensi serta mempermudah dalam pembuatan laporan selama periode perbulan.

Daftar Pustaka (2014-2024)

ABSTRACT

Rama Rifki Fadli, 8020200003, *Informatics Engineering*
Design of a Mobile Based Presence Application for PDAM Muaro Jambi, 2025

Keywords: Design, Application, Presence, Employees, Mobile.

(xii+98+Attachments)

PDAM Muaro Jambi in managing attendance conventionally by recording it on the attendance sheet, will take a long time because the data must be entered one by one. It is also vulnerable to data manipulation or duplicate data occurring, resulting in not getting accurate results. This method is very vulnerable for an agency because the level of discipline cannot be controlled and can be misused by irresponsible people. The aim of this research is to study and analyze attendance data collection at PDAM Muaro Jambi and design a mobile-based attendance application at PDAM Muaro Jambi which will make it easier for employees to take attendance by utilizing a mobile-based application. Researchers carried out system development using the Waterfall method. The results of this research are that PDAM Muaro Jambi employees no longer need to fill in attendance manually because all attendance data is recorded automatically and integrated. Can make it easier to take attendance using mobile and view attendance and make it easier to make reports during the monthly period.

Bibliography (2014-2024)

KATA PENGANTAR

Puji Syukur penulis ucapkan atas kehadiran Tuhan Yang Maha Esa yang telah melimpahkan rahmatnya dengan ilmu pengetahuan sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul **“Perancangan Aplikasi Presensi Berbasis Mobile Pada PDAM Muaro Jambi”** tepat pada waktunya. Penulis juga mengucapkan terima kasih yang sebesar-besarnya kepada seluruh pihak yang telah membantu dalam penulisan laporan ini, antara lain :

1. Bapak Dr.Ir. Eddy Anthony SH, MM selaku Ketua Yayasan Dinamika Bangsa Jambi.
2. Bapak Dr.Ir. Herry Mulyono, MM selaku Pembina Yayasan Dinamika Bangsa Jambi.
3. Bapak Prof. Setiawan Assegaf, ST, MMSI, Ph. D. selaku Rektor Universitas Dinamika Bangsa Jambi.
4. Bapak Beny, S.Kom, M.Sc selaku Dekan Fakultas Ilmu Komputer Universitas Dinamika Bangsa Jambi.
5. Ibu Marrylinteri Istoningtyas, ST, M.Kom selaku Ketua Program Studi Teknik Informatika Universitas Dinamika Bangsa Jambi.
6. Bapak Dr.Nurhadi, S.Kom, M.Cs selaku Pembimbing I Tugas Akhir yang telah banyak memberikan waktu untuk bimbingan, pengarahan dan saran dalam Penyusunan Laporan Tugas Akhir ini.
7. Ibu Eni Rohaini , S.Kom, M.Kom selaku Pembimbing II Tugas Akhir yang telah banyak memberikan waktu untuk bimbingan, pengarahan dan saran dalam Penyusunan Laporan Tugas Akhir ini.
8. Para Pegawai PDAM Muaro Jambi, yang telah memberikan kesempatan kepada penulis untuk melakukan penelitian Tugas Akhir Ini
9. Seluruh Staf dan Dosen Universitas Dinamika Bangsa Jambi yang telah banyak membantu sehingga penulisan laporan ini dapat terselesaikan dengan baik.
10. Kedua orang tua penulis, keluarga, dan sahabat yang turut memberikan dukungan, motivasi, dan doa dalam penulisan laporan Tugas akhir ini.

Dengan keterbatasan ilmu dan pengetahuan yang penulis miliki dalam penulisan laporan Tugas akhir ini, penulis menyadari bahwa masih terdapat kekurangan, baik dalam isi maupun cara penyajian materi. Oleh karena itu, penulis sangat mengharapkan saran dan kritik yang bersifat membangun. Akhir kata, penulis mengharapkan bahwa laporan Tugas akhir ini dapat bermanfaat bagi para pembaca dan semua pihak yang bersangkutan.

Jambi, Februari 2025

Rama Rifki Fadli

DAFTAR ISI

Halaman

COVER	i
LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PERNYATAAN KEASLIAN	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN.....	1
1.1 LATAR BELAKANG	1
1.2 RUMUSAN MASALAH.....	2
1.3 BATASAN MASALAH	3
1.4 TUJUAN DAN MANFAAT PENELITIAN	3
1.4.1 Tujuan Penelitian.....	3
1.4.2 Manfaat Penelitian.....	3
1.5 SISTEMATIKA PENULISAN	4
BAB II LANDASAN TEORI	7
2.1 PERANCANGAN.....	7
2.2 APLIKASI.....	8
2.3 PRESENSI	9
2.4 QR CODE.....	9
2.5 METODE WATERFALL	12
2.6 ALAT BANTU PEMODELAN SISTEM	13
2.6.1 UML (<i>Unified Modeling Language</i>).....	13
2.6.2 <i>Use case Diagram</i>	14
2.6.3 <i>Activity Diagram</i>	16
2.6.4 <i>Class Diagram</i>	17
2.7 ALAT BANTU PERANCANGAN PROGRAM	18
2.7.1 Android.....	18
2.7.2 Flutter	20
2.7.3 Rest API	21
2.7.4 Basis Data (<i>Database</i>).....	22
2.7.5 MySQL.....	22
2.7.6 LARAVEL	23
2.7.7 Visual Studio Code.....	24
2.7.8 Balsamiq Mockup	25
2.8 PENELITIAN SEJENIS	27
BAB III METODOLOGI PENELITIAN	33
3.1 KERANGKA KERJA PENELITIAN.....	33
3.2 METODE PENGEMBANGAN SISTEM	36
3.3 ALAT BANTU PENELITIAN	38
BAB IV ANALISIS DAN PERANCANGAN SISTEM	40

4.1	ANALISIS SISTEM	40
4.1.1	Gambaran Umum PDAM Muaro Jambi	40
4.1.2	Analisis Sistem Yang Berjalan.....	41
4.1.3	Kelemahan Sistem Yang Sedang Berjalan	41
4.1.4	Solusi Pemecahan Masalah	42
4.2	ANALISIS KEBUTUHAN PERANGKAT LUNAK	43
4.2.1	<i>Use Case Diagram</i>	43
4.2.2	Deskripsi <i>Use Case</i>	44
4.2.3	<i>Activity Diagram</i>	55
4.2.4	<i>Class Diagram</i>	67
4.3	RANCANGAN <i>INPUT</i> DAN <i>OUTPUT</i>	67
4.3.1	Rancangan <i>Input</i>	68
4.3.2	Rancangan <i>Output</i>	74
4.4	RANCANGAN STRUKTUR DATA	77
4.5	RANCANGAN STRUKTUR PROGRAM	79
BAB V	IMPLEMENTASI DAN PENGUJIAN	81
5.1	IMPLEMENTASI PROGRAM	81
5.1.1	Implementasi <i>Input</i>	81
5.1.2	Implementasi <i>Output</i>	87
5.2	PENGUJIAN SISTEM / PERANGKAT LUNAK	89
5.3	ANALISIS HASIL YANG DICAPAI OLEH SISTEM	94
5.3.1	Kelebihan Program.....	95
5.3.2	Kekurangan Program.....	95
BAB VI	PENUTUP	97
6.1	KESIMPULAN	97
6.2	SARAN	97
DAFTAR PUSTAKA		
LAMPIRAN		

DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol <i>Use case Diagram</i>	15
Tabel 2.2 Simbol <i>Activity Diagram</i>	16
Tabel 2.3 Simbol <i>Class Diagram</i>	18
Tabel 2.4 Versi <i>Android</i>	19
Tabel 2.5 Penelitian Sejenis	27
Tabel 4.1 Deskripsi <i>Use Case Login</i>	45
Tabel 4.2 Deskripsi <i>Use Case</i> Mengelola Data Admin	46
Tabel 4.3 Deskripsi <i>Use Case</i> Mengelola Data Pegawai	48
Tabel 4.4 Deskripsi <i>Use case</i> Melihat Data Presensi.....	50
Tabel 4.5 Deskripsi <i>Use Case</i> Melihat Laporan	51
Tabel 4.6 Deskripsi <i>Use Case</i> Presensi.....	51
Tabel 4.7 Deskripsi <i>Use Case</i> Izin/Sakit.....	52
Tabel 4.8 Deskripsi <i>Use Case</i> Melihat History Presensi	53
Tabel 4.9 Deskripsi <i>Use Case</i> Ubah <i>Password</i>	54
Tabel 4.10 Deskripsi <i>Use Case Logout</i>	54
Tabel 4.11 Rancangan Tabel Admin.....	78
Tabel 4.12 Rancangan Struktur Tabel Jabatan.....	78
Tabel 4.13 Rancangan Struktur Tabel Pegawai	78
Tabel 4.14 Rancangan Struktur Tabel Presensi	79
Tabel 5.1 Pengujian Modul <i>Login</i> Admin.....	90
Tabel 5.2 Pengujian Modul Mengelola Data Admin	91
Tabel 5.3 Pengujian Modul Mengelola Data Pegawai	92
Tabel 5.4 Pengujian Modul Mengelola Data Presensi	93

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Anatomi QR Code	11
Gambar 2.2 Metode <i>Waterfall</i>	12
Gambar 2.3 Gambar Aplikasi Balsamiq	26
Gambar 3.1 Kerangka Kerja Penelitian	33
Gambar 3.2 Model <i>Waterfall</i>	37
Gambar 4.1 Sistem Yang Sedang Berjalan	41
Gambar 4.2 Diagram <i>Use Case</i>	44
Gambar 4.3 <i>Activity Diagram Login</i>	55
Gambar 4.4 <i>Activity Diagram</i> Menambah Data Admin.....	56
Gambar 4.5 <i>Activity Diagram</i> Mengubah Data Admin	57
Gambar 4.6 <i>Activity Diagram</i> Menghapus Data Admin.....	58
Gambar 4.7 <i>Activity Diagram</i> Menambah Data Pegawai	59
Gambar 4.8 <i>Activity Diagram</i> Mengubah Data Pegawai	60
Gambar 4.9 <i>Activity Diagram</i> Menghapus Data Pegawai	61
Gambar 4.10 <i>Activity Diagram</i> Melihat Data Presensi.....	62
Gambar 4.11 <i>Activity Diagram</i> Melihat Laporan	62
Gambar 4.12 <i>Activity Diagram</i> Mengelola Data Presensi	63
Gambar 4.13 <i>Activity Diagram</i> Mengelola Data Izin/Sakit	64
Gambar 4.14 <i>Activity Diagram</i> Melihat History Presensi	65
Gambar 4.15 <i>Activity Diagram</i> Ubah <i>Password</i>	66
Gambar 4.16 <i>Activity Diagram Logout</i>	66
Gambar 4.17 <i>Class Diagram</i>	67
Gambar 4.18 <i>Input Dan Output</i> Data Presensi.....	68
Gambar 4.19 Rancangan Form <i>Login</i> Pegawai.....	69
Gambar 4.20 Rancangan Halaman Scan Presensi.....	69
Gambar 4.21 Rancangan Halaman Izin/Sakit	70
Gambar 4.22 Rancangan Halaman Data Rekap Presensi.....	71
Gambar 4.23 Rancangan Halaman Ubah <i>Password</i>	71
Gambar 4.24 Rancangan Halaman Qr Code Presensi.....	72
Gambar 4.25 Rancangan Halaman <i>Login</i> Admin	72
Gambar 4.26 Rancangan Halaman Form Data Pegawai.....	73
Gambar 4.27 Rancangan Halaman Form Data Pengguna.....	73
Gambar 4.28 Rancangan Halaman Form Presensi.....	74
Gambar 4.29 Rancangan Halaman Mengelola Pegawai	75
Gambar 4.30 Rancangan Halaman Data Pengguna	75
Gambar 4.31 Rancangan Halaman Data Presensi.....	76
Gambar 4.32 Rancangan Halaman Laporan Pegawai.....	76
Gambar 4.33 Rancangan Halaman Laporan Presensi	77
Gambar 4.34 Rancangan Struktur Program Pegawai.....	79
Gambar 4.35 Rancangan Struktur Program Admin	80
Gambar 5.1 Implementasi Form Login Pegawai	81
Gambar 5.2 Implementasi Halaman Data Presensi Masuk.....	82

Gambar 5.3 Implementasi Halaman Izin/Sakit	83
Gambar 5.4 Implementasi Halaman Data Rekap Presensi.....	83
Gambar 5.5 Implementasi Halaman Ubah <i>Password</i>	84
Gambar 5.6 Implementasi Halaman Qr Code Presensi.....	84
Gambar 5.7 Implementasi Halaman <i>Login</i> Admin	85
Gambar 5.8 Implementasi Halaman Form Data Pegawai.....	85
Gambar 5.9 Implementasi Halaman Form Data Pengguna.....	86
Gambar 5.10 Implementasi Halaman Form Presensi.....	86
Gambar 5.11 Implementasi Halaman Mengelola Pegawai	87
Gambar 5.12 Implementasi Halaman Data Pengguna	88
Gambar 5.14 Implementasi Halaman Data Presensi.....	88
Gambar 5.15 Implementasi Halaman Laporan Pegawai.....	89
Gambar 5.16 Implementasi Halaman Laporan Presensi	89

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Daftar Riwayat Hidup	L1
Lampiran 2 Lembar Bimbingan.....	L2
Lampiran 3 Surat Keterangan	L3
Lampiran 4 Transkrip Wawancara	L4
Lampiran 5 Foto Dokumentasi	L5
Lampiran 6 Listing Program.....	L6

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Teknologi informasi dan komunikasi telah menjadi bagian dari kehidupan manusia dan membawa manfaat serta kemajuan. Kebutuhan informasi yang cepat sangat dibutuhkan oleh masyarakat bahkan komunikasi yang cepat dan akurat juga sangat diperlukan untuk memberikan suatu data yang real khususnya sebuah instansi [1]. Akses yang cepat dan akurat itu dapat kita dapatkan dalam teknologi *mobile* yang saat ini sudah terkoneksi dengan *internet*. Aplikasi *mobile phone* yang berbasis android dapat digunakan untuk melakukan presensi pegawai. Sistem tersebut memudahkan karyawan untuk mengisi presensi ketika jam kerja dimulai dan pulang sesuai jadwal jam kerja, mendeteksi lokasi karyawan ketika mengisi presensi di sistem, dan menyimpan foto bukti lokasi. Sistem dapat menampilkan informasi kehadiran per akun [2].

PDAM Muaro Jambi yang beralamat di Mendalo Darat, Kec. Jambi Luar Kota, Kabupaten Muaro Jambi, Jambi 36361 Bidang Usaha Pengelola Air Bersih. PDAM Tirta Muaro Jambi merupakan bagian dari pemekaran PDAM Tirta Batanghari Kabupaten Batanghari yang semula bernama BPAMD Kabupaten Muaro Jambi yang terbentuk berdasarkan Keputusan Bupati Muaro Jambi Nomor : 24 tahun 2002. Pembentukan dan pendirian PDAM Tirta Muaro Jambi berdasarkan PERDA Muaro Jambi No. 07 Tahun 2003 tanggal 13 Maret 2003.

PDAM Muaro Jambi dalam Pengelolaan presensi secara konvensional dengan mencatat pada lembar absensi, akan membutuhkan waktu yang lama karena harus satu per satu data dimasukkan. Rentan juga adanya manipulasi data atau terjadi data yang ganda, sehingga tidak mendapatkan hasil yang akurat. Cara ini sangatlah rawan bagi suatu instansi karena tingkat kedisiplinan yang tidak dapat di kontrol dan dapat disalah gunakan oleh orang yang tidak bertanggung jawab, kerugian lain yang mungkin muncul pada sistem presensi manual adalah rekapitulasi data yang masih memakan banyak waktu dan tenaga.

Pemanfaatan sistem teknologi informasi yang semakin maju dapat mengatasi hal-hal yang tidak diinginkan diatas dengan adanya sistem absensi yang digitalisasi dengan memanfaatkan aplikasi berbasis mobile android. Perancangan sistem presensi yang lebih modern dengan memperhatikan sisi kejujuran memang perlu diadakan sebagai pengganti presensi manual dengan menggunakan kertas sekaligus dapat menjadi standar baru sistem presensi karyawan. Demi meningkatkan kinerja karyawan, perusahaan membutuhkan sistem informasi yang terkomputerisasi sehingga dapat merekap informasi dan berguna [3]

Berdasarkan uraian permasalahan dan solusi yang ditawarkan maka peneliti tertarik untuk merancang sistem yang baik, yang penulis tuangkan dalam penelitian yang berjudul **“Perancangan Aplikasi Presensi Berbasis Mobile Pada PDAM Muaro Jambi”**.

1.2 RUMUSAN MASALAH

Berdasarkan latar belakang masalah di atas maka penulis dapat merumuskan rumusan masalah dari pembahasan tersebut, yaitu :

1. Bagaimana menganalisis sistem Presensi yang sedang berjalan Pada PDAM Muaro Jambi.
2. Bagaimana merancang Suatu Aplikasi Presensi pada PDAM Muaro Jambi yang dapat memudahkan dalam proses Presensi pegawai.

1.3 BATASAN MASALAH

Dalam penelitian ini akan dibatasi pada hal-hal berikut :

1. Penelitian ini hanya membahas tentang hal-hal yang berkaitan dengan data Presensi Pada PDAM Muaro Jambi.
2. Metode permodelan sistem yang dibuat menggunakan UML (*Unified Modeling Language*) yaitu : *use case diagram*, *class diagram*, *activity diagram*.
3. Pengembangan sistem menggunakan metode *Waterfall*.
4. *Software* yang digunakan dalam membangun aplikasi adalah Android Studio untuk pembuatan aplikasi android.

1.4 TUJUAN DAN MANFAAT PENELITIAN

1.4.1 Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat aplikasi presensi berbasis mobile sebagai berikut :

1. Mempelajari dan menganalisa pendataan Presensi pada PDAM Muaro Jambi.
2. Merancang sebuah aplikasi Presensi berbasis mobile pada PDAM Muaro Jambi yang akan memudahkan pegawai untuk melakukan presensi dengan memanfaatkan aplikasi berbasis mobile.

1.4.2 Manfaat Penelitian

Adapun manfaat secara umum yang diperoleh dari penelitian ini adalah:

1. Membantu PDAM Muaro Jambi untuk dapat monitoring absensi, sakit, izin, melalui aplikasi *mobile* yang digunakan pegawai.
2. Bagi Pegawai, dapat mempermudah dalam melakukan presensi menggunakan *mobile* dan melihat presensi yang telah dilakukan selama periode perbulan ataupun periode tahunan.
3. Dapat mengembangkan wawasan keilmuan dan meningkatkan pemahaman tentang struktur dan sistem kerja dalam pengembangan aplikasi pada sistem operasi Android.

1.5 SISTEMATIKA PENULISAN

Sistematika penulisan ini guna memberikan gambaran secara umum mengenai keseluruhan bab yang saling berhubungan satu sama lainnya dan sesuai dengan ruang lingkup judul serta menghindari terjadinya pembahasan diluar dari tema dan judul penelitian.

BAB I : PENDAHULUAN

Bab ini merupakan pengantar terhadap masalah yang sedang dibahas seperti Latar Belakang Masalah, Rumusan Masalah, Batasan Masalah, Tujuan dan Manfaat Penelitian serta Sistematika Penulisan dalam penyusunan laporan.

BAB II : LANDASAN TEORI

Pada bab ini berisi konsep teori-teori dasar yang mendukung penulisan, dikutip dari buku, jurnal, dan lain-lain seperti perancangan, aplikasi, absensi, *Waterfall*, *Unifed Modeling Language* (UML), *Use case Diagram*, Flutter, MySQL, Kotlin, MySQL, Android Studio, Balsamiq dan penelitian sejenis.

BAB III : METODOLOGI PENELITIAN

Dalam bab ini akan dijelaskan tahapan proses yang dilakukan selama mengerjakan penelitian, metode yang digunakan dan *tools* yang digunakan untuk mengembangkan program baik *software* maupun *hardware* pada perancangan aplikasi *mobile* berbasis mobile android.

BAB IV : ANALISA DAN RANCANGAN SISTEM

Bab ini menjelaskan tentang apa yang akan dilakukan terhadap objek dan hasil analisa, antara lain Analisa Sistem, Perancangan Sistem, Rancangan *Interface*, Implementasi dan *Use Case* yang ditemukan dalam penelitian serta menguraikan hasil penelitian yang mencakup semua aspek yang terkait dengan penelitian.

BAB V : IMPLEMENTASI DAN PENGUJIAN

Dalam bab ini diuraikan mengenai implementasi sistem dari objek penelitian serta pengujian sistem baik itu kelebihan maupun kekurangan dari sistem.

BAB VI : PENUTUP

Bab ini menguraikan kesimpulan yang diperoleh dari proses-proses perancangan dan implementasi sistem yang telah dilakukan serta mengemukakan saran-saran bagi pembangunan dan pengembangan yang akan datang.

BAB II

LANDASAN TEORI

2.1 PERANCANGAN

Perancangan diperlukan untuk menentukan bagaimana suatu aplikasi akan menyelesaikan apa yang harus diselesaikan sehingga menghasilkan aplikasi yang stabil dan mudah digunakan. Berikut adalah pandangan mengenai rancangan menurut para ahli:

Santi [4] “Perancangan adalah suatu kegiatan membuat desain teknis berdasarkan evaluasi yang telah dilakukan pada kegiatan analisis”.

Hendri [5] “Perancangan adalah penggambaran, perencanaan, dan pembuatan sketsa atas pengaturan dari beberapa elemen terpisah dan suatu kesatuan yang utuh dan berfungsi”.

Solihin dan Nusa [6] menyatakan “Perancangan adalah suatu kegiatan yang memiliki tujuan untuk mendesign sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik”.

Dari definisi diatas dapat disimpulkan bahwa perancangan sistem adalah kegiatan yang memiliki tujuan untuk mendesain sistem yang baru untuk menyelesaikan masalah yang dihadapi oleh perusahaan dengan memberikan gambaran yang jelas dan rancang bangun yang lengkap.

2.2 APLIKASI

Aplikasi merupakan penerapan, pengimplementasian suatu hal, data, permasalahan dan pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru, tanpa menghilangkan nilai-nilai dasar dari hal, data, permasalahan atau pekerjaan, dan terdapat berbagai pengertian perancangan dari beberapa ahli, diantaranya :

Menurut Setyawan dan Munari [7] “Aplikasi merupakan suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna”.

Menurut Harmayani [8] “Perangkat lunak atau Aplikasi merupakan Kumpulan data elektronik yang tersimpan dan diatur oleh komputer, bisa berupa program ataupun koneksi untuk menjalankan berbagai macam instruksi perintah”.

Kamus Kamus Besar Bahasa Indonesia (KBBI) [9] “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu”.

Berdasarkan uraian di atas, dapat disimpulkan bahwa aplikasi merupakan satu unit perangkat lunak atau alat terapan yang dibuat untuk mengerjakan tugas tugas khusus bagi user atau pengguna.

Klasifikasi aplikasi dapat dibagi menjadi 2 (dua) yaitu:

1. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.

2. Aplikasi paket, dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

2.3 PRESENSI

Presensi digunakan untuk memantau dan mengelola data kehadiran, sedangkan absensi digunakan untuk melaporkan ketidakhadiran seseorang berikut dengan alasannya. Absensi adalah suatu pendataan kehadiran, bagian dari pelaporan aktifitas suatu institusi, atau komponen institusi itu sendiri yang berisi data-data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan [10].

Absensi berarti tidak hadir, namun bias dikatan pula absensi merupakan ketidakhadiran atau kehadiran suatu objek dalam hal ini adalah orang, dimana orang tersebut terlibat dalam suatu organisasi yang mengharuskan adanya pemberitahuan tentang keadaan atau kehadiran atau ketidakhadirannya dalam ruang lingkup organisasi tersebut [11].

2.4 QRCODE

Qr adalah singkatan dari *Quick Response* karena ditujukan untuk diterjemahkan isinya dengan cepat. QR-Code salah satu tipe dari *barcode* yang dapat dibaca dengan kamera handphone. Qr-Code mampu menyimpan semua jenis data, seperti data angka/numerik, alphanumerik, biner, kanji/kana. Selain itu Qr-Code mampu menampung data secara *horizontal* dan *vertical*.

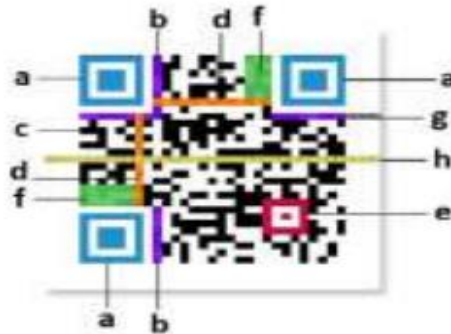
QR Code merupakan teknik yang mengubah data tertulis menjadi kode-kode 2-dimensi yang tercetak kedalam suatu media yang lebih ringkas. QR Code adalah *barcode* 2-dimensi yang diperkenalkan pertama kali oleh perusahaan Jepang Denso-Wave pada tahun 1994 [12].

Versi simbol QR Code berkisar dari Versi 1 ke Versi 40. Setiap versi memiliki konfigurasi modul yang berbeda atau jumlah modul (Modul ini mengacu pada titik-titik hitam dan putih yang membentuk QR Code). "Konfigurasi Modul" mengacu pada jumlah modul yang terkandung dalam simbol, dimulai dengan Versi 1 (21 x 21 modul) sampai ke Versi 40 (177 x 177 modul). Setiap nomor versi lebih tinggi terdiri dari 4 modul tambahan per samping [12].

QR Code memiliki kemampuan mengoreksi kesalahan untuk mengembalikan data jika kode kotor atau rusak. Empat tingkat kesalahan koreksi yang tersedia bagi pengguna, tingkat ini mampu mengoreksi kesalahan pada. Faktor lingkungan dan ukuran QR Code perlu dipertimbangkan untuk menghitung tingkat kesalahan. Tingkat Q atau H dapat dipilih jika kondisi lingkungan kotor yang akan menyebabkan QR Code mengalami kerusakan, sedangkan tingkat L dapat dipilih jika kondisi lingkungan lebih bersih dengan jumlah data yang besar. Tingkat M adalah tingkat yang paling sering digunakan [13].

Tujuan dari QR Code ini adalah untuk menyampaikan informasi secara cepat dan juga mendapat tanggapan secara cepat. Pada awalnya QR Code digunakan untuk pelacakan bagian kendaraan untuk *manufacturing*. Namun sekarang, telah digunakan untuk komersil yang ditujukan pada pengguna telepon seluler. QR Code adalah perkembangan dari barcode atau kode batang yang hanya mampu

menyimpan informasi secara horizontal sedangkan QR Code mampu menyimpan informasi lebih banyak, baik secara *horizontal* maupun *vertical* [14].



Gambar 2.1 Anatomi QR Code [14]

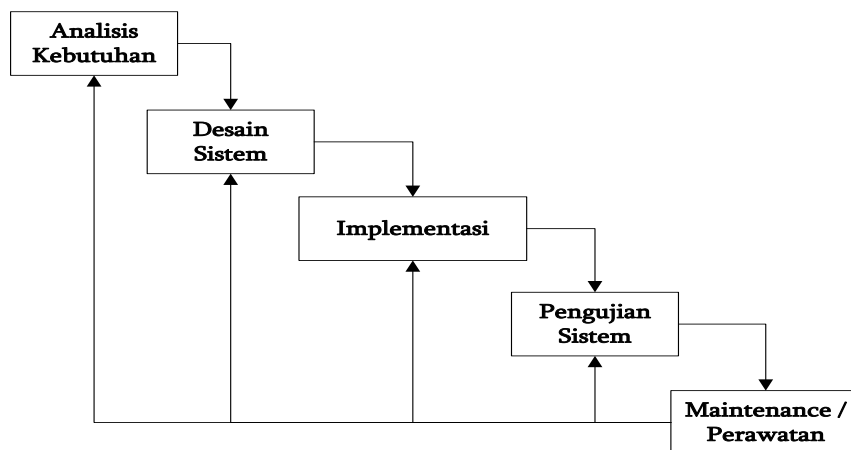
Beberapa penjelasan anatomi *QR Code* antara lain [14] :

- a. *Finder Pattern* berfungsi untuk identifikasi letak *QR Code*.
- b. *Format Information* berfungsi untuk informasi tentang *error correction level* dan *mask pattern*.
- c. *Data* berfungsi untuk menyimpan data yang dikodekan.
- d. *Timing Pattern* merupakan pola yang berfungsi untuk identifikasi koordinat pusat *QR Code*, berbentuk modul hitam putih.
- e. *Alignment Pattern* merupakan pola yang berfungsi memperbaiki penyimpangan *QR Code* terutama distorsi non linier.
- f. *Version Information* adalah versi dari sebuah *QR Code*.

Perbedaan karakteristik utama dari kode QR dibandingkan dengan *barcode* tradisional adalah bahwa kode QR berisi informasi vertikal dan horizontal dan tidak dibatasi oleh satu dimensi. Mengenai kapasitas penyimpanan informasi, kode QR dapat menyimpan sekitar 7.089 digit, sekitar 1.800 karakter Cina (kode 5 Besar), dan kode batang khas menampung maksimum 20 digit [15].

2.5 METODE WATERFALL

Metode *SDLC Waterfall* Merupakan salah satu metode yang mempunyai ciri khas bahwa pengerjaan setiap fase harus dikerjakan terlebih dahulu sebelum melanjutkan ke fase berikutnya. Dengan demikian hasilnya akan fokus terhadap masing-masing fase sehingga pengerjaan dilakukan secara maksimal karena tidak adanya pengerjaan secara parallel [16]. Metode *Waterfall* yang dapat dilihat pada gambar berikut :



Gambar 2.2 Metode *Waterfall* [17]

1. Kelebihan

- a. Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
- b. Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.

2. Kekurangan

- a. Diperlukan majemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
- b. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan.

2.6 ALAT BANTU PEMODELAN SISTEM

Adapun alat bantu pemodelan sistem yang digunakan penulis yaitu UML (*Unified Modeling Language*), *Use case Diagram*, *Activity diagram*, *Class Diagram*. Alat bantu ini digunakan untuk perancangan sistem yang akan dibangun.

2.6.1 UML (*Unified Modeling Language*)

Untuk menganalisis kebutuhan sistem penulis menggunakan alat bantu yaitu dengan *use case diagram*. Tujuan pembuatan *use case* adalah untuk mendapatkan dan menganalisis informasi persyaratan yang cukup untuk mempersiapkan model yang mengkomunikasikan apa yang diperlukan dari perspektif pengguna, tetapi bebas dari detail fisik tentang bagaimana sistem akan dibangun dan diimplementasikan. Dan ada pula beberapa pengertian menurut para ahli antara lain.

Sukamto dan Shalahuddin [18] mengungkapkan “UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisa dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Munawar [19] “UML(*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunai pengembangan sistem yang berorientasi objek”.

Sari dan Utami [20] “UML adalah metode terbuka untuk menspesifikasikan, memvisualisasikan, membuat, dan mendokumentasikan artefak sistem perangkat lunak yang sedang dikembangkan”.

Berdasarkan definisi atas dapat disimpulkan bahwa “*Unified Modeling Language* (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OO (*Object Oriented*)”.

2.6.2 *Use case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apa yang diperbuat sistem. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case diagram* adalah pola atau gambaran berbentuk diagram yang menggambarkan hubungan suatu sistem yang tengah di buat. Dalam penggambarannya, sistem yang dibuat harus berada didalam kotak sistem dan memiliki minimal satu aktor yang berada di luar sistem yang didefinisikan oleh para ahli dibawah ini :

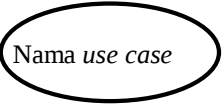
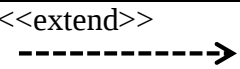
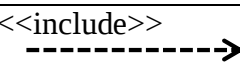
Sukanto dan Shalahuddin [18] menyatakan bahwa “*Use case* atau *diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

Munawar [19] “*Use case Diagram* yaitu deskripsi fungsi dari sebuah sistem dari perseptif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistem sendiri melalui sebuah cerita bagaimana sistem dipakai”.

Sari dan Utami [20] “*Use Case* adalah deskripsi fungsi dari sebuah system dari perspektif/sudut pandang para pengguna sistem”.

Dari beberapa pendapat tersebut dapat disimpulkan bahwa *use case* diagram merupakan permodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Tabel 2.1 Simbol *Use case Diagram* [18]

Nama Simbol	Simbol	Deskripsi
<i>Use case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i> .
<i>Aktor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
<i>Association</i> atau asosiasi		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
<i>Extend</i> atau ekstensi		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
<i>Generalisasi</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<i>Include</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau

		sebagai syarat dijalankan <i>use case</i> ini ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>
--	--	--

2.6.3 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Berikut tiga definisi *activity diagram* menurut para ahli :

Sukanto dan Shalahuddin [18] mengatakan bahwa : “Diagram aktivitas atau *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Munawar [19] mengungkapkan “*Activity diagram* adalah untuk menangkap tingkah laku dinami dari sistem dengan cara menunjukkan aliran pesan dari satu aktifitas ke aktifitas lainnya”.

Muslihudin dan Oktafianto [21] “Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem”.

Jadi kesimpulan dari pengertian *Activity diagram* adalah sebuah model yang menyerupai *flowchart* horizontal yang menunjukkan urutan tindakan yang merekam sebuah alur proses dan hasil dari tindakan tersebut.

Tabel 2.2 Simbol *Activity diagram* [18]

Nama Simbol	Simbol	Deskripsi
Status Awal	●	Status awal aktifitas sistem, sebuah diagram memiliki sebuah status awal

Aktifitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i>		Asosiasi percabangan dimana jika ada pilihan aktivitas atau lebih dari satu
Penggabungan / <i>join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status Akhir

2.6.4 Class Diagram

Class Diagram dapat merupakan implementasi dari sebuah *interface*, yaitu class abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Berikut ini merupakan penjelasan *Class Diagram* dari tiga ahli sebagai berikut :

Sukamto dan Shalahuddin [18] mengungkapkan bahwa “Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

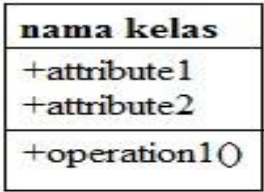
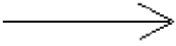
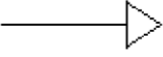
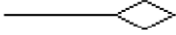
Munawar [19] mengungkapkan “*Class Diagram* menggambarkan atribut, *operation* dan juga *constraint* yang terjadi pada sistem”.

Muslihudin dan Oktafianto [21] menyatakan *Class Diagram* memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-

kolaborasi, serta relasi-relasi. Umum dijumpai pada pemodelan sistem berorientasi objek.

Jadi kesimpulan dari pengertian *Class Diagram* adalah adalah kumpulan *object* yang menggambarkan struktur statis dari sebuah sistem yang menunjukkan *object class* dan hubungannya serta untuk pengembangan sistem tersebut.

Tabel 2.3 Simbol *Class Diagram* [18]

Simbol	Nama	Keterangan
	<i>Kelas</i>	Kelas pada struktur sistem.
	<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	<i>Asosiasi</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Asosiasi berarah</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Generalisasi</i>	Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus).
	<i>Agregasi</i>	Relasi antar kelas dengan makna semua bagian (<i>wholepart</i>)

2.7 ALAT BANTU PERANCANGAN PROGRAM

2.7.1 Android

Android adalah sistem operasi yang memiliki karakteristik open source dan dimiliki oleh perusahaan Google, Inc. Ketika pertama kali dirilis, Android hanya ditujukan untuk perangkat mobile, seperti ponsel. Namun, sejak perilisan Android 3.0 (Honeycomb), sistem operasi Android secara resmi juga digunakan dalam tablet komputer sebagai bagian dari perkembangannya. Wahyutama dkk [22] menerangkan bahwa Android merupakan sistem operasi yang berasal dari Linux dan dirancang untuk perangkat *mobile*, seperti ponsel pintar dan tablet. Android memiliki karakteristik sebagai platform *open source*. Perkembangan Android terus berlanjut dan ditandai dengan pembaruan versi-versinya.

Android terus berkembang dan hal itu di tandai dengan pembaharuan versi-versinya. Terdapat beberapa versi-versi *Android* yaitu :

Tabel 2.4 Versi *Android* [23]

No	Versi Kode	Versi Platform	Release Year	Level Api
1	(no code name)	Android 1.0	2008	Api Level 1
2	(no code name)	Android 1.1	2008	Api Level 2
3	CUPCAKE	Android 1.5	2009	Api Level 3, NDK 1
4	DONUT	Android 1.6	2009	Api Level 4, NDK 2
5	ÉCLAIR	Android 2.0 Android 2.0.1 Android 2.1	2009	Api Level 5 Api Level 6 Api Level 7, NDK 3
6	FROYO	Android 2.2.X	2010	Api Level 8, NDK 4
7	GINGERBREAD	Android 2.3 Android 2.3.1 Android 2.3.2	2010	Api Level 9, NDK 5
8	GINGERBREAD	Android 2.3.3 Android 2.3.7	2010	Api Level 10
9	HONEY COMB	Android 3.0	2011	Api Level 11
10	HONEY COMB	Android 3.1	2011	Api Level 12, NDK 6

11	HONEY COMB	Android 3.2.X	2011	Api Level 13
12	ICE CREAM SANDWICH	Android 4.0, 4.0.1, 4.0.2	2011	Api Level 14, NDK 7
13	ICE CREAM SANDWICH	Android 4.0.3, 4.0.4	2011	Api Level 15, NDK 8
14	JELLYBEAN	Android 4.1.X	2012	Api Level 16
15	JELLYBEAN	Android 4.2.X	2012	Api Level 17
16	JELLYBEAN	Android 4.3.X	2012	Api Level 18
17	KIT KAT	Android 4.4 – Android 4.4.4	2013	Api Level 19
18	LOLLIPOP	Android 5.0	2014	Api Level 20
19	LOLLIPOP	Android 5.1	2014	Api Level 21
20	MARSHMALLOW	Android 6.0	2015	Api Level 23
21	NOUGAT	Android 7.0	2016	Api Level 24
22	NOUGAT	Android 7.1	2016	Api Level 25
23	OREO	Android 8.0	2017	Api Level 26
24	PIE	Android 9.0	2018	Api Level 28
25	Android 10	Android 10	2019	Api Level 29
26	Android 11	Android 11	2020	Api Level 30
27	Android 12	Android 12	2021	Api Level 31-32
23	Android 13	Android 13	2023	Api Level 33
24	Android 14	Android 14	2023	Api Level 24

2.7.2 Flutter

Flutter merupakan *software development kit* (SDK) buatan google yang berfungsi untuk membuat aplikasi mobile menggunakan bahasa pemrograman Dart, baik untuk Android maupun iOS.

Menurut Krisnada dan Tanone [24] “Flutter adalah sebuah SDK atau *framework open source* yang dikembangkan oleh Google untuk membuat atau mengembangkan aplikasi yang dapat berjalan dalam sistem operasi Android dan iOS”.

Menurut Anastasia dan Paputungan[25] “Flutter adalah *framework* yang membantu *developer* untuk membuat aplikasi *mobile multiplatform*”.

Menurut Suhendro dkk [26] “Flutter merupakan SDK (*Software Development Kit*) untuk pengembangan aplikasi *mobile* yang dikembangkan oleh Google. *Framework* ini dapat digunakan untuk membuat atau mengembangkan aplikasi *mobile* yang dapat berjalan pada *device* iOS dan Android”.

Berdasarkan beberapa definisi di atas dapat disimpulkan bahwa *Flutter* adalah *software development kit* atau *framework open source* untuk pengembangan aplikasi *mobile* pada *device* iOS dan Android.

2.7.3 Rest API

REST API merupakan gaya arsitektural yang mendefinisikan aturan-aturan untuk membuat web service. REST API dapat dispesifikasikan lagi ke dalam beberapa modeling language yang akan merepresentasikan format dari JSON/XML [27].

Nama Representational state transfer (REST) diciptakan oleh Roy Fielding dari University of California. Ini adalah layanan web yang sangat sederhana dan ringan dibandingkan dengan SOAP [28].

Cara kerja web service yaitu dengan bernavigasi melalui link-link yang disediakan seperti mengganti state dari halaman web. Sementara REST bekerja dengan bernavigasi melalui link-link HTTP dalam aktivitas tertentu, sehingga seperti terjadi perpindahan antara state. Beberapa perintah HTTP antara lain fungsi GET, POST, PUT, UPDATE atau DELETE. Supaya informasi yang diterima lebih mudah dibaca dan di-parsing disisi client, maka balasan yang dikirimkan oleh API Server dalam bentuk JSON[28].

API Merupakan *software interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk library dan menjelaskan bagaimana agar suatu software dapat berinteraksi dengan software lain [29].

2.7.4 Basis Data (*Database*)

Basis data adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh sistem informasi. Database atau biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam tabel- tabel yang saling berhubungan satu dengan yang lain, dengan menggunakan field/kolom pada tiap tabel yang ada [30]

Suatu bangunan basis data memiliki jenjang sebagai berikut :

1. Karakter, merupakan bagian data terkecil yang berupa angka, huruf, atau karakter khusus yang membentuk sebuah item data atau field.
2. Field/item, merupakan representasi suatu atribut dan record (rekaman/tupel) yang sejenis yang menunjukkan suatu item dari data.
3. *Record*/rekaman/tupel: Kumpulan dari field membentuk suatu record atau rekaman. *Record* menggambarkan suatu unit data individu yang tertentu.
4. *File*, merupakan kumpulan dari record-record yang menggambarkan satu kesatuan data yang sejenis. Mewakili tiap-tiap data.
5. *Database*, merupakan kumupan dari file atau tabel yang membentuk suatu *database*.

2.7.5 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL. *MySQL* merupakan *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi *General Public License* (GPL). Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial.

Menurut Enterprise [31] “MySQL adalah RDBMS yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan”.

Menurut Hidayatullah dan Kawistara [32] “MySQL adalah salah satu aplikasi DBMS yang sudah banyak digunakan oleh para pemrograman aplikasi web”.

Menurut Raharjo [33] “MySQL merupakan software RDBMS (atau *server database*) yang dapat mengelola dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user(*multi user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*)”.

Berdasarkan beberapa definisi di atas dapat disimpulkan bahwa *MySQL* adalah sebuah program database *server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL.

2.7.6 LARAVEL

Laravel adalah kerangka kerja aplikasi web berbasis PHP yang *open source*, menggunakan konsep *model–view–controller*. Laravel berada dibawah lisensi MIT, dengan menggunakan GitHub sebagai tempat berbagi kode.

Menurut Yudhanto dan Prasetyo [34] “Laravel adalah sebuah Framework PHP yang dirilis di bawah lisensi MIT dan dibangun dengan konsep MVC (*Model View Controller*) .

Laravel adalah sebuah MVC *web development framework* yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktifitas pekerjaan dengan sintak yang bersih dan fungsional yang dapat mengurangi banyak waktu untuk implementasi [35].

Laravel memberikan keterbaruan alat untuk berinteraksi dengan database disebut dengan migration [35]. Dengan migration, pengembang dapat dengan mudah untuk melakukan modifikasi sebuah database pada sebuah platform secara independen karena implementasi skema database direpresentasikan dalam sebuah class. Migration dapat berjalan pada beberapa basis data yang telah didukung Laravel (MySQL, PostgreSQL, MSSQL, dan SQLITE) dan untuk implementasi *Active Record* pada Laravel disebut *Eloquent* yang menggunakan standard modern OOP. Laravel juga memberikan sebuah *Command Line Interface* disebut dengan artisan dengan artisan, pengembang dapat berinteraksi dengan aplikasi untuk sebuah aksi seperti *migrations*, *testing*, atau membuat *controller* dan *model*. Selain itu, laravel juga memiliki Blade template engine yang memberikan estetika dan kebersihan kode pada *view* secara parsial.

2.7.7 Visual Studio Code

Visual studio Code merupakan aplikasi *cross platform* yang dapat digunakan berbagai sitem operasi seperti windows, Linux, dan Mac OS. VS Code

termasuk *software* yang ringan namun kuat editor sumbernya dengan deskop. Menggunakan berbagai macam bahasa pemrograman seperti Java, JavaSkrip, Go, C++, dan masih banyak yang lainnya.

Permana [36] menyatakan “Visual Studio Code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows”.

Hendri [37] menyatakan “Visual Studio Code adalah editor *source code* atau text editor yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS yang mendukung program bahasa PHP”.

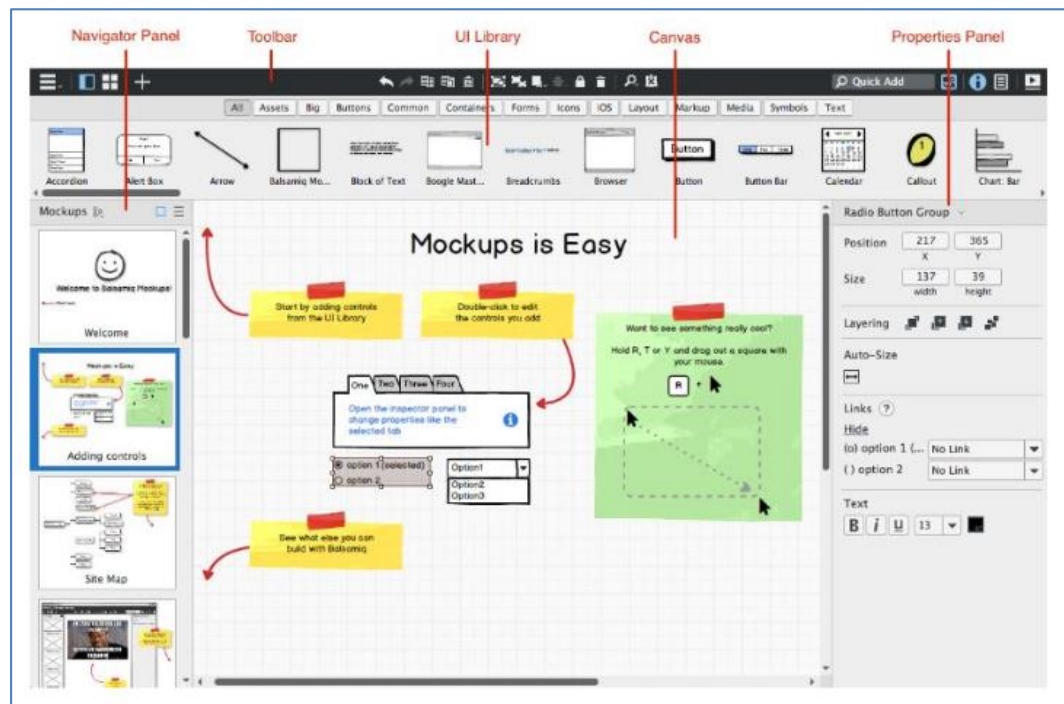
Ramadhan dan Nufriana [38] menyatakan “Visual Studio Code merupakan sebuah aplikasi editor *code open source* yang dikembangkan oleh Microsoft untuk sistem operasi Windows, Linux, dan MacOS”.

Berdasarkan pendapat tersebut dapat kita simpulkan bahwa Visual studio merupakan sebuah teks editor ringan, handal dan *open source* dibuat oleh microsoft untuk sistem operasi *multiplatform*.

2.7.8 Balsamiq Mockup

Balsamiq Mockups adalah aplikasi yang sangat berguna untuk mendesain halaman web. Mockups biasanya di gunakan oleh para designer untuk memulai suatu coretan-coretan, dengan kata lain kalian tidak perlu lagi menggunakan pensil untuk merancang sesuatu [39]. Software ini berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna. Alih-alih menggambar sketsa (*wireframe*) atau prototype rancangan desain website di atas kertas balsamiq mockups membantu seorang web desainer membuat tampilan web

dalam bentuk gambar di komputer. Berikut ini merupakan tampilan dari aplikasi balsamiq, seperti terlihat pada gambar 2.2.



Gambar 2.3 Gambar Aplikasi Balsamiq

1. *Toolbar*

Pada bagian Toolbar berisi serangkaian ikon untuk melakukan tindakan pada area antarmuka pengguna yang terdiri dari tombol navigasi pada aplikasi balsamiq.

2. *UI Library*

UI Library merupakan kumpulan tools untuk membuat perancangan yang sudah disediakan langsung oleh Balsamiq Mockups.

3. Kanvas

Pada bagian ini merupakan area kerja utama sebagai tempat mockup UI anda hidup. Setelah menambahkan kontrol UI, pengguna dapat memindahkan mengubah ukuran dan menyesuaikan dengan kebutuhan.

4. Panel Navigator

Panel navigator di sebelah kiri menunjukkan daftar mockup, aset, dan simbol dalam proyek pengguna. Pengguna dapat memilih beberapa objek di panel navigator.

5. Panel Properti

Panel dengan posisi kanan pada aplikasi menampilkan properti untuk objek atau kontrol yang dipilih. Menu ini bergantung pada ikon mana yang dipilih. Panel ini dapat dimunculkan ataupun disembunyikan.

2.8 PENELITIAN SEJENIS

Studi penelitian sejenis adalah kajian dari pendapat orang lain atau peneliti yang dapat dijadikan sebagai sumber teori serta acuan bagi peneliti dalam melaksanakan penelitian. Tinjauan penelitian sejenis dapat dilakukan untuk menemukan metode yang relevan dengan permasalahan penelitian. Hasil tinjauan penelitian sejenis dapat dijadikan sebagai dasar bagi peneliti untuk memilih metode dan melakukan pengembangan terhadap solusi dari permasalahan yang ada. Tabel 2.5 menyajikan penelitian-penelitian yang relevan dengan topik yang peneliti angkat.

Tabel 2.5 Penelitian Sejenis

No.	Penulis	Judul	Masalah	Metode	Kesimpulan
1.	Bagus Joyo Tirto Nugroho, Dkk [40]	Perancangan Aplikasi Absensi	Dispora Sumatera Utara masih	<i>Blackbox (blackbox testing)</i>	Dengan mengimplementas ikan saat ini sudah

		Pegawai Berbasis Web Dengan Blackbox Testing pada DISPORA Sumatera Utara	melakukan pengolahan absensi pegawai secara manual, menghambat efisiensi dan integrasi sistem informasi		secara komputerisasi dan waktu dalam hal mencatat data absensi pegawai Dispora Sumatera Utara karena proses absensi sudah tidak memerlukan buku atau arsip yang tersedia. Dengan itu aplikasi absensi pegawai ini dapat memberikan kemudahan dalam melakukan absensi bagi Dinas Pemuda dan Olahraga Sumatera Utara.
2.	Badie Uddin, Dkk [41]	Rancangan Sistem Informasi Aplikasi Absensi Karyawan Berbasis Mobile GPS Pada PD. Kilau Laju Intan (PD. KLI)	Dalam pelaksanaan kegiatan absensi, dilakukan dengan membubuhkan tanda tangan diatas kertas, namun memiliki kendala yaitu tidak ada back up jika arsip rusak dan hilang dikarenakan data hanya di tulis menggunakan kertas. Demikian juga jika karyawan	<i>Waterfall</i>	Dengan mengimplementasikan sistem informasi aplikasi absensi karyawan dapat mampu mencatat lokasi karyawan secara akurat, mengurangi kecurangan, dan memastikan bahwa karyawan berada di lokasi kerja yang sesuai. Selain itu, aplikasi ini juga meningkatkan akurasi data kehadiran, memudahkan manajemen dalam memantau kinerja

			tidak hadir dengan alasan cuti dan sebagainya maka bagian absensi harus memasukkan data manual melalui MS Excel		karyawan, dan mempercepat proses administrasi terkait absensi.
3.	Syahrul Usman, Dkk [42]	Pengembangan Absensi Berbasis Mobile Aplikasi Pada Badan Kepegawaian Dan Pengembangan Sumber Daya Manusia Kabupaten Bone	Di berbagai instansi pemerintahan maupun instansi swasta telah mengambil kebijakan untuk tidak menggunakan lagi absen mesin sidik jari yang berpotensi sebagai media penularan Covid-19. Hal ini tentunya akan berpengaruh kepada pencatatan kinerja seorang ASN dan pencatatan administrasi tiap instansi	<i>Komunikasi data Representational State Transfer (Rest)</i>	Sistem ini juga dapat mempermudah pegawai untuk melakukan absensi secara digital melalui perangkat mobile, mengurangi ketergantungan pada sistem manual, dan meminimalisir kesalahan administrasi. Selain itu, aplikasi ini juga menyediakan fitur pelaporan dan monitoring yang memudahkan pengelolaan data kehadiran secara real-time, sehingga mendukung peningkatan manajemen sumber daya manusia di lingkungan pemerintah daerah.
4.	Muhamad Alda, Dkk [43]	Perancangan Aplikasi Sistem Informasi	Proses kehadiran manual masih	<i>Waterfall</i>	Dengan mengimplementasikan sistem

		Absensi Online Berbasis Mobile (Studi Kasus: Sma Negeri 1 Dolok Masihul)	tidak efisien dan dapat memakan waktu yang lama. Saat terjadi kesalahan, data kehadiran menjadi tidak terstruktur dan sulit dipantau		informasi aplikasi absensi karyawan dapat mampu menghasilkan rancangan sistem informasi dalam bidang absensi siswa Sma Negeri 1 Dolok Masihul. Dengan adanya aplikasi absensi berbasis android, dapat mempermudah guru mata pelajaran dalam melakukan proses absensi siswa di dalam kelas.
5	Puput Apriadi, Entis Sutrisna [44]	Perancangan Aplikasi Absensi Karyawan Berbasis Mobile Menggunakan GPS (Studi Kasus PT. Trans Retail Indonesia)	Masalah yang dihadapi yaitu Harga HID Proximity Card sangat mahal, Jika terjadi kerusakan pada kartu untuk pembelian nomor kartu HID Proximity Card harus satu package yang sudah tersedia dan sering disalahgunakan untuk pemalsuan absen misalnya di titipkan kepada rekan kerja sehingga menimbulkan tindak manipulasi absensi. Mesin	<i>Rapid Application Development (RAD)</i>	Dengan menggunakan aplikasi Absensi Karyawan pada PT. Trans Retail Indonesia ini memberikan kemudahan kepada Karyawan dalam proses absensi masuk dan pulang. Kemudian menggunakan aplikasi Absensi Karyawan pada PT. Trans Retail Indonesia dapat meningkatkan efisiensi kehadiran karyawan, baik dari segi waktu dan biaya, serta dapat menjadikan nilai

			untuk melakukan absen sering terjadi kendala pada saat melakukan absensi.		tambah dari fasilitas yang dimiliki oleh PT. Trans Retail Indonesia.
--	--	--	---	--	--

Berdasarkan tabel 2.5 penelitian sejenis tersebut yang menjadikan rujukan pada penelitian. Adapun persamaan yang menjadi permasalahan utama yang melatar belakangi adalah Dalam pelaksanaan kegiatan absensi, dilakukan dengan membubuhkan tanda tangan diatas kertas, namun memiliki kendala yaitu tidak ada back up jika arsip rusak dan hilang dikarenakan data hanya di tulis menggunakan kertas.

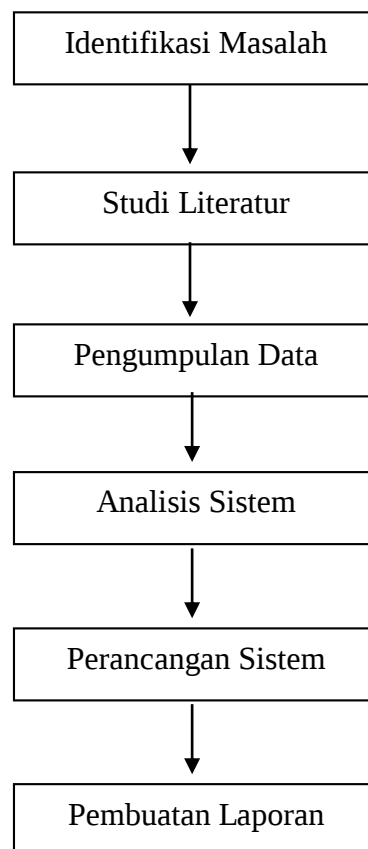
Perbedaan dengan penelitian terdahulu yang akan di tambahkan pada penelitian penulis pada Penelitian ini mengembangkan sistem absensi berbasis *mobile* dengan objek penelitian pada PDAM Muaro Jambi serta adanya fitur Scan Qr Code. Sistem ini dirancang dengan metode *waterfall* dan menggunakan diagram *Use Case*, *Class*, serta *Activity* untuk visualisasi.

BAB III

METODOLOGI PENELITIAN

3.1 KERANGKA KERJA PENELITIAN

Kerangka kerja penelitian merupakan ilustrasi dari tahap-tahap kegiatan yang penulis lakukan dalam penelitian agar penelitian dapat dilakukan secara sistematis dan memiliki arah dalam bagaimana melakukannya. Adapun kerangka kerja yang digunakan adalah sebagai berikut :



Gambar 3.1 Kerangka Kerja Penelitian

Berdasarkan kerangka kerja yang telah digambarkan pada gambar diatas maka dapat diuraikan pembahasan masing – masing tahap dalam penelitian adalah sebagai berikut :

1. Identifikasi Masalah

Pada tahap ini penulis melakukan penelitian yaitu mengidentifikasi masalah PDAM Muaro Jambi dalam Pengelolaan presensi secara konvensional dengan mencatat pada lembar absensi, akan membutuhkan waktu yang lama karena harus satu per satu data dimasukkan. Rentan juga adanya manipulasi data atau terjadi data yang ganda, sehingga tidak mendapatkan hasil yang akurat. Cara ini sangatlah rawan bagi suatu instansi karena tingkat kedisiplinan yang tidak dapat di kontrol dan dapat disalah gunakan oleh orang yang tidak bertanggung jawab, kerugian lain yang mungkin muncul pada sistem presensi manual adalah rekapitulasi data yang masih memakan banyak waktu dan tenaga

2. Studi Literatur

Pada tahap ini penulis menambah wawasan guna mendapatkan sebuah topik yang layak diangkat sebagai sebuah penelitian dengan mempelajari dan memahami teori dan konsep dimana penulis banyak melakukan penelitian pada jurnal, internet dan buku yang referensinya didapatkan dari berbagai sumber. Studi literatur membantu peneliti agar memahami bidang yang sedang dikaji terhadap apa yang telah peneliti sebelumnya lakukan sehingga peneliti dapat menjadikan penelitiannya sebagai penelitian yang berbeda.

3. Pengumpulan Data

Pengumpulan data adalah kegiatan yang dilakukan untuk mengumpulkan informasi yang berkaitan dengan apa yang akan diperlukan. Sebagai bahan pendukung yang sangat penting bagi penulis untuk mencari atau mengumpulkan data yang diperlukan dalam penelitian ini dan harus menjabarkan permasalahan dan menentukan solusi yang ingin dicapai. berikut metode pengumpulan data yang akan dilakukan:

a. Pengamatan langsung (*Observation*)

Pada kegiatan pengamatan ini, Penulis melakukan pengamatan langsung terhadap objek penelitian dengan melihat proses presensi PDAM Muaro Jambi, kemudian penulis melakukan pencatatan sistem kerja yang sedang berjalan untuk dianalisis kelemahan dari sistem yang berjalan serta dapat mengembangkan sistem yang baru untuk membantu mempercepat proses presensi pegawai.

b. Wawancara

Wawancara merupakan teknik pengumpulan data yang dilakukan dengan cara mengadakan tanya jawab kepada pihak PDAM Muaro Jambi yaitu Bapak Asko, untuk memperoleh informasi-informasi yang dibutuhkan untuk merancang aplikasi yang akan dibuat agar mendapat data yang relevan mengenai permasalahan yang dihadapi. Wawancara ini dilakukan untuk mendapatkan informasi atau penjelasan langsung dari pihak yang berwenang tentang proses presensi yang sedang berjalan pada PDAM Muaro Jambi.

4. Analisis Sistem

Berdasarkan indentifikasi masalah diatas, maka pada tahap analisis sistem ini dilakukan analisa kebutuhan perangkat lunak dengan cara menguraikan sistem ke dalam komponen-komponen pembentuknya untuk mengetahui bagaimana komponen-komponen tersebut bekerja dan saling berinteraksi satu sama lain untuk mencapai tujuan sistem.

5. Perancangan Sistem

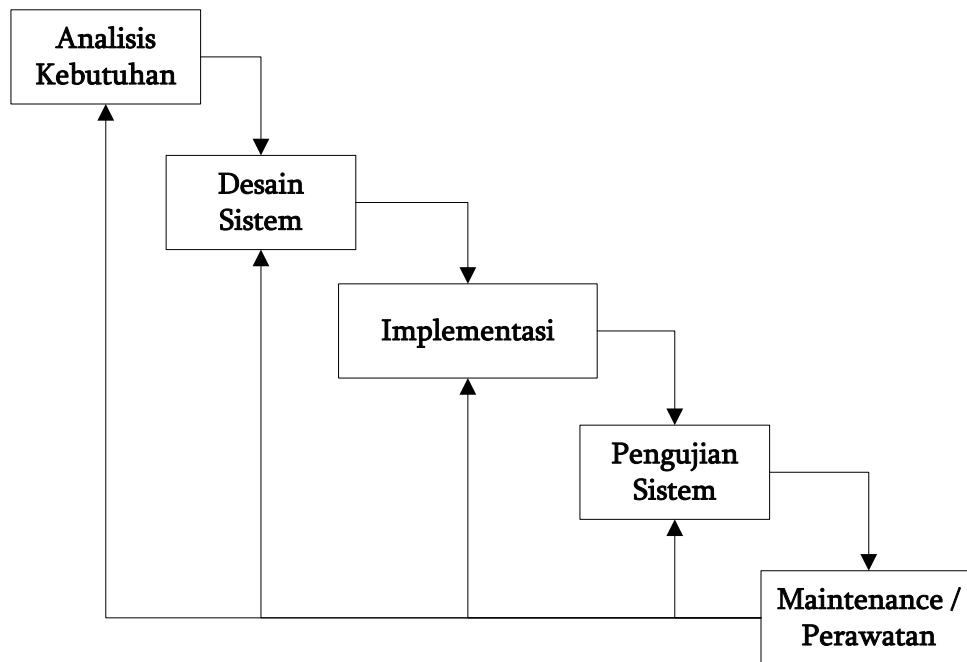
Perancangan sistem sangat penting dalam membangun sebuah aplikasi karena proses ini menggambarkan bagaimana suatu sistem dibentuk mulai dari penggambaran perencanaan sampai pada tahapan pembuatan fungsi yang berguna bagi jalannya sebuah aplikasi. Pada penelitian ini, perancangan sistem menggunakan diagram UML (*Unified Modeling Language*) meliputi *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*.

6. Pembuatan Laporan

Pada tahap ini dilakukan penyusunan laporan dari semua tahapan kerja penelitian untuk dapat digunakan pada waktu yang akan datang dan untuk tahapan pengembangan aplikasi selanjutnya.

3.2 METODE PENGEMBANGAN SISTEM

Penelitian ini menggunakan metode pengembangan sistem model waterfall. Model ini dipilih karena pendekatannya yang lebih terstruktur dan sistematis, sehingga dianggap lebih efisien untuk digunakan dalam proses pembuatan sistem. Adapun model *waterfall* yang digunakan dalam proses pengembangan aplikasi dapat dilihat pada gambar 3.2 berikut :



Gambar 3.2 Model *Waterfall* [17]

Berdasarkan model *waterfall* pada Gambar 3.2 Adapun penjelasan dari metode pengembangan sistem dengan model *waterfall* ini adalah sebagai berikut:

1. Analisis Kebutuhan

Dalam tahap ini dilakukan identifikasi kebutuhan dari sistem yang akan dibuat dengan mengetahui permasalahan dan sistem yang sedang terjadi pada PDAM Muaro Jambi dan mencari solusi yang diperlukan yaitu dengan merancang aplikasi dan merancang fungsional dan *non fungsional* sistem agar sistem yang diperlukan sesuai dengan kebutuhannya.

2. Desain Sistem

Pada tahap ini dilakukan perancangan data, antar muka (*interface*) dan model sistem dengan menggunakan *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Serta melakukan perancangan *input* dan *output* untuk tampilan pada program dan struktur data untuk *database* yang akan digunakan.

3. Implementasi

Pada tahap ini sistem yang telah dirancang, diimplementasikan dengan menggunakan program bantu yaitu Android Studio, PHP dan MySQL, kemudian dilakukan pengujian terhadap tiap-tiap unit atau modul yang telah dibuat.

4. Pengujian Sistem

Pada tahap ini dilakukan pengujian menggunakan metode *black box* dimana penulis melakukan pengecekan hasil keluaran dari aplikasi dan apabila hasil keluar tidak sesuai atau terjadi kesalahan maka penulis melakukan perbaikan agar hasil keluar dari aplikasi sesuai dengan hasil yang diharapkan.

5. *Maintenance* / Perawatan

Tahapan perawatan ini tidak dilakukan oleh penulis dikarenakan tahapan ini membutuhkan waktu yang cukup lama dan tahap *maintenance* / pemeliharaan harus dilakukan secara berkala.

3.3 ALAT BANTU PENELITIAN

Penelitian yang dilakukan ini membutuhkan alat bantu pengembangan program (*tools*) baik berupa perangkat keras (*hardware*) maupun perangkat lunak (*software*) serta bahan penelitian yang digunakan untuk merancang program antara lain :

1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini terdiri dari berbagai komponen yang mendukung proses pengembangan sistem secara optimal.

- a. Laptop : ASUS TUF GAMING F15 FX506HM

- b. Processor : Intel Core i7-11800H
- c. RAM : 16 GB DDR4
- d. Memory : 1 TB SSD M.2 NVMe + 512 GB SSD M.2 NVMe
- e. Graphics Card Unit : NVIDIA GeForce RTX 3060 Laptop GPU

Serta satu unit *Smartphone* Oppo A74 5G dengan spesifikasi sebagai berikut :

- a. Prosesor : Snapdragon Octacore
- b. Memori : 4 GB
- c. Penyimpanan : 64GB
- d. Sistem operasi : Android 13 (Tiramisu)

2. Perangkat Lunak (*Software*)

Kebutuhan *software* yang digunakan dalam penelitian ini adalah sebagai berikut :

- a. Android Studio Versi 4
- b. Java JDK (*Java Development Kit*) versi 17
- c. SDK (*Software Development Kit*) Android Api 32
- d. MySQL
- e. Laravel
- f. Flutter
- g. Visual Studio Code
- h. Webbrowser (Google Chrome, Mozilla Firefox)
- i. Microsoft Office (Word, Visio)

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 ANALISIS SISTEM

4.1.1 Gambaran Umum PDAM Muaro Jambi

PDAM Tirta Muaro Jambi merupakan bagian dari pemekaran PDAM Tirta Batanghari Kabupaten Batanghari yang semula bernama BPAMD Kabupaten Muaro Jambi yang terbentuk berdasarkan Keputusan Bupati Muaro Jambi Nomor : 24 tahun 2002. Pembentukan dan pendirian PDAM Tirta Muaro Jambi berdasarkan PERDA Muaro Jambi No. 07 Tahun 2003 tanggal 13 Maret 2003.

VISI” Menjadi Perusahaan Daerah yang Sehat, Mandiri, dan Profesional”.

MISI PDAM Tirta Muaro Jambi :

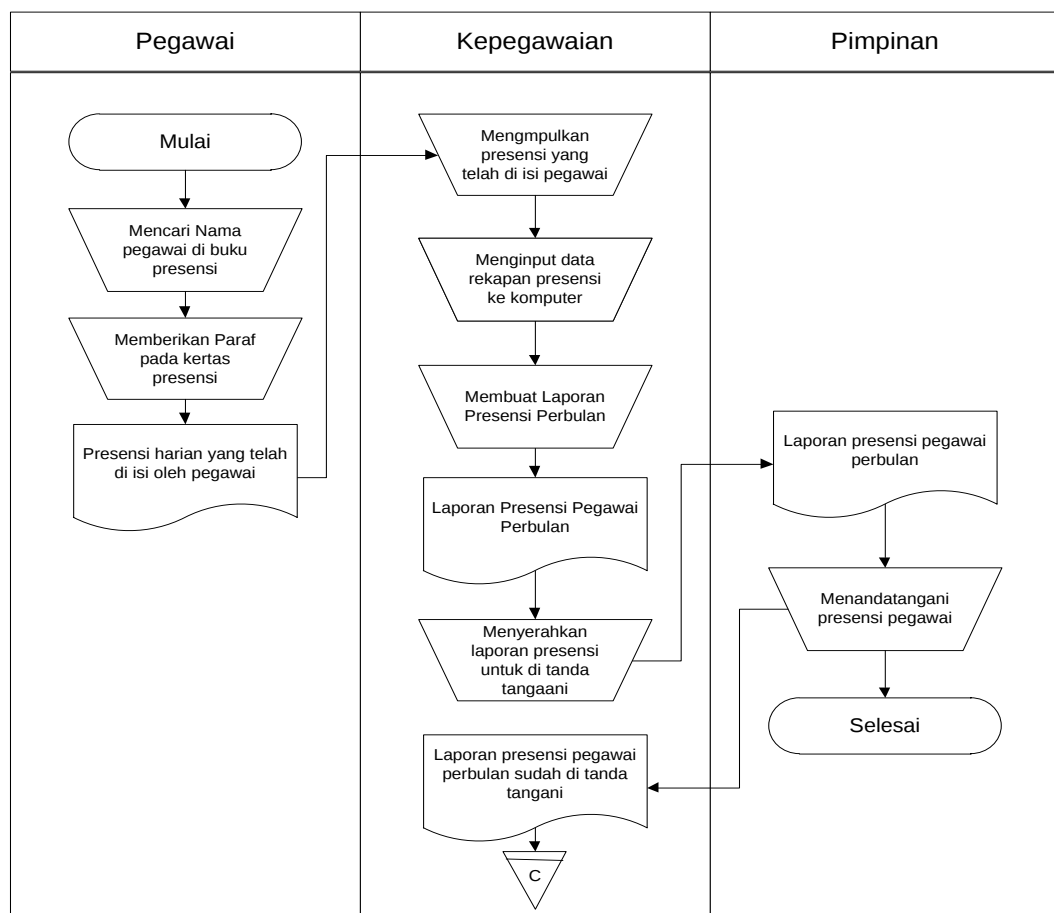
1. Mengelola Perusahaan secara Profesional
2. Meningkatkan kualitas SDM dan Kesejahteraan Karyawan
3. Memberikan Pelayanan Yang Memenuhi Harapan Pelanggan
4. Menghasilkan Air Minum yang Memenuhi Persyaratan (Kualitas, Kuantitas, Kontinuitas)
5. Meningkatkan cakupan pelayanan

4.1.2 Analisis Sistem Yang Berjalan

Menganalisis suatu sistem yang sedang berjalan merupakan salah satu tahap untuk menganalisis suatu sistem apakah sesuai dengan tujuan utama sistem itu sendiri yaitu mempermudah user sistem. Sebagai berikut :

1. Pegawai menemui bagian administrasi, kemudian informasikan data diri
2. Administrasi mencatat data Pegawai ke dalam buku absensi
3. Tiap akhir periode administrasi menyerahkan laporan kepada pimpinan.

Berdasarkan penjabaran diatas maka disusun *flowchart document* berikut ini :



Gambar 4.1 Sistem Yang Sedang Berjalan

4.1.3 Kelemahan Sistem Yang Sedang Berjalan

Berdasarkan hasil pengamatan, maka penulis memperoleh kesimpulan bahwa proses pengolahan data Absensi pada Kantor PDAM Tirta Muaro Jambi saat ini masih terdapat beberapa kendala yaitu :

1. Tingkat keamanan data yang masih rendah karena beberapa data masih disimpan dalam bentuk arsip yang beresiko dapat terjadi kerusakan, maupun kehilangan data dan dalam proses pencatatan data absensi Pegawai masih menggunakan kertas/arsip sehingga membutuhkan waktu yang lama dan kerap kali terjadi kesalahan selama pencatatan yang tanpa disadari sehingga data yang salah terlanjur tersimpan dan tidak terkoreksi kembali.
2. Proses pencarian data membutuhkan waktu yang relatif lama karena data terdapat pada buku agenda dalam bentuk catatan tertulis yang disimpan dalam lemari arsip yang datanya telah sangat banyak, serta disimpan dalam bentuk arsip yang terpisah.
3. Setiap data dicatat kedalam buku agenda di anggap kurang efektif dan efisien karena data tidak saling terintegrasi sehingga mengalami kesulitan saat harus merekap data dalam pembuatan laporan karena harus menyusun kembali data-data lama.

4.1.4 Solusi Pemecahan Masalah

Dari permasalahan tersebut salah satu solusi yang dapat di gunakan oleh Kantor PDAM Tirta Muaro Jambi untuk mengatasi permasalahan tersebut adalah dengan merancang suatu sistem informasi Absensi pada Kantor PDAM Tirta Muaro Jambi. Adapun solusi dari sistem yang akan dirancang tersebut

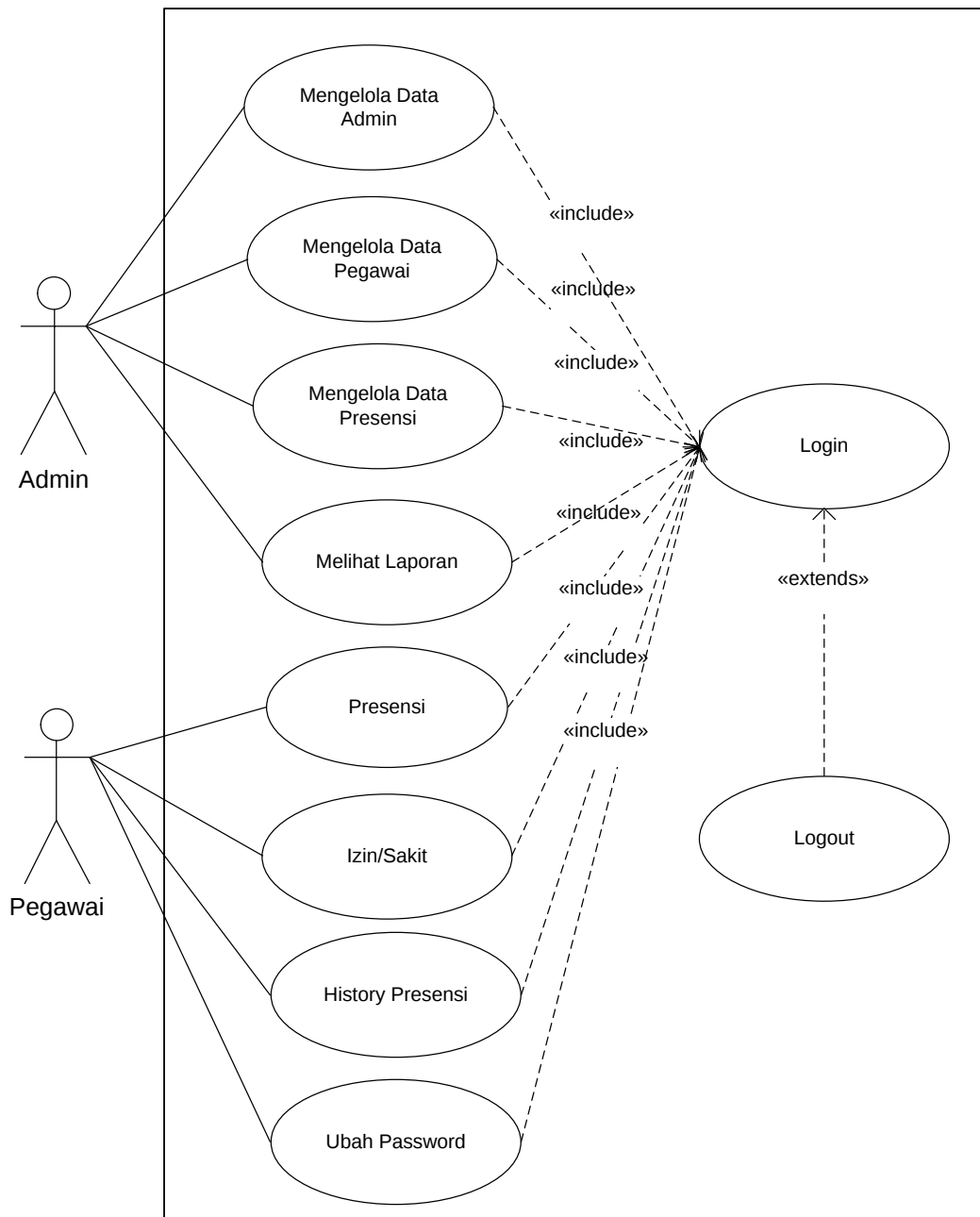
sebagai berikut :

1. Sistem yang dirancang dilengkapi sistem keamanan data, pencarian data dan fungsi peringatan data yang kosong, sehingga dapat meminimalisir terjadinya pembobolan oleh pihak yang tidak terkait dengan sistem, kehilangan data serta kesalahan penginputan data.
2. Proses pencarian data lebih cepat dan langsung dapat digunakan untuk kepentingan tertentu sehingga lebih menghemat waktu, dengan memanfaatkan fungsi pencarian pada sistem.
3. Pengolahan data akan dilakukan secara terkomputerisasi, dimana data-data absensi Pegawai menggunakan teknologi QR CODE (*Quick Response Code*) saling terintegrasi dan data yang telah diinput sebelumnya nanti akan diproses lebih cepat serta akurat sehingga dapat membantu dalam pembuatan laporan.

4.2 ANALISIS KEBUTUHAN PERANGKAT LUNAK

4.2.1 Use Case Diagram

Use case diagram menyajikan interaksi antara aktor admin dan pegawai sebagai pengguna yang dapat berinteraksi dengan sistem, dan mengoperasikan sistem seperti terlihat pada gambar 4.2.



Gambar 4.2 Diagram Use Case

4.2.2 Deskripsi Use Case

Berdasarkan *Use Case Diagram* yang dibahas sebelumnya maka dapat di deskripsikan *Use Case* pada sistem, yang dituangkan dalam bentuk tabel berikut:

1. Deskripsi Use Case Login

Deskripsi *Use Case login* merupakan aktivitas yang dilakukan user dalam melakukan *login* yang berfungsi untuk melakukan pengolahan data pada aplikasi, dan dideskripsikan pada tabel 4.1.

Tabel 4.1 Deskripsi Use Case Login

Nama	Login	
Aktor	Admin dan Pegawai	
Deskripsi	Aktor melakukan <i>login</i> dan masuk kehalaman utama	
Exception	<i>Username</i> dan <i>Password</i> tidak benar	
Pre Condition	<i>Username</i> dan <i>Password</i> sudah harus tersimpan dalam <i>database</i>	
Aktor		Sistem
Skenario Normal		
1. Aktor mengakses halaman <i>login</i>		
		2. Sistem menampilkan halaman <i>login</i> .
3. Aktor melakukan <i>login</i> dengan mengisi <i>Username</i> dan <i>Password</i> , kemudian meng-klik tombol <i>Login</i>		
		4. Sistem membuka koneksi <i>database</i>
		5. Sistem melakukan validasi <i>Username</i> dan <i>Password</i> benar
		6. Validasi berhasil, sistem menampilkan halaman utama
Skenario Alternatif		
S-1 5a: Sistem akan menampilkan pesan apabila ada data yang tidak terisi dan proses login gagal		
S-1 5b: Sistem memberikan kesempatan untuk mengulang kembali		
Post Condition	Aktor berhasil mengakses halaman utama	

2. Deskripsi Use Case Mengelola Data Admin

Deskripsi *Use Case* untuk mengelola data admin ini akan menjelaskan fungsional dari *Use Case* mengelola data admin terbagi atas menambah data,

mengubah data dan menghapus data admin pada *database*. Seperti yang terlihat pada tabel 4.2 :

Tabel 4.2 Deskripsi Use Case Mengelola Data Admin

Nama	Mengelola Data Admin
Aktor	Pimpinan
Deskripsi	Aktor menambah, mengubah, dan menghapus data Admin pada <i>database</i>
Exception	Data yang diinputkan tidak lengkap, data yang dimodifikasi tidak lengkap dan data yang akan dihapus tidak tersedia
Pre Condition	Aktor telah melakukan <i>login</i> sebagai admin
Aktor	Sistem
Skenario Normal	
1. Aktor memilih menu admin	
	2. Sistem membuka koneksi ke <i>database</i>
	3. Sistem menampilkan data admin
4. - Jika ingin menambah data admin maka <i>sub scenario</i> yang berlaku adalah S1 - Jika ingin mengubah data admin maka <i>sub scenario</i> yang berlaku adalah S2 - Jika ingin menghapus data admin maka <i>sub scenario</i> yang berlaku adalah S3	
Sub Scenario	
S1- Tambah Data Admin	
1. Aktor mengklik tambah	
	2. Sistem menampilkan form inputan
3. Aktor menambahkan data admin	
4. Aktor mengklik simpan	
	5. Sistem memvalidasi data inputan dan menampilkan pesan.
	6. Sistem membuka koneksi <i>database</i>

	7. Sistem menyimpan data yang diinputkan.
	8. Sistem menampilkan pesan “Berhasil simpan data admin”
S2- Ubah Data Admin	
1. Aktor memilih data yang akan di ubah	
2. Aktor memilih tombol ubah	
	3. Menampilkan data yang akan di ubah
4. Aktor memodifikasi data	
5. Aktor memilih tombol simpan	
	6. Sistem memvalidasi data inputan dan menampilkan pesan.
	7. Sistem membuka koneksi <i>database</i>
	8. Sistem menyimpan data yang diinputkan.
	9. Sistem menampilkan pesan “Berhasil ubah data admin”
S3- Hapus Data Admin	
1. Aktor mengklik tombol Hapus.	
	2. Sistem menkonfirmasi hapus data
	3. Sistem menampilkan pesan konfirmasi untuk menghapus data.
4. Aktor mengklik pilihan ok.	
	5. Sistem membuka koneksi <i>database</i>
	6. Sistem menghapus data admin
	7. Sistem menampilkan pesan “Berhasil hapus data admin”
Skenario Alternatif	
S-1 5a: Sistem akan menampilkan pesan apabila ada data yang tidak terisi dan proses penyimpanan data dibatalkan	
S-1 5b: Sistem memberikan kesempatan untuk mengulang kembali	
S-2 6a: Sistem akan menampilkan pesan apabila ada data yang tidak lengkap dan proses penyimpanan data dibatalkan	
S-2 6b: Sistem memberikan kesempatan untuk mengulang kembali.	
S-3 4a: jika aktor menjawab batal, sistem kembali menampilkan <i>table</i> admin	

3. Deskripsi Use Case Mengelola Data Pegawai

Post Condition	Aktor berhasil mengelola data Admin untuk menambah, mengubah dan menghapus data admin pada <i>database</i>
-----------------------	--

Deskripsi *Use Case* untuk mengelola data Pegawai ini akan menjelaskan fungsional dari *Use Case* mengelola data Pegawai terbagi atas mengubah data dan menghapus data Pegawai pada *database*. Seperti yang terlihat pada tabel 4.3 :

Tabel 4.3 Deskripsi Use Case Mengelola Data Pegawai

Nama	Mengelola Data Pegawai
Aktor	Admin
Deskripsi	Aktor menambah, mengubah, dan menghapus data Pegawai pada <i>database</i>
Exception	Data yang diinputkan tidak lengkap, data yang dimodifikasi tidak lengkap dan data yang akan dihapus tidak tersedia
Pre Condition	Aktor telah melakukan <i>login</i> sebagai admin
Aktor	Sistem
Skenario Normal	
1. Aktor memilih menu Pegawai	
	2. Sistem membuka koneksi ke <i>database</i>
	3. Sistem menampilkan data Pegawai
4. - Jika ingin menambah data Pegawai maka sub scenario yang berlaku adalah S1 - Jika ingin mengubah data Pegawai maka sub scenario yang berlaku adalah S2 - Jika ingin menghapus data Pegawai maka sub scenario yang berlaku adalah S3	
Sub Scenario	
S1- Tambah Data Pegawai	
1. Aktor mengklik tambah	
	2. Sistem menampilkan form inputan
3. Aktor menambahkan data Pegawai	

4. Aktor mengklik simpan	
	5. Sistem memvalidasi data inputan benar.
	6. Sistem membuka koneksi <i>database</i>
	7. Sistem menyimpan data yang diinputkan.
	8. Sistem menampilkan pesan “Berhasil simpan data pegawai”
S2- Ubah Data Pegawai	
1. Aktor memilih data yang akan di ubah	
2. Aktor memilih tombol ubah	
	3. Menampilkan data yang akan di ubah
4. Aktor memodifikasi data	
5. Aktor memilih tombol simpan	
	6. Sistem memvalidasi data inputan benar.
	7. Sistem membuka koneksi <i>database</i>
	8. Sistem menyimpan data yang diinputkan.
	9. Sistem menampilkan pesan “Berhasil ubah data pegawai”
S3- Hapus Data Pegawai	
1. Aktor mengklik tombol Hapus.	
	2. Sistem menkonfirmasi hapus data
	3. Sistem menampilkan pesan konfirmasi untuk menghapus data.
4. Aktor mengklik pilihan ok.	
	5. Sistem membuka koneksi <i>database</i>
	6. Sistem menghapus data Pegawai
	7. Sistem menampilkan pesan “Berhasil hapus data pegawai”
Skenario Alternatif	
S-1 5a: Sistem akan menampilkan pesan apabila ada data yang tidak terisi dan proses penyimpanan data dibatalkan	
S-1 5b: Sistem memberikan kesempatan untuk mengulang kembali	
S-2 6a: Sistem akan menampilkan pesan apabila ada data yang tidak lengkap dan proses penyimpanan data dibatalkan	

S-2 6b: Sistem memberikan kesempatan untuk mengulang kembali. S-3 4a: Jika aktor menjawab batal, sistem kembali menampilkan tabel Pegawai	
Post Condition	Aktor berhasil mengolah data Pegawai untuk menambah, mengubah dan menghapus data Pegawai pada <i>database</i>

4. Deskripsi Use case Mengelola Data Presensi

Deskripsi *use case* mengelola Data Presensi merupakan aktivitas yang dilakukan aktor dalam mengakses Data Presensi. Seperti yang terlihat pada tabel 4.4.

Tabel 4.4 Deskripsi Use case Melihat Data Presensi

Nama	Mengelola Data Presensi	
Aktor	Admin	
Deskripsi	Aktor melihat informasi data presensi	
Exception	-	
Pre Condition	Aktor telah membuka aplikasi	
Aktor		Sistem
Skenario Normal		
1. Aktor memilih menu presensi yang ada		
		2. Membuka koneksi ke <i>database</i>
		3. Menampilkan tampilan data presensi pegawai
Skenario Alternatif		
-		
Post Condition	Aktor berhasil melihat menu presensi pegawai	

5. Deskripsi Use Case Melihat Laporan

Deskripsi *Use Case* untuk melihat laporan ini akan menjelaskan fungsional dari *Use Case* membuat laporan transaksi yang dilakukan oleh admin. Seperti yang terlihat pada tabel 4.5 :

Tabel 4.5 Deskripsi Use Case Melihat Laporan

Nama	Melihat Laporan
Aktor	Admin
Deskripsi	Aktor melihat laporan transaksi
Exception	-
Pre Condition	Aktor telah <i>login</i> ke dalam sistem sebagai admin
Aktor	Sistem
Skenario Normal	
1. Aktor mengklik menu laporan.	
	2. Sistem menampilkan halaman laporan presensi
3. Aktor memasukan periode presensi dan memilih pegawai	
4. Aktor mengklik tombol cari.	
	5. Sistem membuka koneksi ke database
	6. Sistem melakukan pencarian data presensi pegawai berdasarkan periode
	7. Sistem menampilkan data presensi
Skenario Alternatif	
-	
Post Condition	Aktor melihat laporan.

6. Deskripsi Use Case Presensi

Deskripsi *Use Case* untuk presensi menjelaskan fungsional dari *Use Case* presensi masuk yang dilakukan oleh pegawai. Seperti yang terlihat pada tabel 4.6 :

Tabel 4.6 Deskripsi Use Case Presensi

Nama	Presensi
Aktor	Pegawai
Deskripsi	Aktor melakukan presensi
Exception	-
Pre Condition	Aktor telah <i>login</i> ke dalam aplikasi

Aktor	Sistem
Skenario Normal	
1. Aktor mengklik menu presensi	
	2. Sistem menampilkan tampilan scan Qr Code
3. Aktor mengarahkan ke kode QR Code	
	4. Sistem memvalidasi QR Code yang di scan dan Berhasil melakukan presensi
	5. Sistem membuka koneksi ke database
	6. Sistem menyimpan data presensi masuk
	7. Sistem menampilkan pesan presensi masuk berhasil
Skenario Alternatif	
5.a : Sistem menampilkan pesan Kode yang di Scan salah	
5.b : Sistem memberikan kesempatan kembali untuk melakukan presensi	
Post Condition	Aktor berhasil melakukan presensi masuk

7. Deskripsi Use Case Izin/Sakit

Deskripsi Use Case untuk melakukan izin/sakit menjelaskan fungsional dari

Use Case meminta izin/sakit tidak dapat bekerja yang dilakukan oleh pegawai.

Seperti yang terlihat pada tabel 4.7 :

Tabel 4.7 Deskripsi Use Case Izin/Sakit

Nama	Izin/Sakit
Aktor	Pegawai
Deskripsi	Aktor melakukan izin/sakit
Exception	-
Pre Condition	Aktor telah <i>login</i> ke dalam aplikasi
Aktor	Sistem
Skenario Normal	
1. Aktor mengklik menu izin	
	2. Sistem menampilkan halaman untuk meminta izin
3. Aktor memfoto surat izin atau surat keterangan sakit	
4. Aktor mengklik tombol izin	
	5. Sistem membuka koneksi ke database
	6. Sistem menyimpan data izin pegawai

	7. Sistem menampilkan pesan izin pegawai berhasil
Skenario Alternatif	
-	
Post Condition	Aktor berhasil melakukan izin atau sakit pada sistem

8. Deskripsi Use Case Melihat History Presensi

Deskripsi *Use Case* untuk melihat history presensi ini akan menjelaskan fungsional ketika pegawai melihat data history presensi perbulan. Seperti yang terlihat pada tabel 4.8 :

Tabel 4.8 Deskripsi Use Case Melihat History Presensi

Nama	Melihat History Presensi	
Aktor	Pegawai	
Deskripsi	Aktor melihat history presensi	
Exception	-	
Pre Condition	Aktor telah <i>login</i> ke dalam aplikasi	
Aktor	Sistem	
Skenario Normal		
1. Aktor mengklik menu history presensi		
	2. Sistem menampilkan halaman presensi	
3. Aktor memasukan periode bulan dan tahun		
4. Aktor mengklik tombol cari.		
	5. Sistem membuka koneksi ke database	
	6. Sistem melakukan pencarian data presensi berdasarkan periode	
	7. Sistem menampilkan data presensi	
Skenario Alternatif		
-		
Post Condition	Aktor melihat data history presensi	

9. Deskripsi Use Case Ubah Password

Deskripsi *Use Case* untuk ubah password ini akan menjelaskan fungsional pegawai mengubah *password*. Seperti yang terlihat pada tabel 4.9 :

Tabel 4.9 Deskripsi *Use Case* Ubah *Password*

Nama	Ubah Password	
Aktor	Pegawai	
Deskripsi	Aktor mengubah password	
Exception	-	
Pre Condition	Aktor telah <i>login</i> ke dalam aplikasi	
Aktor		Sistem
Skenario Normal		
1. Aktor mengklik menu akun		
	2. Sistem menampilkan data akun	
3. Aktor memilih menu ubah password		
	4. Sistem menampilkan form halaman password	
5. Aktor menginputkan password lama dan password baru		
	6. Sistem membuka koneksi database	
	7. Sistem memvalidasi password lama dan password lama sesuai	
	8. Sistem menyimpan password baru	
	9. Sistem menampilkan pesan ubah password.	
Skenario Alternatif		
7a. sistem menampilkan pesan gagal ganti password, password lama anda salah		
7b. sistem memberikan kesempatan kembali untuk mengganti password		
Post Condition	Aktor berhasil mengubah password	

10. Deskripsi *Use Case* Logout

Rincian fungsionalitas dari *Use Case Logout* terlihat pada tabel 4.10 berikut :

Tabel 4.10 Deskripsi *Use Case* Logout

Nama	<i>Logout</i>
Aktor	Pegawai dan Admin
Deskripsi	Proses yang dilakukan oleh aktor untuk melihat <i>logout</i>

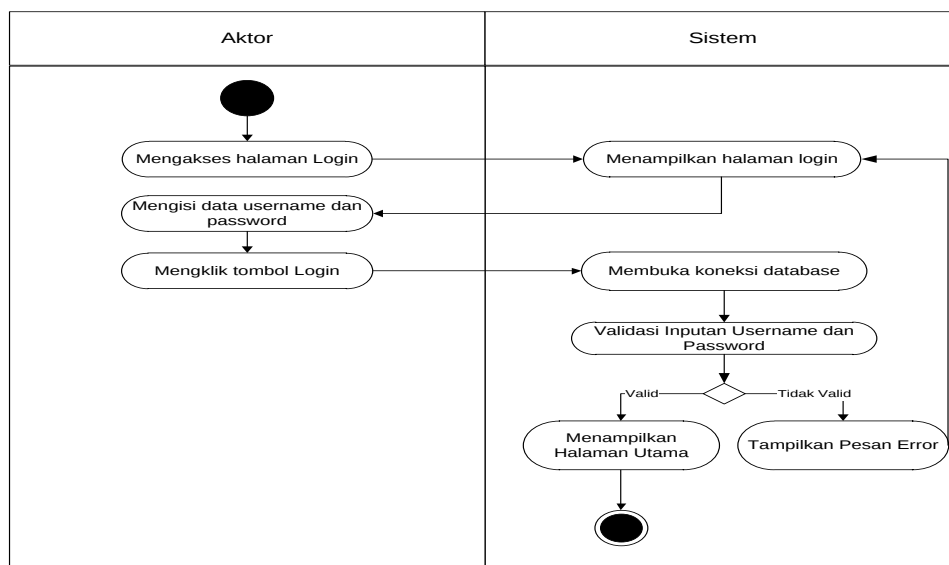
Exception	-
Pre Condition	Aktor telah <i>login</i> kedalam sistem
Aktor	Sistem
Skenario Normal	
1. Aktor memilih menu <i>Logout</i>	
	2. Sistem menghapus seluruh <i>session</i>
Post Condition	Aktor telah keluar dari sistem

4.2.3 Activity Diagram

Dengan menggunakan Activity Diagram, proses bisnis yang kompleks dapat divisualisasikan dengan lebih sederhana sehingga mudah dipahami oleh tim pengembang, pemangku kepentingan, atau pihak terkait lainnya.

1. Activity Diagram Login

Activity Diagram Login menggambarkan aktifitas Aktor untuk melakukan *login*, dengan cara memasukan *username* dan *name* dan *password*. *Activity Diagram Login* ditunjukkan pada Gambar 4.3.



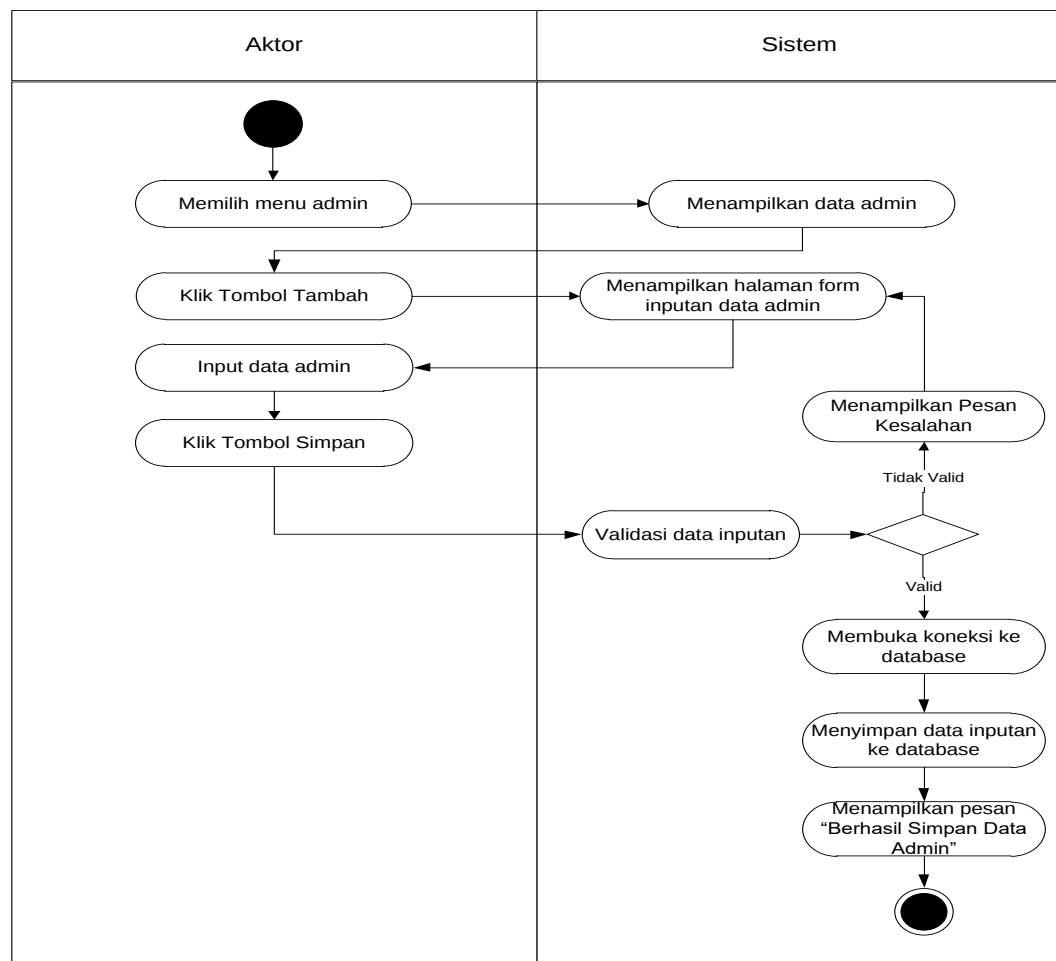
Gambar 4.3 Activity Diagram Login

2. Activity Diagram Mengelola Data Admin

Activity Diagram mengelola data Admin yang ditunjukkan pada Gambar 4.4 sampai dengan Gambar 4.6 di bawah ini menjelaskan aktivitas sistem saat menambah, mengubah dan menghapus data Admin.

a. Activity Diagram Menambah Data Admin.

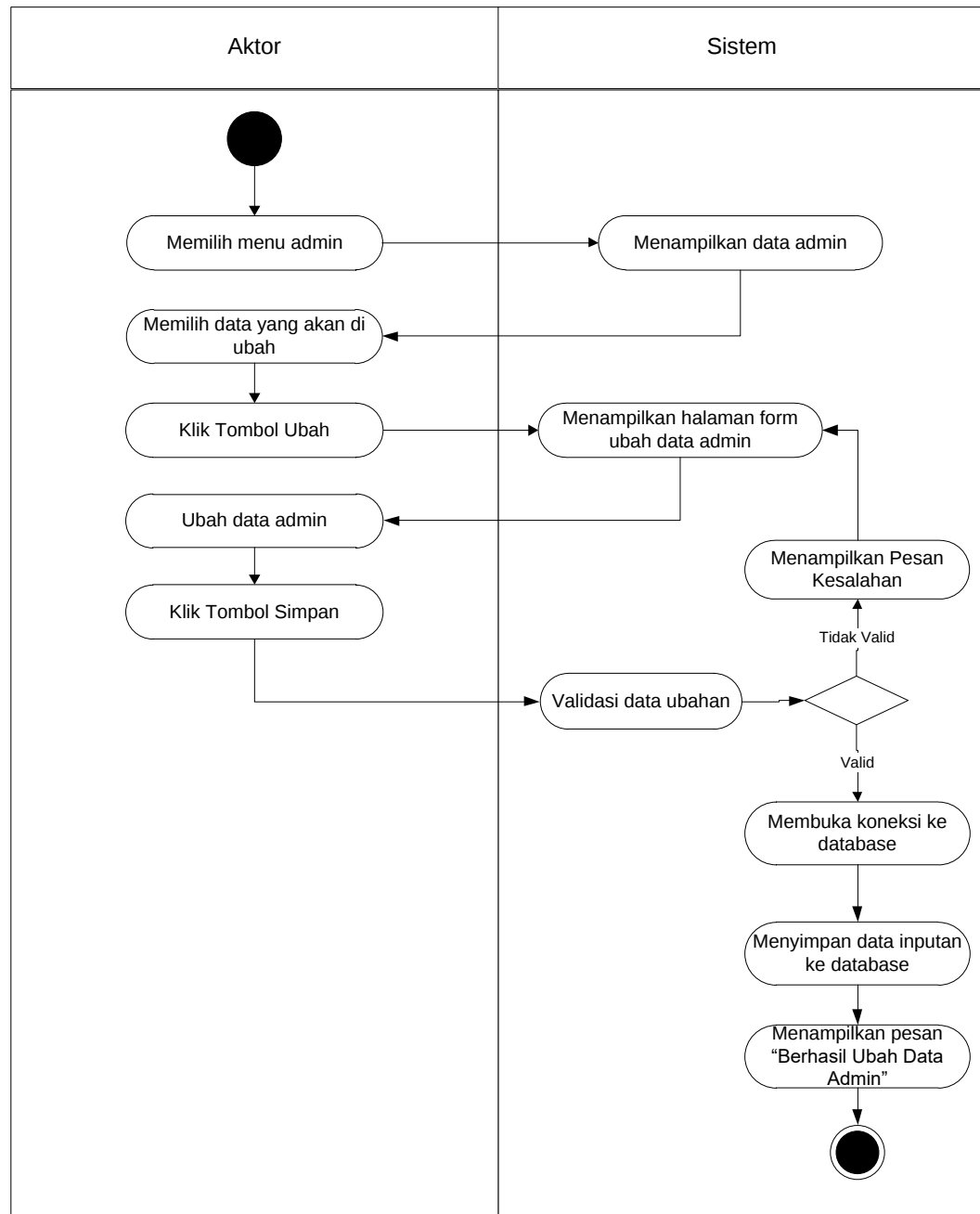
Activity Diagram menambah data admin menggambarkan aktifitas Admin dalam proses menambah data admin baru ke dalam sistem. Untuk lebih jelasnya dapat dilihat pada gambar 4.4.



Gambar 4.4 Activity Diagram Menambah Data Admin

b. *Activity Diagram Mengubah Admin*

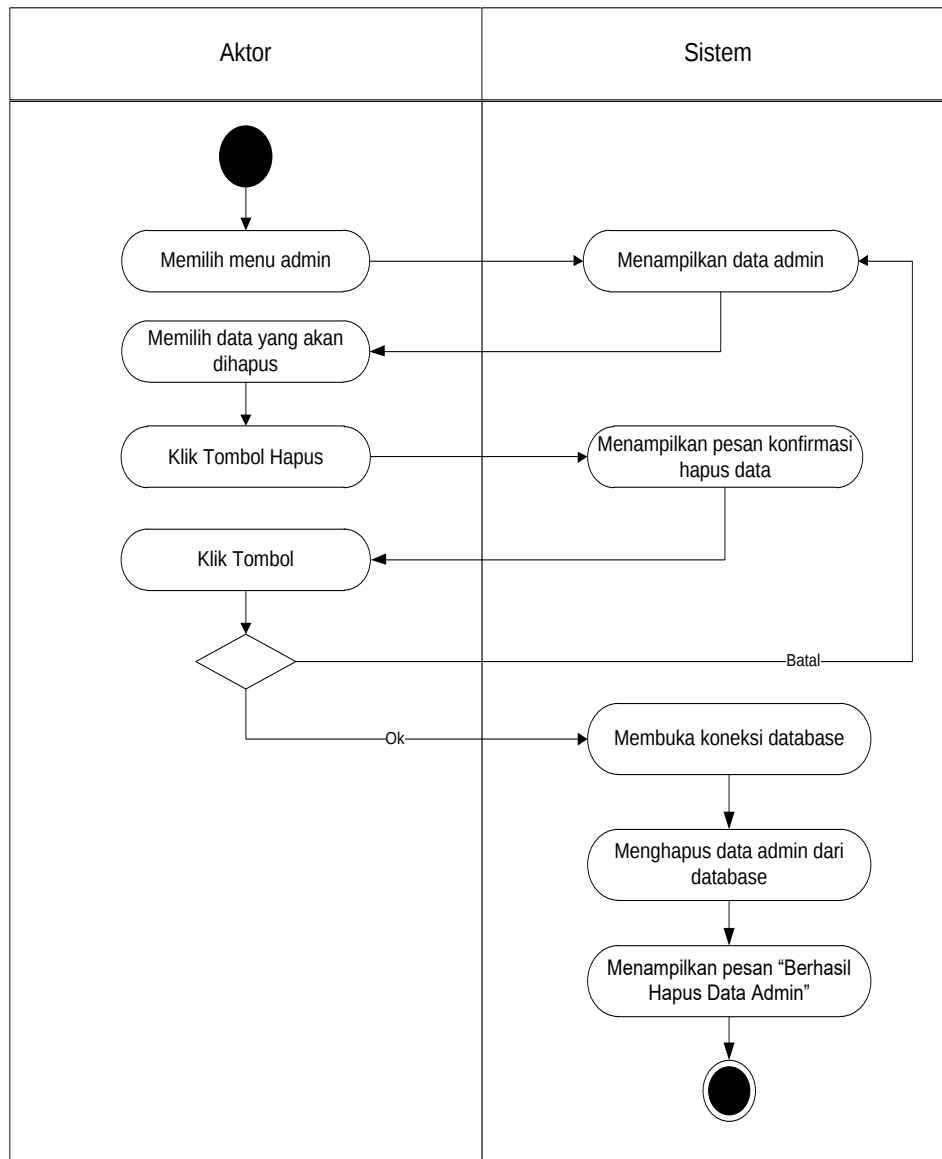
Activity Diagram mengubah admin merupakan langkah – langkah dari Admin untuk mengubah admin dari dalam sistem. Untuk lebih jelasnya dapat dilihat pada gambar 4.5.



Gambar 4.5 Activity Diagram Mengubah Data Admin

c. *Activity Diagram Menghapus Data Admin*

Activity Diagram menghapus admin merupakan langkah – langkah dari Admin untuk menghapus data admin dari dalam sistem. Untuk lebih jelasnya dapat dilihat pada gambar 4.6.



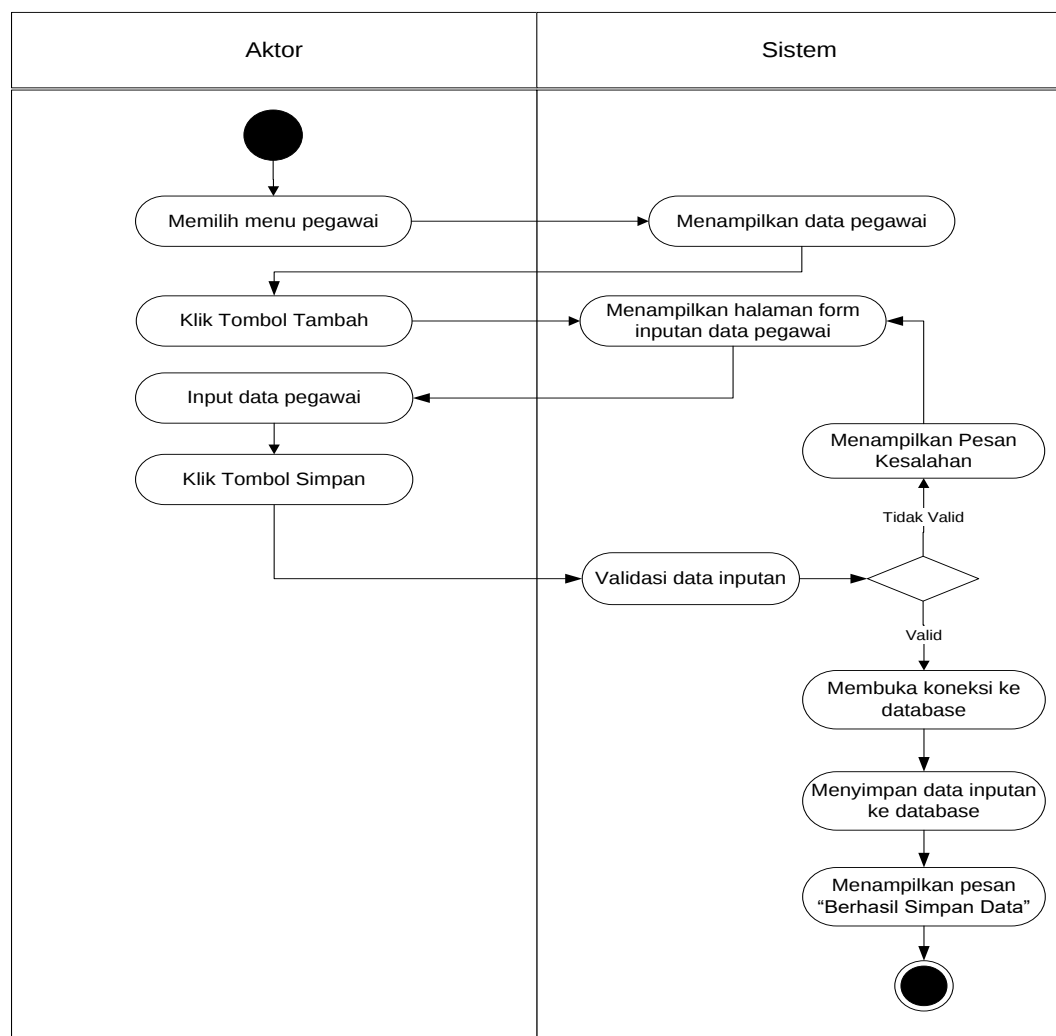
Gambar 4.6 Activity Diagram Menghapus Data Admin

3. Activity Diagram Mengelola Data Pegawai

Activity Diagram mengeloladata Pegawai yang ditunjukkan pada Gambar 4.7 sampai dengan Gambar 4.9 di bawah ini menjelaskan aktivitas sistem saat menambah dan menghapus data Pegawai.

a. Activity Diagram Menambah Data Pegawai

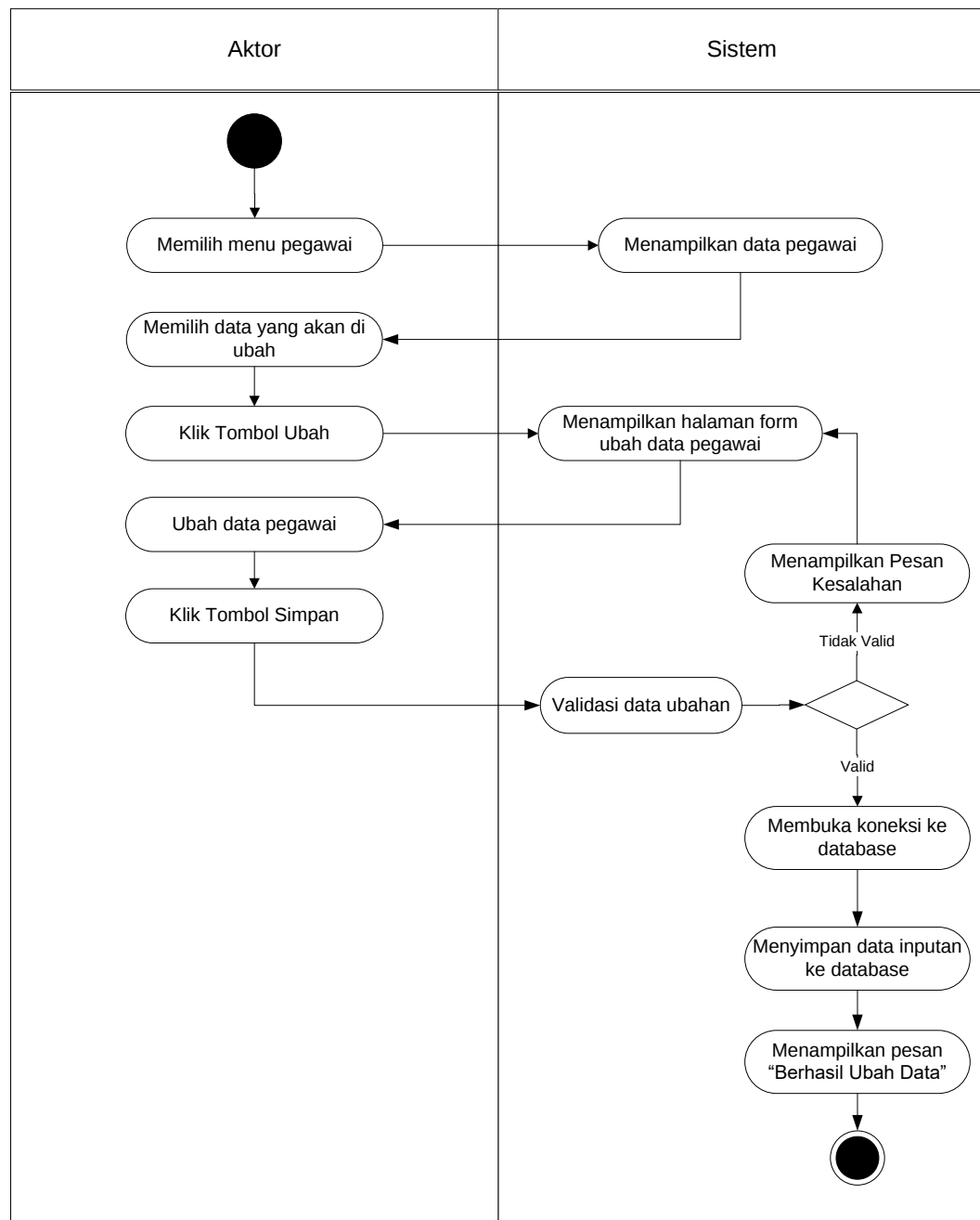
Activity Diagram menambah data Pegawai menggambarkan aktifitas Admin dalam proses menambah data Pegawai baru ke dalam sistem. Untuk lebih jelasnya dapat dilihat pada gambar 4.8.



Gambar 4.7 Activity Diagram Menambah Data Pegawai

b. *Activity Diagram Mengubah Data Pegawai*

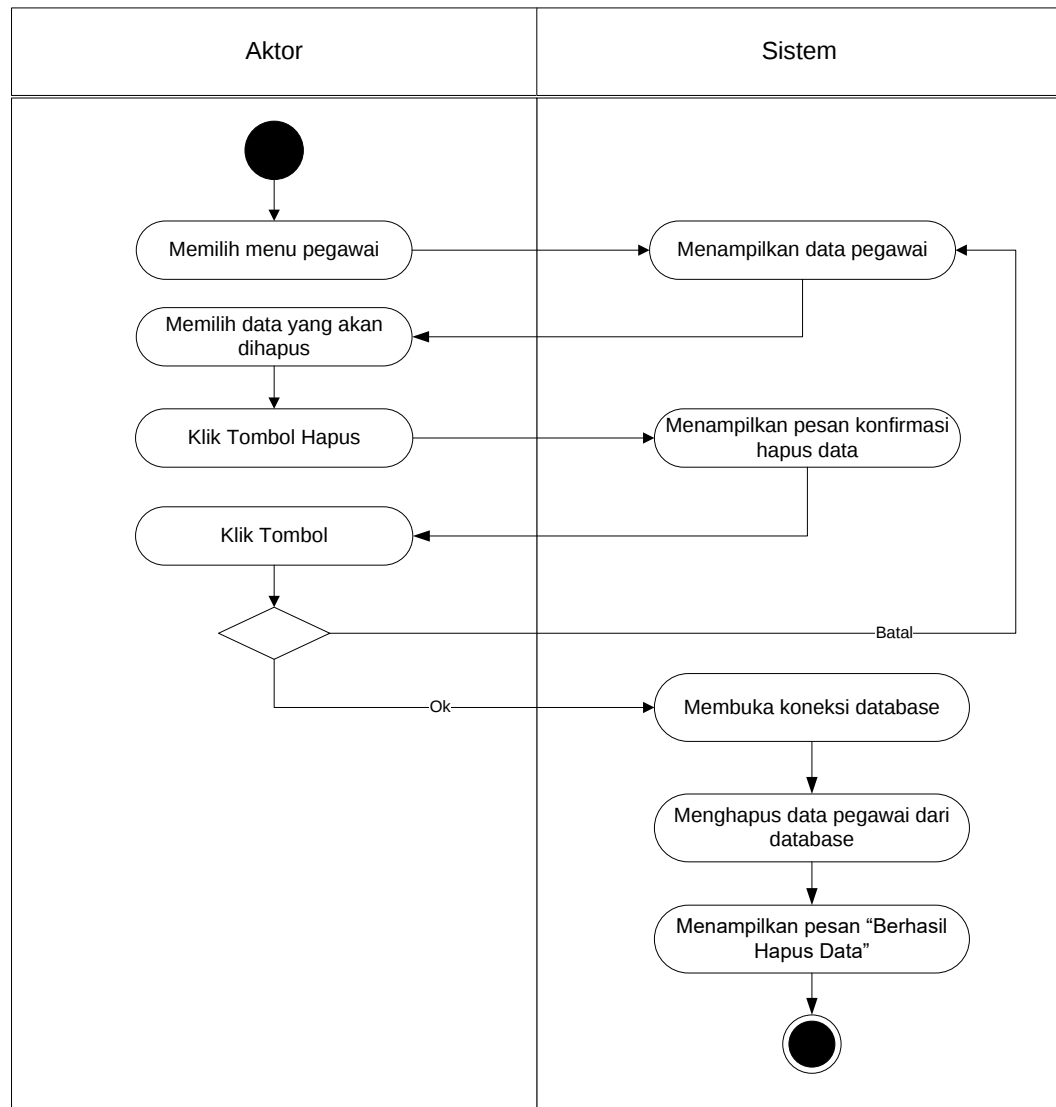
Activity Diagram mengubah Pegawai merupakan langkah – langkah dari Admin untuk mengubah Pegawai dari dalam sistem. Untuk lebih jelasnya dapat dilihat pada gambar 4.8.



Gambar 4.8 Activity Diagram Mengubah Data Pegawai

c. *Activity Diagram Menghapus Data Pegawai*

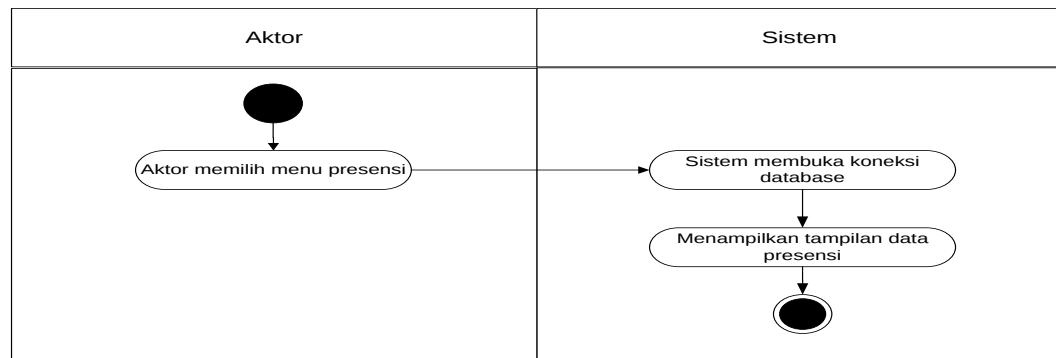
Activity Diagram menghapus Pegawai merupakan langkah – langkah dari Admin untuk menghapus data Pegawai dari dalam sistem. Untuk lebih jelasnya dapat dilihat pada gambar 4.9.



Gambar 4.9 Activity Diagram Menghapus Data Pegawai

4. *Activity Diagram* Melihat Data Presensi

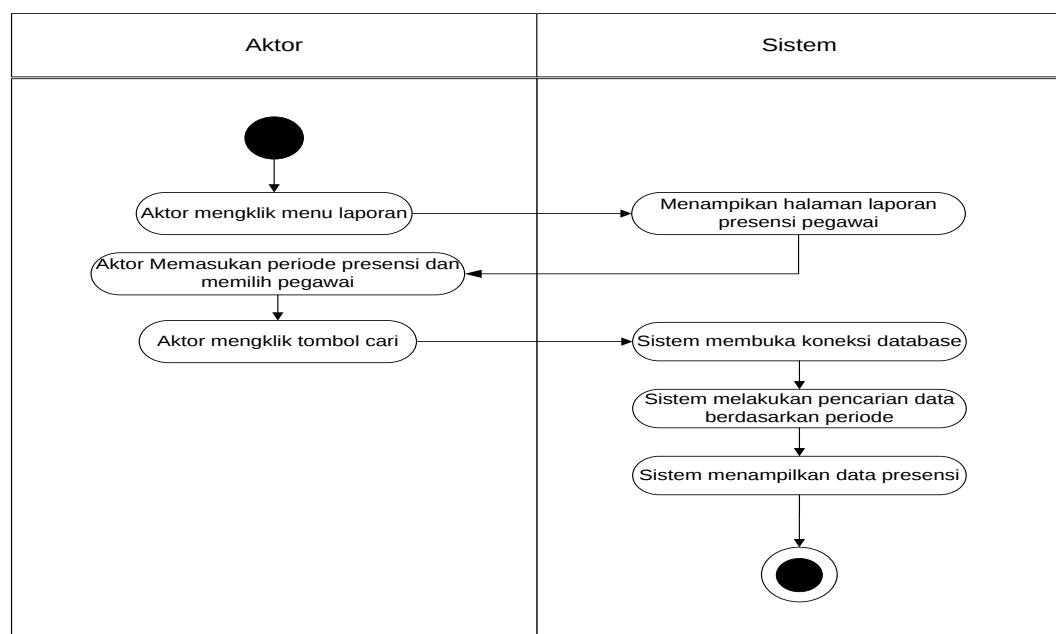
Activity Diagram melihat data presensi merupakan aktivitas admin pada saat ingin melihat informasi data presensi pegawai. Seperti yang terlihat pada gambar 4.10.



Gambar 4.10 *Activity Diagram* Melihat Data Presensi

5. *Activity Diagram* Melihat Laporan

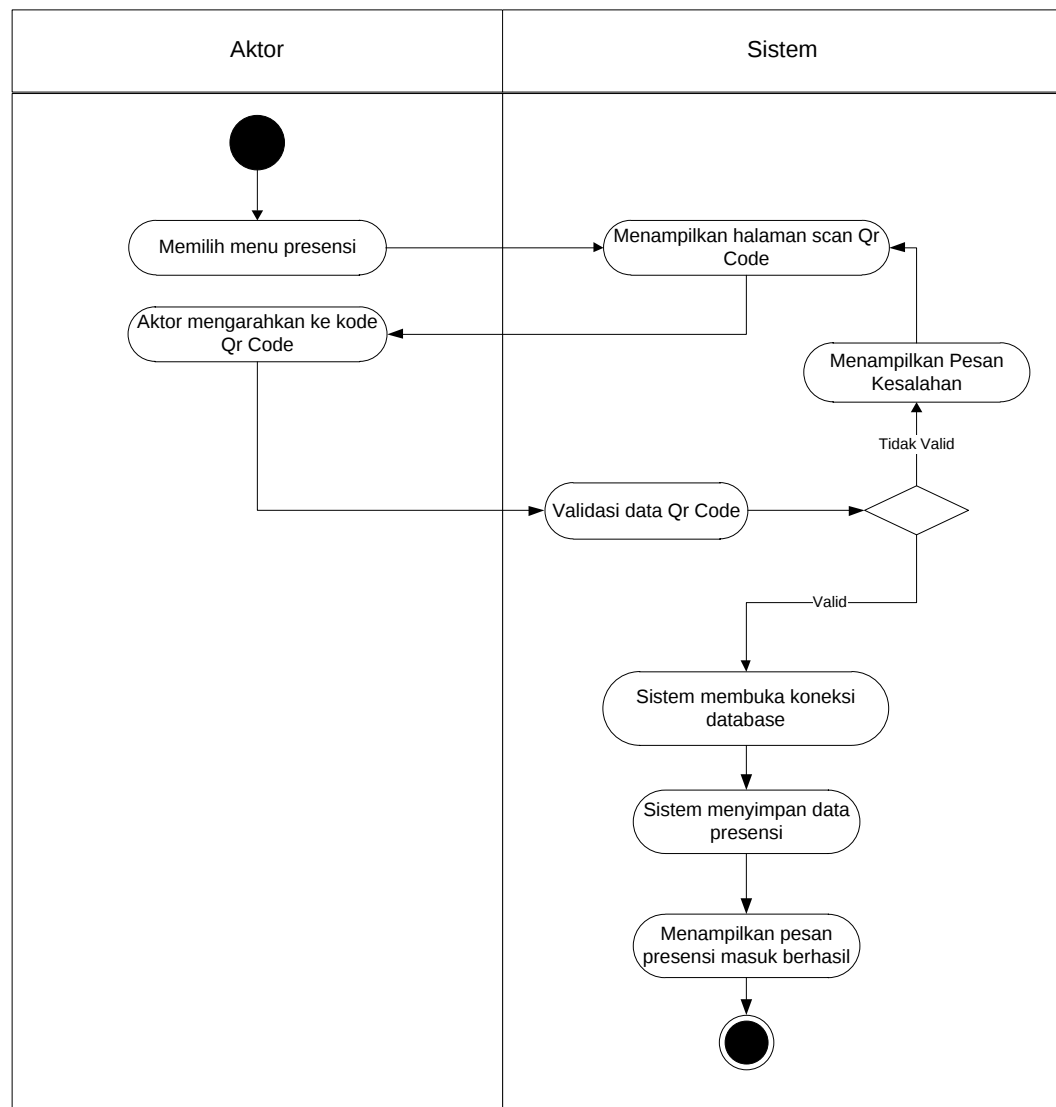
Activity Diagram melihat laporan ini menjelaskan aktivitas sistem saat lihat laporan. Seperti yang terlihat pada gambar 4.11.



Gambar 4.11 *Activity Diagram* Melihat Laporan

6. Activity Diagram Mengelola Data Presensi

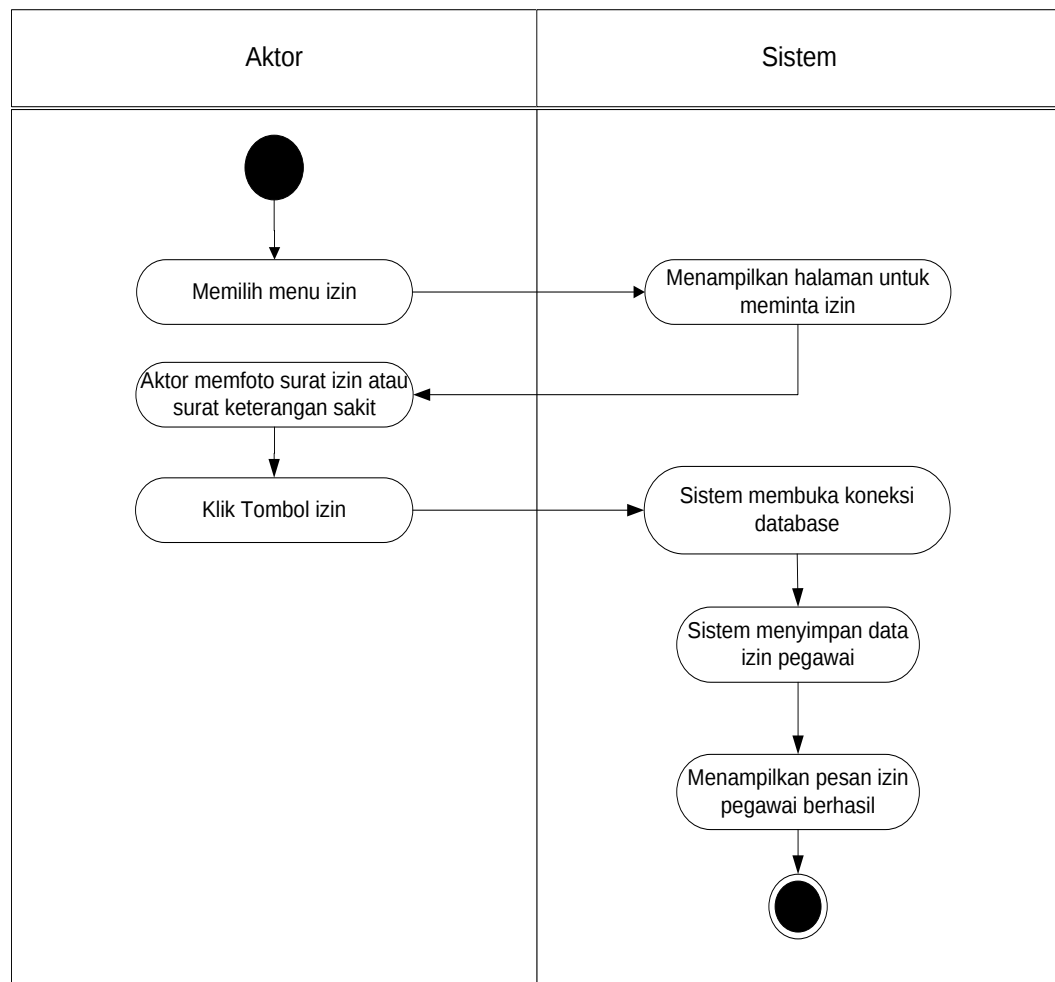
Activity Diagram mengelola data Presensi menjelaskan aktivitas sistem saat pegawai melakukan presensi. Seperti ditunjukkan pada Gambar 4.12.



Gambar 4.12 Activity Diagram Mengelola Data Presensi

7. *Activity Diagram* Mengelola Data Izin/Sakit

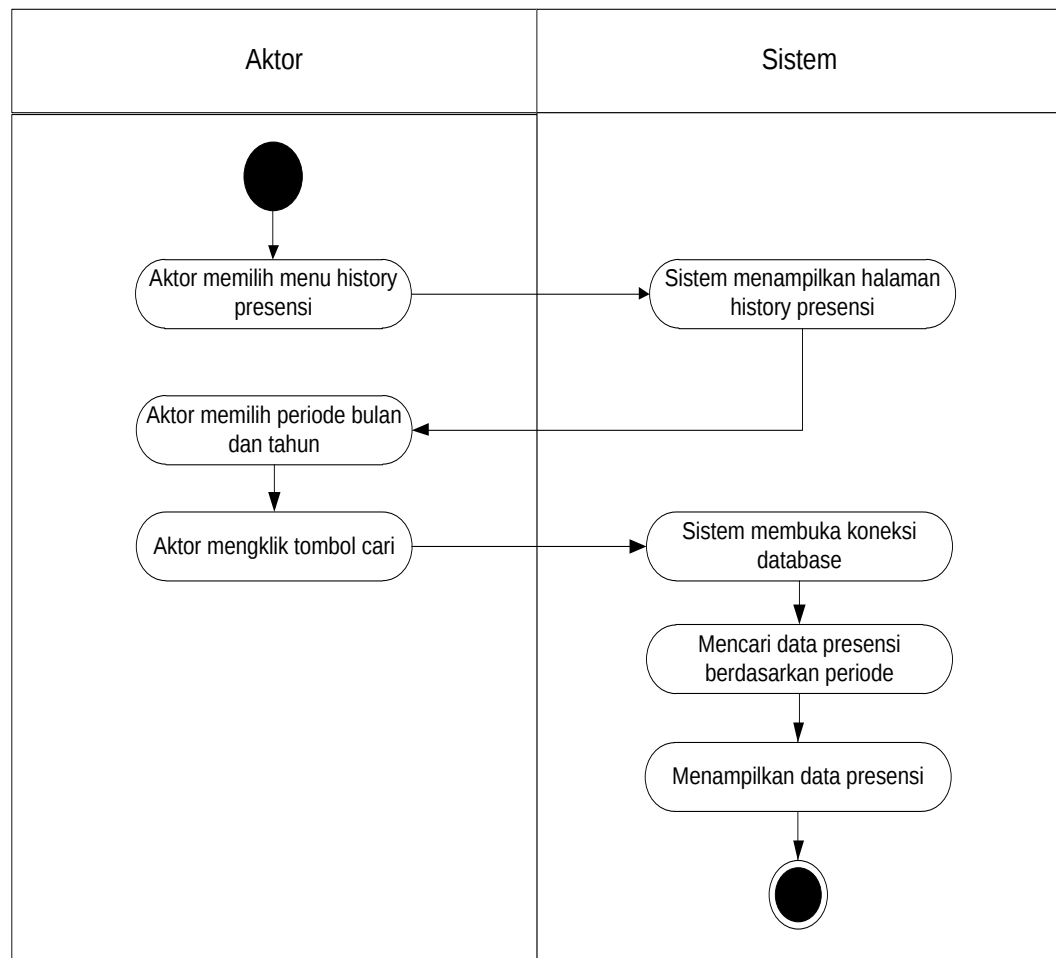
Activity Diagram mengelola data izin/sakit menjelaskan aktivitas sistem saat pegawai meminta izin yang dilakukan melalui aplikasi. Seperti ditunjukkan pada Gambar 4.13.



Gambar 4.13 *Activity Diagram* Mengelola Data Izin/Sakit

8. *Activity Diagram* Melihat *History* Presensi

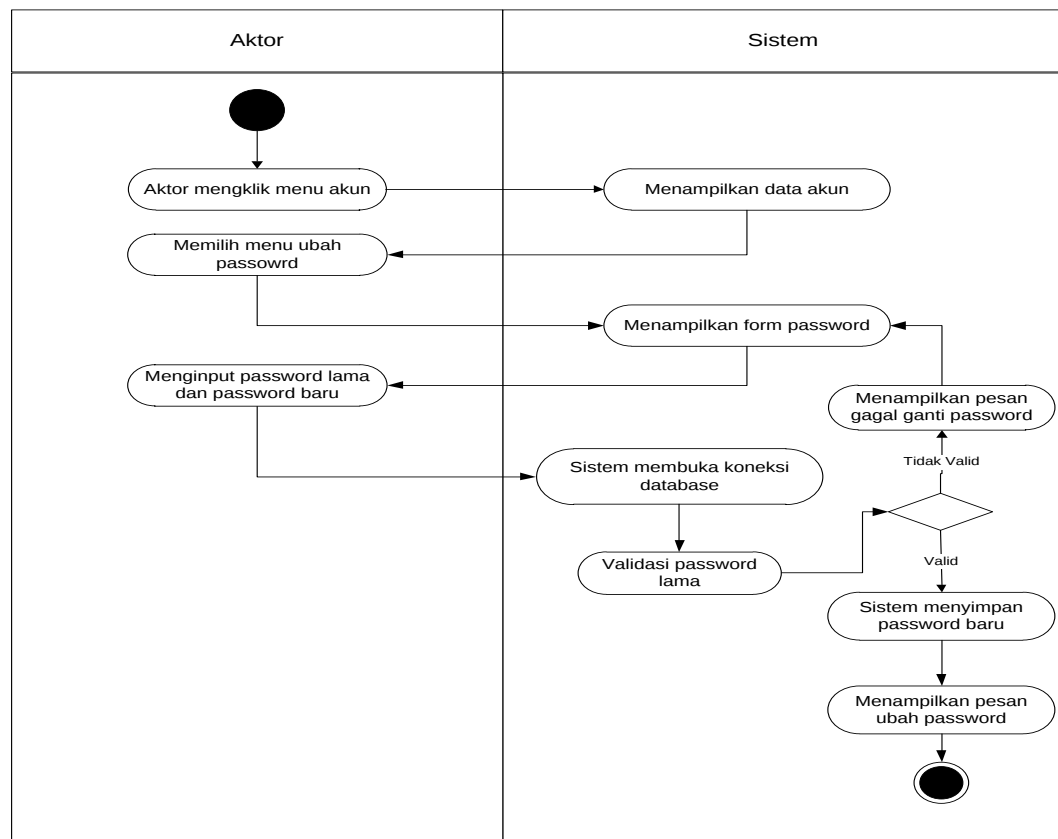
Activity Diagram melihat *history* presensi menjelaskan fungsional ketika karyawan melihat *history* presensi selama periode bulan. Seperti yang terlihat pada gambar 4.14.



Gambar 4.14 Activity Diagram Melihat History Presensi

9. Activity Diagram Ubah Password

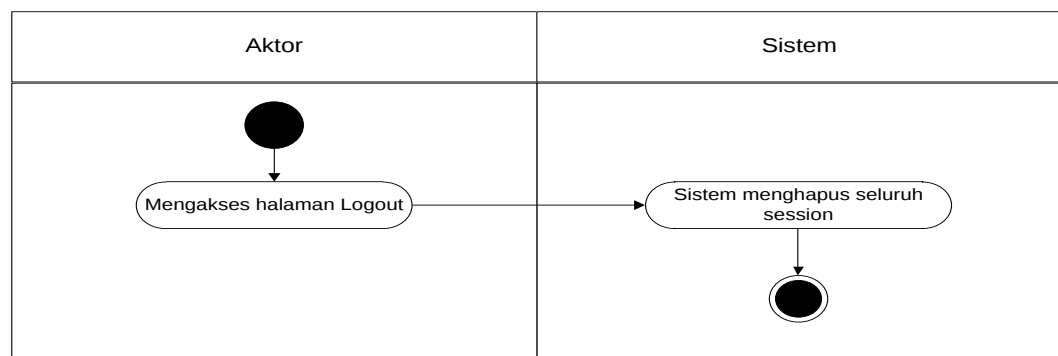
Activity Diagram ubah password menjelaskan fungsional ketika karyawan mengubah password. Seperti yang terlihat pada gambar 4.15.



Gambar 4.15 Activity Diagram Ubah Password

10. Activity Diagram Logout

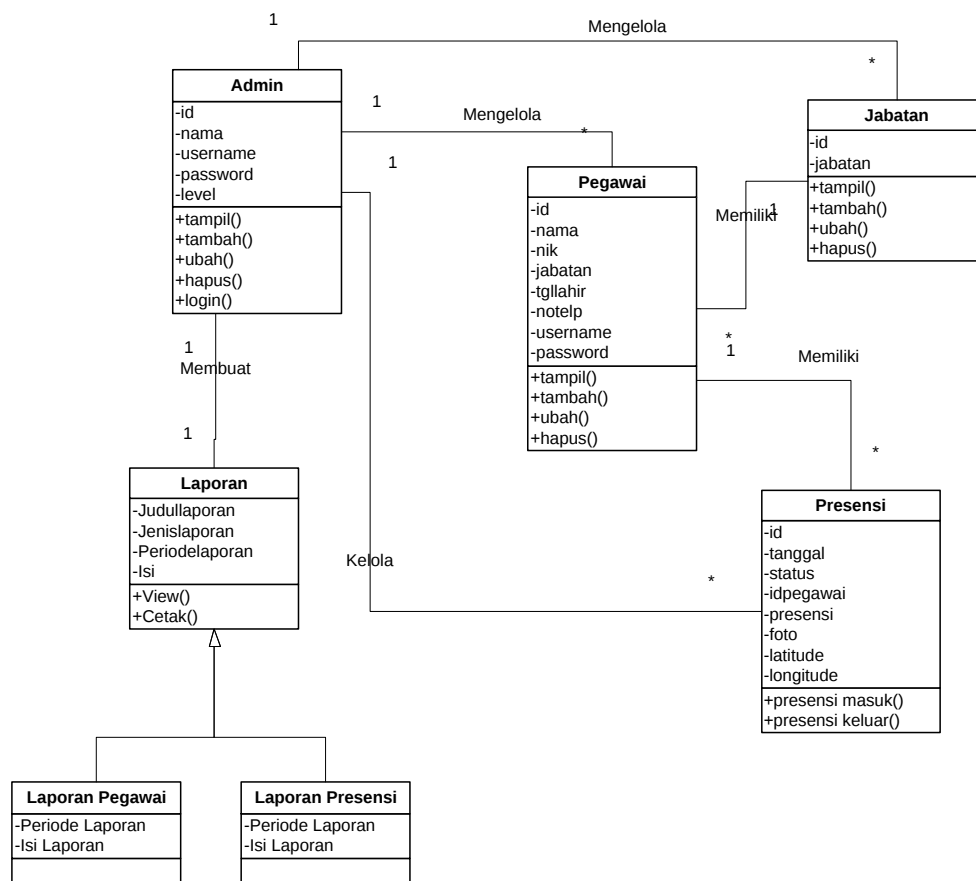
Activity Diagram logout yang dilakukan untuk keluar dari sistem. Seperti yang terlihat pada gambar 4.16.



Gambar 4.16 Activity Diagram Logout

4.2.4 Class Diagram

Class Diagram menggambarkan perilaku dan keadaan dengan menghubungkan antara kelas-kelas. Tampilan dari *class diagram* dapat dilihat pada gambar 4.17.



Gambar 4.17 Class Diagram

4.3 RANCANGAN INPUT DAN OUTPUT

Rancangan *input* dan *output* merupakan rancangan dari form-form masukan dari sistem dan keluaran dari sistem. Berikut adalah rancangan *input* dan *output*.

Berikut adalah gambar *input* dan *output* data presensi pegawai seperti terlihat pada Gambar 4.18.

PERUSAHAAN UMUM DAERAH AIR MINUM
TIRTA MUARO JAMBI
UNIT PELAYANAN METRO MENDALO
Alamat Kantor: Jl. Lintas Jambi Ma. Bulian Km 14 Mendalo Darat

REKAP ABSENSI PEGAWAI

BULAN : 7 Januari 2025

NO	NAMA		S	I	A	PARAF
1	Abunjani.SE	Ka. Unit				<i>[Signature]</i>
1	Mukhlis	Kasi Adm & Keuangan				<i>[Signature]</i>
2	Khadafi Tam Redho	Kasir				<i>[Signature]</i>
3	Ikah Atikah	Kasir				<i>[Signature]</i>
4	Aida Nindya	Kasir				<i>[Signature]</i>
5	Haryani Nur.SE	Kasubsi Hub Lang				<i>[Signature]</i>
6	Juwairiyah.SE	Staff Hub Lang				<i>[Signature]</i>
7	Rahawi	Staff Hub Lang				<i>[Signature]</i>
8	Della M	Staff Hub Lang				<i>[Signature]</i>
9	M. Supriadi	Staff Hub Lang				<i>[Signature]</i>
10	Angga Putra	Staff Hub Lang/Penagihan				<i>[Signature]</i>
11	Wando Lesmana	Staff Hub Lang/Penagihan				<i>[Signature]</i>
12	Askolani	Staff Hub Lang/Penagihan				<i>[Signature]</i>
13	Sekar Dwi Rama Siwi	Staff Hub Lang/Penagihan				<i>[Signature]</i>
14	Askolani	Staff Hub Lang/Penagihan				<i>[Signature]</i>
15	Rama Rifki Fadli	Staff Hub Lang/Penagihan				<i>[Signature]</i>
16	M. Rapi Mulyawan	Staff Hub Lang/Penagihan				<i>[Signature]</i>
17	Kiki Rifki E	Staff Hub Lang/Penagihan				<i>[Signature]</i>
18	Yun Fadli	Staff Hub Lang/Penagihan				<i>[Signature]</i>
19	Boni	Staff Hub Lang/Penagihan				<i>[Signature]</i>
20	Syahrullah	Staff Hub Lang/Penagihan				<i>[Signature]</i>
21	Eprija	Trans & Distr	-	-	-	<i>[Signature]</i>
22	Haryanto	Trans & Distr	-	-	-	<i>[Signature]</i>
23	Erpin	Kasubsi Produksi	-	-	-	<i>[Signature]</i>
24	Supriadi	Trans & Distr	-	-	-	<i>[Signature]</i>
25			-	-	-	

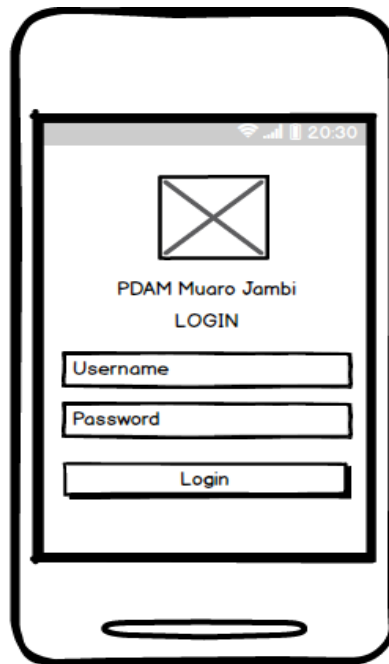
Gambar 4.18 Input Dan Output Data Presensi

4.3.1 Rancangan Input

Rancangan rancangan tampilan *input* dari sistem yang akan dibangun adalah sebagai berikut:

1. Rancangan Form Login Pegawai

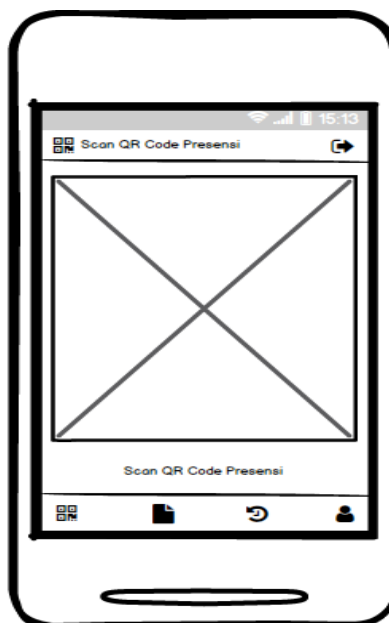
Merupakan halaman yang menampilkan dari form login untuk pegawai. Seperti yang terlihat pada gambar 4.19.



Gambar 4.19 Rancangan Form *Login* Pegawai

2. Rancangan Halaman Scan Presensi

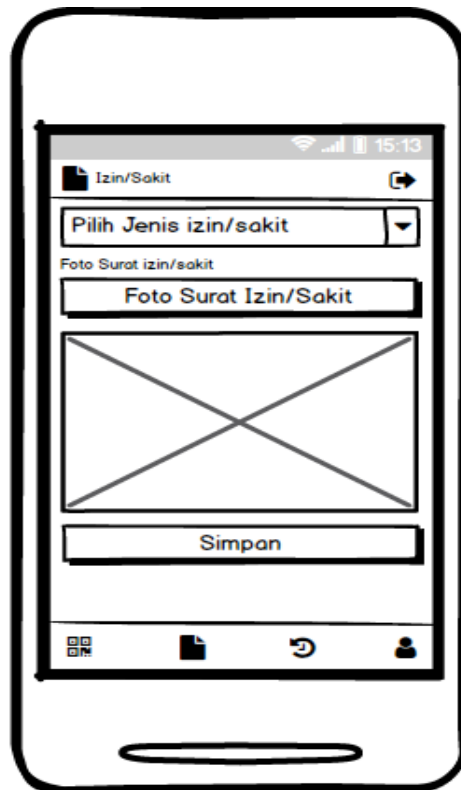
Merupakan halaman yang dapat digunakan untuk menambah data presensi masuk. Seperti yang terlihat pada gambar 4.20.



Gambar 4.20 Rancangan Halaman Scan Presensi

3. Rancangan Halaman Izin/Sakit

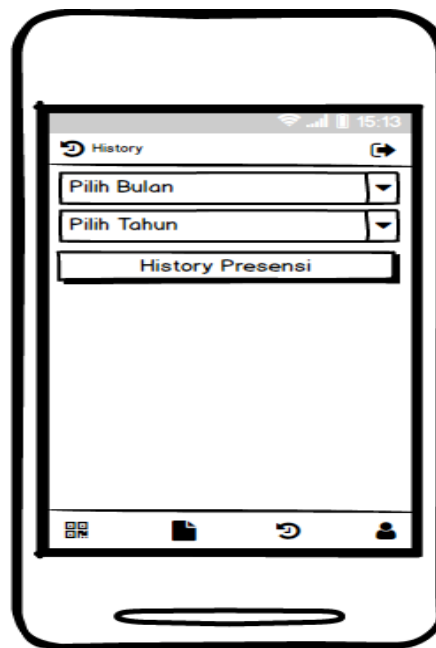
Merupakan halaman yang dapat digunakan untuk melakukan izin/sakit pada presensi. Seperti yang terlihat pada gambar 4.21.



Gambar 4.21 Rancangan Halaman Izin/Sakit

4. Rancangan Halaman History Presensi

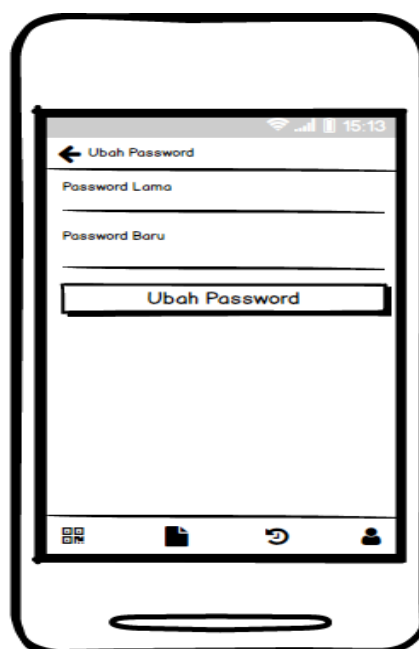
Merupakan halaman yang dapat digunakan untuk melihat data history presensi pegawai berdasarkan periode perbulan. Seperti yang terlihat pada gambar 4.22.



Gambar 4.22 Rancangan Halaman Data Rekap Presensi

5. Rancangan Halaman Ubah Password

Merupakan halaman yang dapat digunakan untuk melakukan perubahan password. Seperti yang terlihat pada gambar 4.23.



Gambar 4.23 Rancangan Halaman Ubah Password

6. Rancangan Halaman Scan Qr Code Presensi

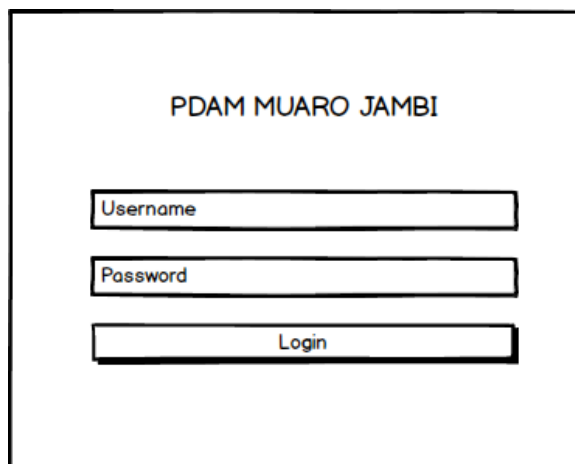
Berikut ini merupakan rancangan halaman scan Qr Code Presensi. Seperti terlihat pada gambar 4.24 :



Gambar 4.24 Rancangan Halaman Qr Code Presensi

7. Rancangan Halaman *Login* Admin

Berikut adalah rancangan dari halaman *login* admin yang dapat digunakan admin untuk masuk kedalam sistem admin yang dapat digunakan untuk mengelola menu menu yang ada pada sistem. Seperti terlihat pada gambar 4.25 :

The image shows a rectangular frame representing a web page. At the top center, the text "PDAM MUARO JAMBI" is displayed. Below the text are three input fields. The first field is labeled "Username", the second field is labeled "Password", and the third field is labeled "Login".

Gambar 4.25 Rancangan Halaman *Login* Admin

8. Rancangan Halaman Form Data Pegawai

Rancangan halaman form data pegawai merupakan halaman yang dapat diakses oleh admin untuk menambah data pegawai. Seperti terlihat pada gambar 4.26 :

Gambar 4.26 Rancangan Halaman Form Data Pegawai

9. Rancangan Halaman Form Data Pengguna

Rancangan halaman form data pengguna merupakan halaman yang dapat diakses oleh admin untuk menambah dan mengubah pengguna aplikasi. Seperti terlihat pada gambar 4.27.

Gambar 4.27 Rancangan Halaman Form Data Pengguna

10. Rancangan Halaman Form Presensi

Rancangan halaman form presensi merupakan halaman yang dapat diakses oleh admin menambah data presensi melalui aplikasi berbasis website. Seperti terlihat pada gambar 4.28.

The screenshot shows a web application interface for attendance management. On the left is a sidebar with a menu containing: Dashboard, Presensi, Master Data, Pegawai, Pengguna, Laporan, Pegawai, Presensi Pegawai, and Perbulan. The main content area is titled 'Presensi' and features a 'Kembali' button in the top right corner. The form includes several input fields: 'Tanggal' (Date), 'Jam' (Time), 'Status Presensi' (a dropdown menu with 'Pilih' as the placeholder), and 'Presensi (Masuk / Pulang)' (a dropdown menu with 'Pilih' as the placeholder). Below these are fields for 'Pegawai' (a dropdown menu with 'Pilih' as the placeholder) and 'Foto (Surat Keterangan Sakit/ Izin)' (a 'Choose File' button). A 'Simpan' button is located at the bottom left of the form area.

Gambar 4.28 Rancangan Halaman Form Presensi

4.3.2 Rancangan Output

Rancangan tampilan *output* dari sistem yang akan dibangun adalah sebagai berikut:

1. Rancangan Halaman Mengelola Data Pegawai

Rancangan mengelola data Pegawai merupakan halaman yang dapat diakses oleh admin untuk menambah, mengubah dan menghapus data Pegawai. Seperti terlihat pada gambar 4.29 :

Gambar 4.29 Rancangan Halaman Mengelola Pegawai

2. Rancangan Halaman Mengelola Data Pengguna

Rancangan mengelola data pengguna merupakan halaman yang dapat diakses oleh admin untuk menambah, mengubah dan menghapus data pengguna.

Seperti terlihat pada gambar 4.30 :

Gambar 4.30 Rancangan Halaman Data Pengguna

3. Rancangan Halaman Presensi

Rancangan mengelola data presensi merupakan halaman yang dapat diakses oleh admin untuk melihat data presensi. Seperti terlihat pada gambar 4.31 :

Gambar 4.31 Rancangan Halaman Data Presensi

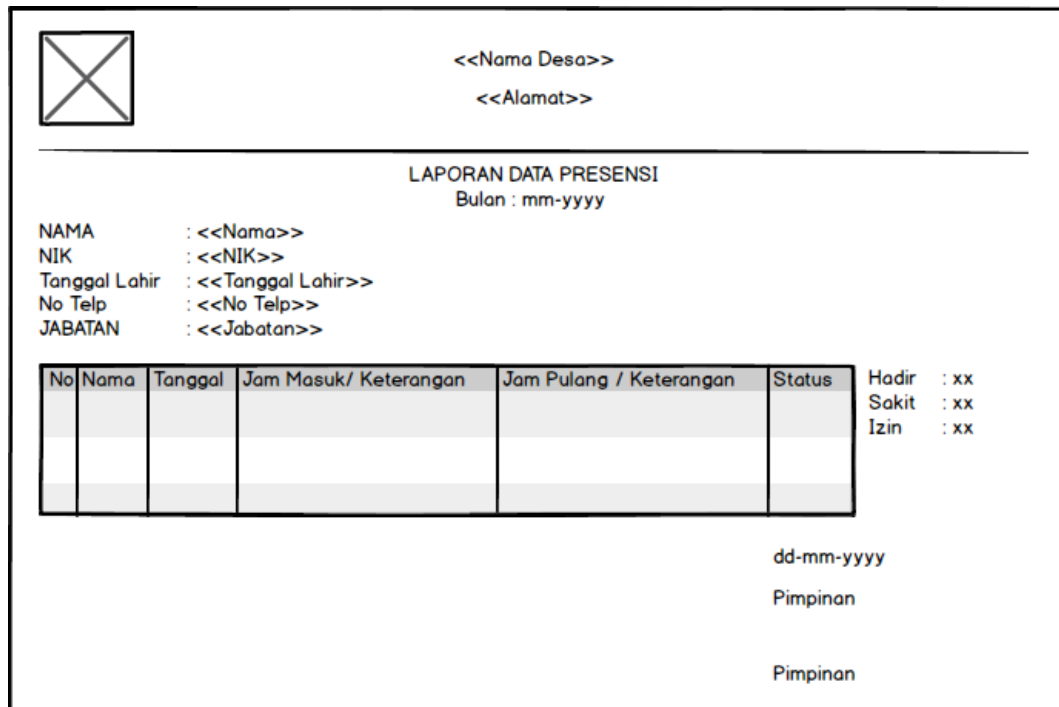
4. Rancangan Laporan Pegawai

Rancangan laporan pegawai ini berfungsi ketika admin ingin melihat data laporan semua pegawai. Tampilan halaman laporan pegawai dapat dilihat pada gambar 4.32.

Gambar 4.32 Rancangan Halaman Laporan Pegawai

5. Rancangan Laporan Presensi

Rancangan laporan presensi pegawai ini berfungsi ketika admin ingin melihat data presensi pegawai periode bulanan. Tampilan halaman laporan presensi dapat dilihat pada gambar 4.33.



X

<<Nama Desa>>
 <<Alamat>>

LAPORAN DATA PRESENSI
 Bulan : mm-yyyy

NAMA : <<Nama>>
 NIK : <<NIK>>
 Tanggal Lahir : <<Tanggal Lahir>>
 No Telp : <<No Telp>>
 JABATAN : <<Jabatan>>

No	Nama	Tanggal	Jam Masuk / Keterangan	Jam Pulang / Keterangan	Status

Hadir : xx
 Sakit : xx
 Izin : xx

dd-mm-yyyy
 Pimpinan

 Pimpinan

Gambar 4.33 Rancangan Halaman Laporan Presensi

4.4 RANCANGAN STRUKTUR DATA

Struktur data yang digunakan dalam sistem yang baru, dirancang dalam beberapa tabel sebagai berikut :

1. Tabel Admin

Tabel admin digunakan untuk menyimpan data-data admin. Nama tabel admin dalam *database* adalah admin. Seperti yang terlihat pada Tabel 4.11.

Tabel 4.11 Rancangan Struktur Tabel Admin

No	Field Name	Type	Length	Keterangan
1	Id	<i>Int</i>	5	Id admin (<i>Primary key</i>)
2	Nama	<i>Varchar</i>	30	Nama lengkap dari admin
3	<i>Username</i>	<i>Varchar</i>	20	<i>Username</i> dari admin
4	<i>Password</i>	<i>Varchar</i>	10	<i>Password</i>
5	<i>Level</i>	<i>Varchar</i>	10	<i>Level</i> dari admin

2. Tabel Jabatan

Tabel jabatan digunakan untuk menyimpan data jabatan pada aplikasi. Seperti yang terlihat pada Tabel 4.12 :

Tabel 4.12 Rancangan Struktur Tabel Jabatan

No	Field Name	Type	Length	Keterangan
1	Id	<i>Int</i>	3	Id (<i>Primary key</i>)
2	Jabatan	<i>Varchar</i>	15	Jabatan

3. Tabel Pegawai

Tabel pegawai merupakan tabel yang digunakan oleh admin untuk menyimpan data pegawai. Nama tabel pegawai dalam *database* adalah pegawai. Seperti yang terlihat pada Tabel 4.13 :

Tabel 4.13 Rancangan Struktur Tabel Pegawai

No	Field Name	Type	Length	Keterangan
1	Id	<i>Int</i>	5	Id (<i>Primary key</i>)
2	Nama	<i>Varchar</i>	35	Nama
3	Nik	<i>Varchar</i>	16	Nik
4	Jabatan	<i>Varchar</i>	20	Jabatan
5	Tgllahir	<i>Date</i>	-	Tanggal lahir
6	Notelp	<i>Varchar</i>	13	No telp
7	<i>Username</i>	<i>Varchar</i>	30	<i>Username</i>
8	<i>Password</i>	<i>Varchar</i>	20	<i>Password</i>

4. Tabel Presensi

Tabel presensi merupakan tabel yang digunakan oleh admin untuk menyimpan data presensi pegawai. Seperti yang terlihat pada Tabel 4.14 :

Tabel 4.14 Rancangan Struktur Tabel Presensi

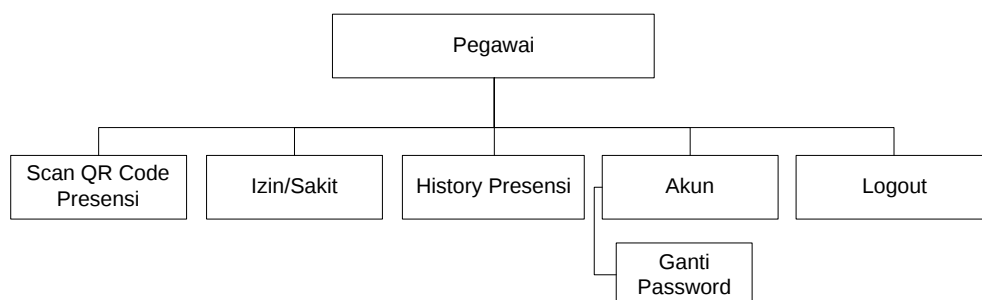
No	Field Name	Type	Length	Keterangan
1	Id	Int	5	Id (<i>Primary key</i>)
2	Tanggal	Datetime	-	Tanggal
3	Status	Enum	Hadir,Izin, Sakit	Status presensii
4	Idpegawai	Int	3	Id Pegawai (<i>Foreign Key</i>)
5	Presensi	Varchar	20	Presensi masuk atau pulang
6	Foto	Longblob	-	Foto surat izin/sakit
7	Latitude	Varchar	20	Latitude koordinat
9	Longitude	Varchar	20	Longitude koordinat

4.5 RANCANGAN STRUKTUR PROGRAM

Rancangan struktur program merupakan bayangan dari sebuah program dimana nanti akan menjelaskan bagian besar struktur program yang akan dibangun oleh penulis, rancangannya terdapat pada gambar dibawah ini.

1. Rancangan Struktur Program Pegawai

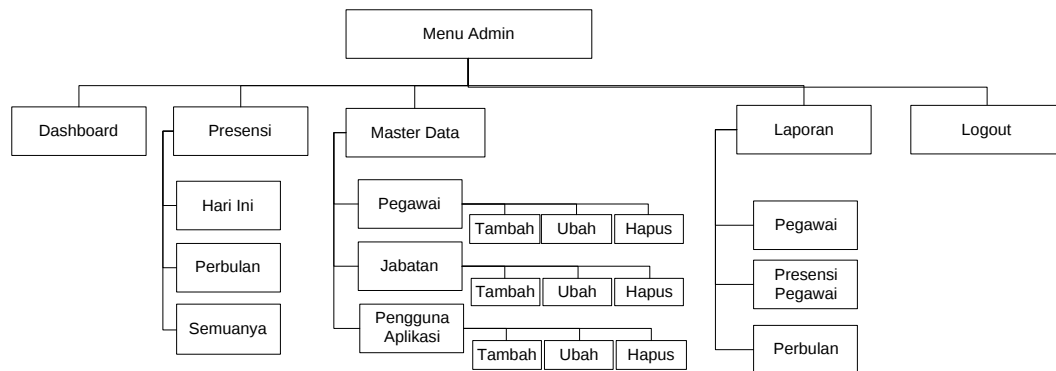
Rancangan struktur program pegawai terdapat menu presensi, izin/sakit, history presensi dan akun. Seperti terlihat pada gambar 4.33.



Gambar 4.34 Rancangan Struktur Program Pegawai

2. Rancangan Struktur Program Admin

rancangan struktur program admin ini terdapat menu yang dapat digunakan oleh admin untuk mengelola data pada sistem preesnsi. Struktur program admin terdapat menu dashboard, presensi, master data, laporan kemudian terdapat menu *logout* untuk keluar dari sistem.



Gambar 4.35 Rancangan Struktur Program Admin

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 IMPLEMENTASI PROGRAM

Implementasi program merupakan hasil dari rancangan pada bab sebelumnya dibuat atau proses menterjemahkan implementasi menjadi hasil tampilan yang dapat difungsikan. Adapun implementasi program untuk Sistem informasi Absensi Pegawai Menggunakan Teknologi QR CODE (*Quick Response Code*) yang dapat dijabarkan, yaitu :

5.1.1 Implementasi *Input*

Implementasi implementasi tampilan *input* dari sistem yang akan dibangun adalah sebagai berikut:

1. Implementasi Form Login Pegawai

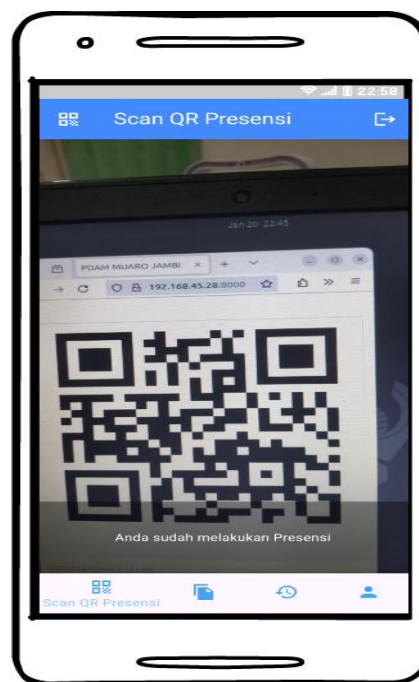
Merupakan halaman yang menampilkan dari form login untuk pegawai. Seperti yang terlihat pada gambar 5.1.



Gambar 5.1 Implementasi Form Login Pegawai

2. Implementasi Halaman Scan Presensi

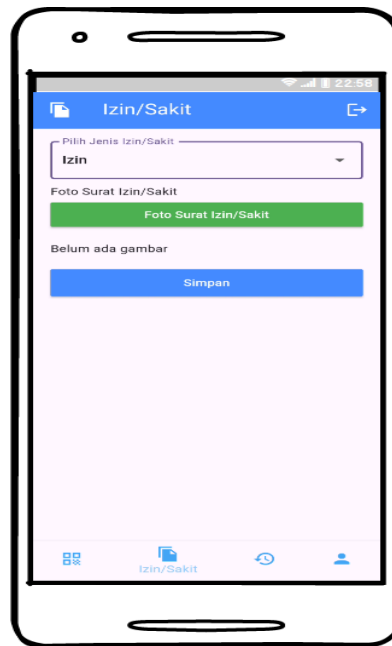
Halaman ini dirancang untuk memudahkan pengguna dalam menambah data presensi. Dengan menggunakan halaman ini, informasi kehadiran dapat dicatat dengan cepat dan efisien. Pengguna hanya perlu memasukkan data yang diperlukan, sehingga proses pencatatan presensi dapat dilakukan dengan lebih terorganisir dan akurat. Seperti yang terlihat pada gambar 5.2.



Gambar 5.2 Implementasi Halaman Data Presensi Masuk

3. Implementasi Halaman Izin/Sakit

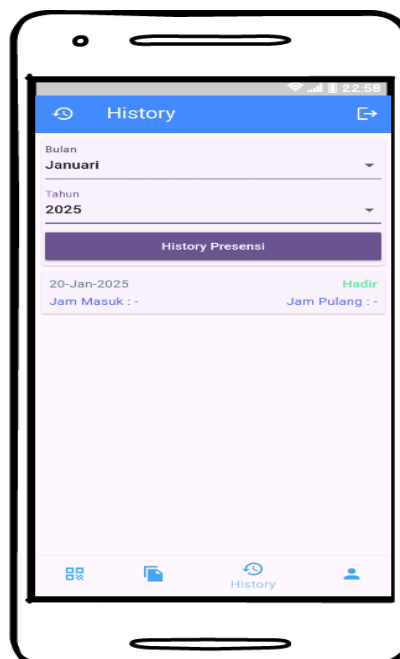
Halaman ini berfungsi untuk mencatat izin atau sakit pada data presensi. Pengguna dapat mengakses halaman ini untuk memperbarui status kehadiran mereka dengan alasan izin atau sakit. Dengan adanya halaman ini, proses pengelolaan data presensi menjadi lebih fleksibel dan mudah dipantau. Seperti yang terlihat pada gambar 5.3.



Gambar 5.3 Implementasi Halaman Izin/Sakit

4. Implementasi Halaman History Presensi

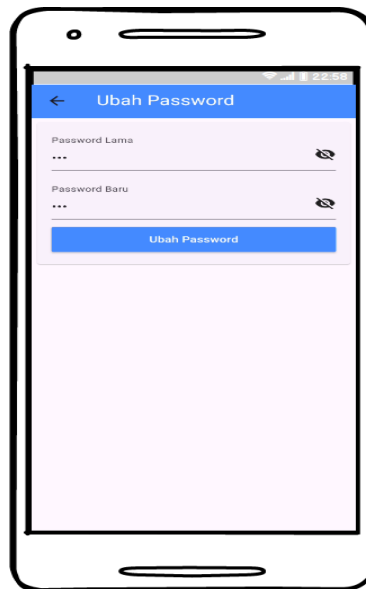
Merupakan halaman yang dapat digunakan untuk melihat data history presensi pegawai berdasarkan periode perbulan. Seperti yang terlihat pada gambar 5.4.



Gambar 5.4 Implementasi Halaman Data Rekap Presensi

5. Implementasi Halaman Ubah *Password*

Merupakan halaman yang dapat digunakan untuk melakukan perubahan *password*. Seperti yang terlihat pada gambar 5.5.



Gambar 5.5 Implementasi Halaman Ubah *Password*

6. Implementasi Halaman Scan Qr Code Presensi

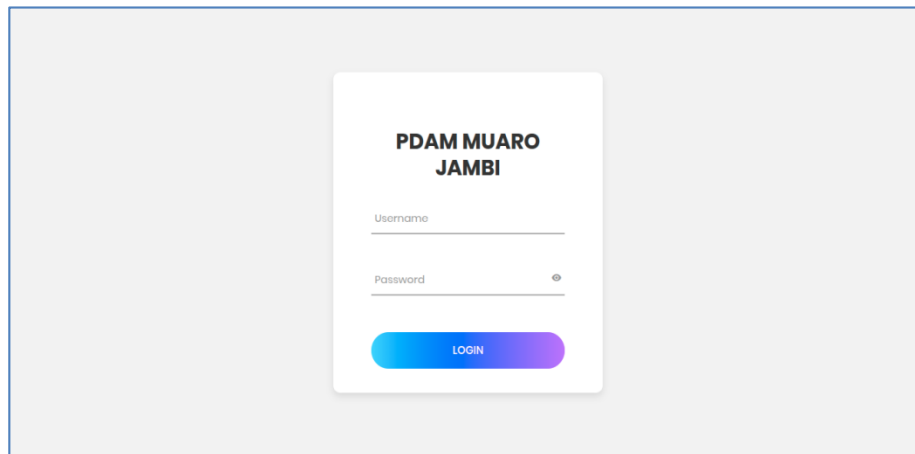
Berikut ini merupakan implementasi halaman scan Qr Code Presensi. Seperti terlihat pada gambar 5.6 :



Gambar 5.6 Implementasi Halaman Qr Code Presensi

7. Implementasi Halaman *Login Admin*

Berikut adalah implementasi dari halaman *login* admin yang dapat digunakan admin untuk masuk kedalam sistem admin yang dapat digunakan untuk mengelola menu menu yang ada pada sistem. Seperti terlihat pada gambar 5.7.



Gambar 5.7 Implementasi Halaman *Login Admin*

8. Implementasi Halaman Form Data Pegawai

Implementasi halaman form data pegawai merupakan halaman yang dapat diakses oleh admin untuk menambah data pegawai. Seperti terlihat pada gambar 5.8.

Gambar 5.8 Implementasi Halaman Form Data Pegawai

9. Implementasi Halaman Form Data Pengguna

Implementasi halaman form data pengguna merupakan halaman yang dapat diakses oleh admin untuk menambah dan mengubah pengguna aplikasi. Seperti terlihat pada gambar 5.9.

The screenshot shows the 'Tambah Pengguna' form. It has a sidebar on the left with the 'TIRTA DHARMA' logo and navigation links: Dashboard, Presensi, Master Data, and Laporan. The main content area is titled 'Tambah Pengguna' and contains the following fields: 'Nama Pengguna' (text input), 'Username' (text input), 'Level' (dropdown menu with 'Pilih...' selected), 'Password' (text input), and a 'Simpan' button. The top of the page shows 'PDAM MUARO JAMBI' and 'Administrator' with a dropdown arrow. A 'Kembali' button is in the top right corner of the form area. The footer says 'Copyright © PDAM MUARO JAMBI'.

Gambar 5.9 Implementasi Halaman Form Data Pengguna

10. Implementasi Halaman Form Presensi

Implementasi halaman form presensi merupakan halaman yang dapat diakses oleh admin menambah data presensi melalui aplikasi berbasis website. Seperti terlihat pada gambar 5.10.

The screenshot shows the 'Presensi' form. It has the same sidebar as the previous form. The main content area is titled 'Presensi' and contains the following fields: 'Tanggal' (text input with '01/20/2025'), 'Status Presensi' (dropdown menu with 'Pilih...' selected), 'Presensi (Masuk / Pulang)' (dropdown menu with 'Pilih...' selected), 'Pegawai' (dropdown menu with 'Pilih...' selected), and 'Foto (Surat Keterangan Sakit/ Izin)' (file upload area with a 'Choose File' button and 'No file chosen' text). There is a 'Simpan' button at the bottom. The top of the page shows 'PDAM MUARO JAMBI' and 'Administrator' with a dropdown arrow. A 'Kembali' button is in the top right corner of the form area. The footer says 'Copyright © PDAM MUARO JAMBI'.

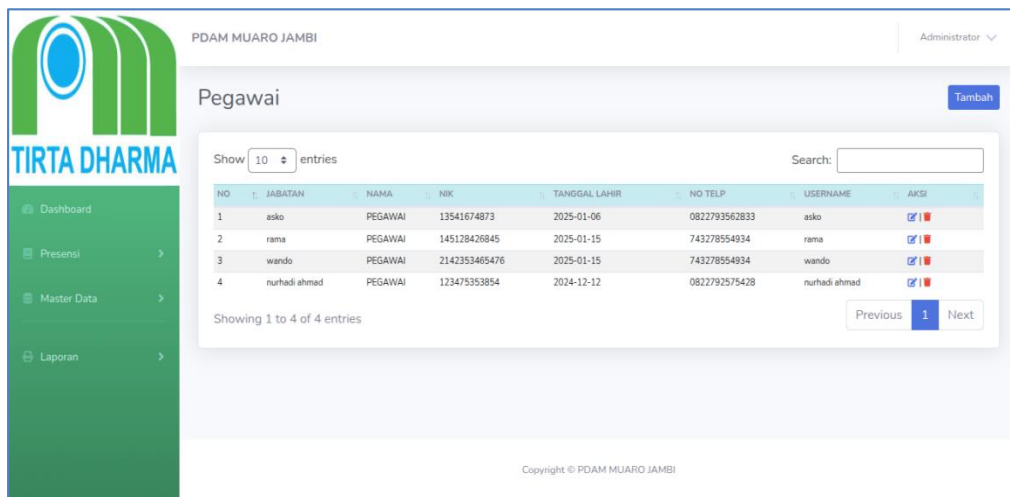
Gambar 5.10 Implementasi Halaman Form Presensi

5.1.2 Implementasi *Output*

Implementasi tampilan *output* dari sistem yang akan dibangun adalah sebagai berikut:

1. Implementasi Halaman Mengelola Data Pegawai

Implementasi mengelola data Pegawai merupakan halaman yang dapat diakses oleh admin untuk menambah, mengubah dan menghapus data Pegawai. Seperti terlihat pada gambar 5.11 :

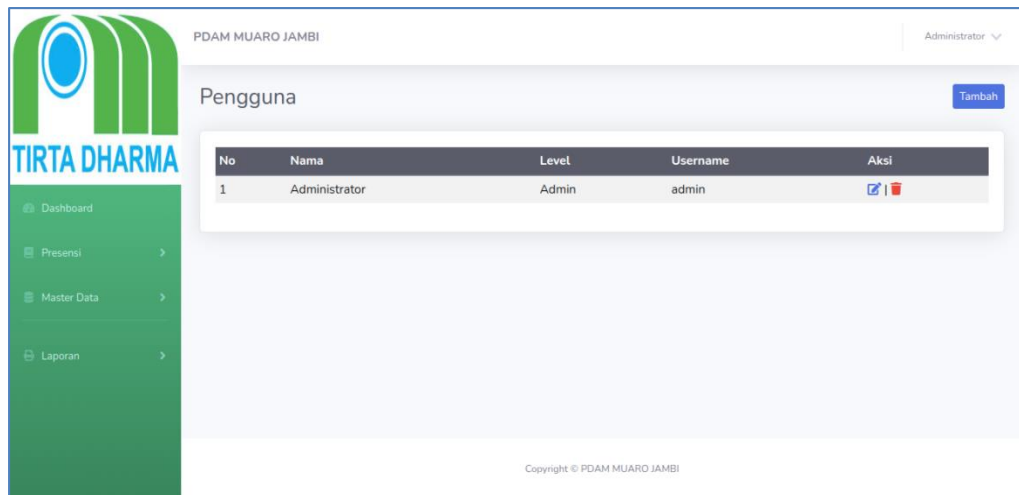


NO	JABATAN	NAMA	NIK	TANGGAL LAHIR	NO TELP	USERNAME	AKSI
1	asko	PEGAWAI	13541674873	2025-01-06	0822793562833	asko	Edit Delete
2	rama	PEGAWAI	145128426845	2025-01-15	743278554934	rama	Edit Delete
3	wando	PEGAWAI	2142353465476	2025-01-15	743278554934	wando	Edit Delete
4	nurhadi ahmad	PEGAWAI	123475353854	2024-12-12	0822792575428	nurhadi ahmad	Edit Delete

Gambar 5.11 Implementasi Halaman Mengelola Pegawai

2. Implementasi Halaman Mengelola Data Pengguna

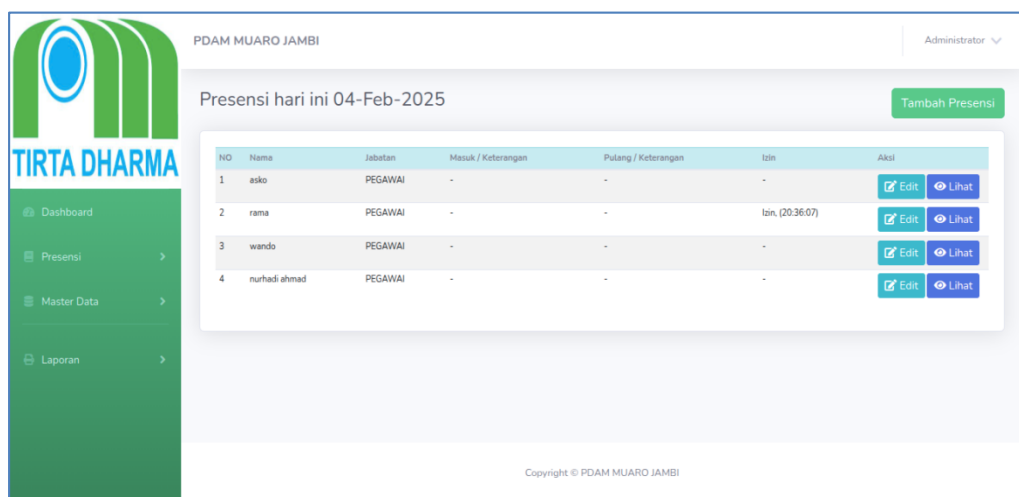
Implementasi mengelola data pengguna merupakan halaman yang dapat diakses oleh admin untuk menambah, mengubah dan menghapus data pengguna. Seperti terlihat pada gambar 5.12 :



Gambar 5.12 Implementasi Halaman Data Pengguna

3. Implementasi Halaman Presensi

Implementasi mengelola data presensi merupakan halaman yang dapat diakses oleh admin untuk melihat data presensi. Seperti terlihat pada gambar 5.13 :



Gambar 5.14 Implementasi Halaman Data Presensi

4. Implementasi Laporan Pegawai

Implementasi laporan pegawai ini berfungsi ketika admin ingin melihat data laporan semua pegawai. Tampilan halaman laporan pegawai dapat dilihat pada gambar 5.15.



PDAM MUARO JAMBI

Mendalo Darat, Kec. Jambi Luar Kota, Kabupaten Muaro Jambi, Jambi 36361

Pegawai

NO	JABATAN	NAMA	NIK	TANGGAL LAHIR	NO TELP
1	asko	PEGAWAI	1505050708110001	2025-01-06	0822793562833
2	rama	PEGAWAI	1505054310760003	2025-01-15	0895958235783
3	wando	PEGAWAI	1505054301090001	2025-01-15	0853485934584
4	nurhadi ahmad	PEGAWAI	1505054111020001	2024-12-12	0822792575428

PDAM MUARO JAMBI, 4 Februari 2025

Mengetahui

PIMPINAN

Gambar 5.15 Implementasi Halaman Laporan Pegawai

5. Implementasi Laporan Presensi

Implementasi laporan presensi pegawai ini berfungsi ketika admin ingin melihat data presensi pegawai periode bulanan. Tampilan halaman laporan presensi dapat dilihat pada gambar 5.16.



PDAM MUARO JAMBI

Mendalo Darat, Kec. Jambi Luar Kota, Kabupaten Muaro Jambi, Jambi 36361

Presensi Pegawai

Bulan : Februari - 2025

NAMA : rama

NIK : 145128426845

TANGGAL LAHIR : 2025-01-15

NO TELP : 743278554934

JABATAN : PEGAWAI

NO	Tanggal	Masuk / Keterangan	Pulang / Keterangan	Izin	Sakit
1	03-Feb-2025	Masuk(08:46:00) / Telat (76 Menit)	Pulang(12:44:00) / Mangkir (226 Menit)	-	-
2	04-Feb-2025	-	-	Izin (20:36:07)	-
3	05-Feb-2025	Masuk(20:36:23) / Telat (796 Menit)	Pulang(20:36:49) / Lembur (247 Menit)	-	-

Hadir : 2

Sakit : 0

Izin : 1

PDAM MUARO JAMBI, 4 Februari 2025

Mengetahui

PIMPINAN

Gambar 5.16 Implementasi Halaman Laporan Presensi

5.2 PENGUJIAN SISTEM / PERANGKAT LUNAK

Pengujian sistem digunakan untuk memastikan bahwa perangkat lunak yang telah dibuat telah selesai desainnya dan semua fungsi dapat dipergunakan dengan baik tanpa ada kesalahan. Jika, kesimpulan tidak sesuai maka penulis akan memperbaiki kembali program tersebut, akan tetapi jika telah sesuai maka penulis dapat melakukan pengujian ke bagian lainnya. Uji coba dilakukan pada komputer milik penulis, yang telah disebutkan speksifikasinya pada bab sebelumnya.

1. Pengujian Modul Login Admin

Pada tahap ini dilakukan pengujian pada modul *login* admin untuk mengetahui apakah proses *login* admin atau fungsional *login* admin dapat berjalan dengan baik. Hasil pengujian pada modul ini penulis sajikan sebagai berikut :

Tabel 5.1 Pengujian Modul *Login* Admin

Modul yang diuji	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
<i>Login</i> Admin (berhasil)	- Masukkan <i>username</i> dan <i>password</i> admin secara benar	<i>Username</i> dan <i>password</i> yang sesuai dengan <i>database</i>	Admin berhasil login dan tampil halaman utama	Admin berhasil login dan tampil halaman utama	Baik
<i>Login</i> Admin (gagal)	- Masukkan <i>username</i> dan <i>password</i> admin yang salah dan belum terdaftar dalam <i>database</i>	<i>Username</i> dan <i>Password</i> yang tidak sesuai dengan <i>database</i>	Admin gagal login dan tampil pesan “Username /password tidak sesuai”	Admin gagal login dan tampil pesan “Username /password tidak sesuai”	Baik

2. Pengujian Modul Mengelola Data Admin

Pada tahap ini dilakukan pengujian pada modul data admin oleh admin untuk mengetahui apakah proses mengelola data admin atau fungsional mengelola data admin dapat berjalan dengan baik. Hasil pengujian pada modul ini penulis sajikan pada tabel 5.2.

Tabel 5.2 Pengujian Modul Mengelola Data Admin

Modul yang diuji	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
Tambah Data Admin (berhasil)	- Pilih menu data admin - Klik Tambah - Input Data - Klik Simpan	Input data lengkap	Data berhasil ditambah dan tampilkan pesan “ Berhasil simpan admin”	Data berhasil ditambah dan tampilkan pesan “ Data telah tersimpan”	Baik
Tambah Data Admin (gagal)	- Pilih menu data admin - Klik Tambah - Klik Simpan	Input data tidak lengkap	Data gagal ditambah dan Tampilkan pesan “ Data gagal disimpan”	Data gagal ditambah dan Tampilkan pesan “ Data gagal disimpan”	Baik
Edit Data Admin (berhasil)	- Pilih menu data admin - Pilih record - Klik Edit - Input Data - Klik Ubah	Input data lengkap	Data berhasil diubah dan Tampilkan pesan “Berhasil ubah admin”	Data berhasil diubah dan Tampilkan pesan “Berhasil ubah admin”	Baik
Edit Data Admin (gagal)	- Pilih menu data admin - Pilih record - Klik Edit - Kosongkan field - Klik Ubah	Input data tidak lengkap	Data gagal diubah dan Tampilkan pesan “ Data gagal diubah”	Data gagal diubah dan Tampilkan pesan “ Data gagal diubah”	Baik

Modul yang diuji	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
Hapus Data Admin (berhasil)	- Pilih record data admin yang akan dihapus - Klik hapus	-	Data berhasil dihapus dan tampilkan Pesan Bahwa “Berhasil hapus admin”	Data berhasil dihapus dan tampilkan Pesan Bahwa “Berhasil hapus admin”	Baik

3. Pengujian Modul Mengelola Data Pegawai

Pada tahap ini dilakukan pengujian pada modul data Pegawai oleh admin untuk mengetahui apakah proses mengelola data Pegawai atau fungsional mengelola data Pegawai dapat berjalan dengan baik. Hasil pengujian pada modul ini penulis sajikan pada tabel 5.3.

Tabel 5.3 Pengujian Modul Mengelola Data Pegawai

Modul yang diuji	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
Tambah Data Pegawai (berhasil)	- Pilih menu data Pegawai - Klik Tambah - Input Data - Klik Simpan	Input data lengkap	Data berhasil ditambah dan tampilkan pesan “ Berhasil simpan Pegawai”	Data berhasil ditambah dan tampilkan pesan “ Data telah tersimpan”	Baik
Tambah Data Pegawai (gagal)	- Pilih menu data Pegawai - Klik Tambah - Klik Simpan	Input data tidak lengkap	Data gagal ditambah dan Tampilkan pesan “ Data gagal disimpan”	Data gagal ditambah dan Tampilkan pesan “ Data gagal disimpan”	Baik

Modul yang diuji	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
Edit Data Pegawai (berhasil)	- Pilih menu data Pegawai - Pilih record - Klik Edit - Input Data - Klik Ubah	Input data lengkap	Data berhasil diubah dan Tampilkan pesan “Berhasil ubah Pegawai”	Data berhasil diubah dan Tampilkan pesan “Berhasil ubah Pegawai”	Baik
Edit Data Pegawai (gagal)	- Pilih menu data Pegawai - Pilih record - Klik Edit - Kosongkan field - Klik Ubah	Input data tidak lengkap	Data gagal diubah dan Tampilkan pesan “ Data gagal diubah”	Data gagal diubah dan Tampilkan pesan “ Data gagal diubah”	Baik
Hapus Data Pegawai (berhasil)	- Pilih record data Pegawai yang akan dihapus - Klik hapus	-	Data berhasil dihapus dan tampilkan Pesan Bahwa “Berhasil hapus Pegawai”	Data berhasil dihapus dan tampilkan Pesan Bahwa “Berhasil hapus Pegawai”	Baik

4. Pengujian Modul Mengelola Data Presensi

Pada tahap ini dilakukan pengujian pada modul data presensi masuk pada aplikasi pegawai untuk mengetahui apakah proses presensi masuk atau fungsional mengelola data presensi masuk dapat berjalan dengan baik. Hasil pengujian pada modul ini penulis sajikan pada tabel 5.5.

Tabel 5.4 Pengujian Modul Mengelola Data Presensi

Modul yang diuji	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapat	Kesimpulan
Tambah Data Presensi (berhasil)	- Pilih menu data Presensi masuk - Arahkan ke kode QR Code Presensi	Input data lengkap	Data berhasil ditambah dan tampilkan pesan “Anda sudah melakukan presensi”	Data berhasil ditambah dan tampilkan pesan “Anda sudah melakukan presensi”	Baik
Tambah Data Presensi (gagal)	- Pilih menu data Presensi masuk - Scan sembarang QR code	Inputkan sembarang Qr Code	Data gagal ditambah dan Tampilkan pesan “Gagal melakukan presensi”	Data gagal ditambah dan Tampilkan pesan “Gagal melakukan presensi”	Baik

5.3 ANALISIS HASIL YANG DICAPAI OLEH SISTEM

Setelah melakukan berbagai pengujian pada sistem pengolahan data presensi di Kantor PDAM Tirta Muaro Jambi, dapat disimpulkan bahwa sistem ini berjalan dengan baik dalam mencatat dan memproses data kehadiran. Pengujian yang dilakukan menunjukkan bahwa sistem mampu mencatat kehadiran karyawan secara akurat, termasuk dalam fitur izin dan sakit. Selain itu, respons sistem terhadap input data cukup cepat, dan antarmuka pengguna mudah dipahami, memudahkan pengelolaan data presensi. Namun, beberapa perbaikan masih diperlukan dalam hal pemeliharaan sistem untuk menghindari potensi kesalahan pengolahan data di masa depan. Secara keseluruhan, sistem ini memberikan dampak positif dalam meningkatkan efisiensi pencatatan dan pengelolaan data presensi di kantor. Adapun kelebihan dan kelemahan dari sistem ini adalah sebagai berikut:

5.3.1 Kelebihan Program

Setelah melakukan pengujian terhadap program yang dibuat dapat dijabarkan mengenai kelebihan program yang dapat dilihat sebagai berikut :

1. Sistem ini dapat diakses oleh pengguna dalam Sistem Informasi Absensi Pegawai yang memanfaatkan Teknologi QR Code (*Quick Response Code*) di Kantor PDAM Tirta Muaro Jambi, dengan adanya fitur pencarian yang memudahkan proses pencarian data.
2. Sistem ini dapat mengelola berbagai data, seperti data Admin, Absensi, dan Pegawai. Sistem ini efektif dalam mengolah data Admin, Absensi, serta data Pegawai secara keseluruhan.
3. Aplikasi mobile digunakan untuk memasukkan data presensi, yang mempercepat proses administrasi baik pada saat kedatangan maupun kepulangan.

5.3.2 Kekurangan Program

Setelah melakukan pengujian terhadap program yang dibuat dapat dijabarkan mengenai kelebihan program yang dapat dilihat sebagai berikut :

1. Tampilan antarmuka sistem saat ini masih tergolong standar, dengan desain yang sederhana dan fungsional. Meskipun dapat digunakan dengan mudah, tampilan ini belum memiliki elemen desain yang menarik atau inovatif.
2. Aplikasi hanya dapat dijalankan sistem berbasis Android belum tersedia untuk IOS.

BAB VI

PENUTUP

6.1 KESIMPULAN

Berdasarkan hasil analisa dan pembahasan yang telah dilakukan pada bab-bab sebelumnya, maka penulis memberikan kesimpulan tentang perancangan aplikasi absensi PDAM Muaro Jambi sebagai berikut:

1. PDAM Muaro Jambi dalam Pengelolaan presensi secara konvensional dengan mencatat pada lembar absensi, akan membutuhkan waktu yang lama karena harus satu per satu data dimasukkan. Rentan juga adanya manipulasi data atau terjadi data yang ganda, sehingga tidak mendapatkan hasil yang akurat.
2. Dengan adanya aplikasi ini pegawai PDAM Muaro Jambi tidak perlu lagi mengisi absensi secara manual karena semua data absensi tercatat secara otomatis dan terintegrasi.
3. Dapat mempermudah dalam melakukan presensi menggunakan mobile dan melihat presensi serta mempermudah dalam pembuatan laporan selama periode perbulan.

6.2 SARAN

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam aplikasi yang telah dibuat, oleh sebab itu ada beberapa saran yang dapat dilakukan untuk mengembangkan aplikasi absensi ini ke depannya:

1. Sebelum sistem baru diterapkan, diperlukan jaringan internet yang stabil dan pelatihan untuk pengguna, baik itu admin maupun operator, yang akan

mengoperasikan sistem tersebut. Hal ini bertujuan agar implementasi sistem dapat berjalan dengan lancar dan sesuai dengan yang diharapkan.

2. Diharapkan aplikasi ini dapat berjalan di berbagai platform, tidak terbatas hanya pada Android. Dengan kemampuan multi-platform, aplikasi akan lebih fleksibel dan dapat diakses oleh lebih banyak pengguna, termasuk yang menggunakan perangkat dengan sistem operasi berbeda.
3. Disarankan untuk menambahkan fitur notifikasi sebagai pemberitahuan ketika data telah berhasil diinput oleh pegawai.

DAFTAR PUSTAKA

- [1] Mulyanti, “Perancangan Aplikasi Presensi Pegawai Menggunakan Metode Geofencing Dan Perhitungan Jarak Pada Puskesmas Inuman,” *J. Perencanaan, Sains, Teknol. dan Komput.*, vol. 3, no. 2, pp. 565–575, 2020.
- [2] A. D. Laksono and R. Aulianita, “Sistem Informasi Presensi Karyawan Berbasis Android dengan Metode Agile Pada PRESTISA Bekasi,” *Inf. Manag. Educ. Prof.*, vol. 8, no. 2, pp. 101–110, 2023.
- [3] M. Niswah and J. Devitra, “Sistem Informasi Manajemen Presensi Kehadiran Karyawan Berbasis Web Dan SMS Gateway Pada SMK Dr. Indra Adnan Indragiri Collage Tembilahan,” *Manaj. Sist. Inf.*, vol. 8, no. 2, pp. 232–241, 2023.
- [4] I. H. Santi, *Analisa Perancangan Sistem*. Jawa Tengah: PT. Nasya Expanding Management, 2020.
- [5] Hendri, Intan Permata, and Fachruddin, “Perancangan Sistem Informasi Penjualan Dan Jasa Perbaikan Mobil Pada Bengkel Alam Jaya,” *J. Process.*, vol. 15, no. 1, pp. 30–42, 2020, doi: 10.33998/processor.2020.15.1.708.
- [6] H. H. Solihin and A. A. Fuja Nusa, “Rancang Bangun Sistem Informasi Penjualan, Pembelian Dan Persediaan Suku Cadang Pada Bengkel Tiga Putra Motor Garut,” *Infotronik J. Teknol. Inf. dan Elektron.*, vol. 2, no. 2, p. 107, 2017, doi: 10.32897/infotronik.2017.2.2.37.
- [7] H. A. S. Setyawan, Muhammad Yusril Munari, *Panduan Lengkap Membangun Sistem Monitoring Kinerja Mahasiswa Internship Berbasis Web Dan Global Positioning System Front Cover*. Bandung: Kreatif Industri Nusantara, 2020.
- [8] Harmayani, *Aplikasi Komputer*. Medan: Yayasan Kita Menulis, 2021.
- [9] E. Setiawan, “KBBI - Kamus Besar Bahasa Indonesia,” *kamus besar Bhs. Indones.*, 2019.
- [10] R. H. Gunawan, “Pembuatan Absensi Berbasis Android Menggunakan Metode Waterfall Untuk Program Studi Pendidikan Teknologi Informasi Ipi Garut,” *Gunahumas*, vol. 2, no. 1, pp. 318–328, 2020, doi: 10.17509/ghm.v2i1.23052.
- [11] A. Husain, A. H. A. Prastian, and A. Ramadhan, “Perancangan Sistem Absensi Online Menggunakan Android Guna Mempercepat Proses Kehadiran Karyawan Pada PT. Sintech Berkah Abadi,” *Technomedia J.*, vol. 2, no. 1, pp. 105–116, 2017, doi: 10.33050/tmj.v2i1.319.
- [12] M. A. Murtadho, N. A. Musthofa, and S. Mutrofin, “Implementasi Quick Response (Qr) Code Pada Aplikasi Validasi Dokumen Menggunakan

Perancangan Unified Modelling Language (Uml),” *Antivirus J. Ilm. Tek. Inform.*, vol. 10, no. 1, pp. 42–50, 2016, doi: 10.35457/antivirus.v10i1.87.

- [13] J. D. Irawan and E. Adriantantri, “Pemanfaatan QR-Code Sebagai Media Promosi Toko,” *J. Mnemon.*, vol. 1, no. 2, p. 57, 2018.
- [14] B. Wasito and H. Novian, “Pemanfaatan Quick Response Code Untuk Pencarian Informasi Produk,” *J. Inform. dan Bisnis*, vol. 9, no. 2, pp. 1–8, 2020.
- [15] Suhariato, L. B. A. Pambudi, A. Rahagiyanto, and G. E. J. Suyoso, “Implementasi QR Code untuk Efisiensi Waktu Pemesanan Menu Makanan dan Minuman di Restoran maupun Kafe,” *BIOS J. Teknol. Inf. dan Rekayasa Komput.*, vol. 1, no. 1, pp. 35–39, 2020, doi: 10.37148/bios.v1i1.7.
- [16] W. Nugraha, M. Syarif, and W. S. Dharmawan, “Penerapan Metode Sdlc Waterfall Dalam Sistem Informasi Inventori Barang Berbasis Desktop,” *JUSIM (Jurnal Sist. Inf. Musirawas)*, vol. 3, no. 1, pp. 22–28, 2018, doi: 10.32767/jusim.v3i1.246.
- [17] M. Alda, *Rekayasa Perangkat Lunak Berorientasi Objek*. Bandung: CV. MEDIA SAINS INDONESIA, 2021.
- [18] R. A. Sukamto and M. Shalahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung, 2019.
- [19] Munawar, *Analisis perancangan sistem berorientasi objek dengan UML(Unified Modeling Language)*. Bandung: Informatika Bandung, 2019.
- [20] R. F. Sari and A. U. S, *Rekayasa Perangkat Lunak Berorientasi Objek Menggunakan PHP*. Andi Offset, 2021.
- [21] M. Muslihudin and Oktafianto, *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*. Yogyakarta: Andi Offset, 2016.
- [22] F. Wahyutama, F. Samopa, and H. Suryotrisongko, “Penggunaan Teknologi Augmented Reality Berbasis Barcode sebagai Sarana Penyampaian Informasi Spesifikasi dan Harga Barang yang Interaktif Berbasis Android, Studi Kasus pada Toko Elektronik ABC Surabaya,” *J. Tek. ITS*, vol. 2, no. 3, pp. A481–A486, 2014.
- [23] A. Garg, “Android Versions List – From Version 1.0 to Android 15.” <https://trainings.internshala.com/blog/android-versions/> (accessed Jun. 16, 2024).
- [24] F. E. Krisnada and R. Tanone, “Aplikasi Penjualan Tiket Kelas Pelatihan Berbasis Mobile menggunakan Flutter,” *J. Tek. Inform. dan Sist. Inf.*, vol. 5, no. 3, pp. 281–295, 2020, doi: 10.28932/jutisi.v5i3.1865.
- [25] F. D. Anastasia and I. V Paputungan, “Implementasi BLOC Pattern Pada Pengembangan Frontend Top Up LinkAja untuk Aplikasi AGEN46,” *J. SNATI*, vol. 2, no. 1, pp. 2807–5935, 2022.

- [26] J. M. Suhendro, M. Sudarma, and C. D. Khrisne, "Rancang Bangun Aplikasi Seluler Penyedia Jasa Perawatan Dan Kecantikan Menggunakan Framework Flutter," *J. SPEKTRUM*, vol. 8, no. 2, p. 68, 2021, doi: 10.24843/spektrum.2021.v08.i02.p9.
- [27] N. Wulandari, A. Wibowo, and B. Susanto, "Penerapan RESTful API untuk Membangun Program Pembayaran Piutang Menggunakan Otentikasi OAuth 2.0," *J. Terap. Teknol. Inf.*, vol. 5, no. 1, pp. 1–10, 2021, doi: 10.21460/jutei.2021.51.230.
- [28] R. Choirudin and A. Adil, "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 284–293, 2019, doi: 10.30812/matrik.v18i2.407.
- [29] M. A. Novianto and S. Munir, "Analisis dan Implementasi Restful API guna Pengembangan Sistem Informasi Akademik pada Perguruan Tinggi," *J. Inform. Terpadu*, vol. 8, no. 1, pp. 47–61, 2022, doi: 10.54914/jit.v8i1.409.
- [30] A. Prayitno and Y. Safitri, "Pemanfaatan Sistem Informasi Perpustakaan Digital Berbasis Website Untuk Para Penulis," *IJSE-Indonesian J. Software Eng.*, vol. 756–759, no. 1, pp. 138–140, 2014, doi: 10.4028/www.scientific.net/AMR.756-759.138.
- [31] J. Enterprise, *HTML, PHP, dan MySQL untuk Pemula*. Jakarta: PT. Elex Media Komputindo, 2018.
- [32] P. Hidayatullah and J. K. Kawistara, *Pemrograman Web*. Bandung: Informatika Bandung, 2019.
- [33] B. Raharjo, *Belajar Otodidak MySQL Teknik Pembuatan dan Pengelolaan Database*. Bandung: Informatika Bandung, 2015.
- [34] Y. Yudhanto and H. A. Prasetyo, *Mudah menguasai framework laravel*. Jakarta: PT. Elex Media Komputindo, 2019.
- [35] F. Luthfi, "Penggunaan Framework Laravel dalam Rancang Bangun Modul Back-End Artikel Website Bisnisbisnis.ID," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 2, no. 1, pp. 34–41, 2017, doi: 10.14421/jiska.2017.21-05.
- [36] Y. A. Permana and P. Romadlon, "Perancangan Sistem Informasi Penjualan Perumahan Menggunakan Metode SDLC Pada PT. Mandiri Land Prosperous Berbasis Mobile," *J. Teknol. Pelita Bangsa*, vol. 10, no. 2, pp. 1511–1518, 2019, doi: 10.1134/s0320972519100129.
- [37] A. H. Hendri and Mochammad Arief Sutisna, "Sistem Informasi Pelaksanaan Kegiatan Komisi Kepolisian Nasional Berbasis Desktop," *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 2, no. 1, pp. 14–23, 2021, doi: 10.37859/coscitech.v2i1.2393.
- [38] N. A. Ramdhan and D. A. Nufriana, "Rancang Bangun Dan Implementasi

Sistem Informasi Skripsi Oline Berbasis WEB,” *J. Ilm. Intech Inf. Technol. J. UMUS*, vol. 1, no. 02, pp. 1–12, 2019, doi: 10.46772/intech.v1i02.75.

- [39] A. Setiawati, A. Rahim, and D. Kisbianty, “Pengembangan dan Pengujian Aspek Usability pada Sistem Informasi Perpustakaan (Studi Kasus : STIKOM Dinamika Bangsa Jambi),” *Processor*, vol. 13, no. 1, pp. 1173–1178, 2018.
- [40] B. Nugroho, M. S. Hasibuan, and M. H. Annabil, “Perancangan Aplikasi Absensi Pegawai Berbasis Web Dengan Blackbox Testing pada DISPORA Sumatera Utara,” *J. Comput. Sci. Informatics Eng.*, vol. 02, no. 4, pp. 199–209, 2023, doi: 10.55537/cosie.v2i4.714.
- [41] B. Uddin *et al.*, “Aplikasi Absensi Karyawan Berbasis Mobile GPS Pada PD. Kilau Laju Intan (PD. KLI),” *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 6, pp. 920–929, 2022.
- [42] S. Usman, J. Jeffry, and F. Aziz, “Pengembangan Absensi berbasis Mobile Aplikasi pada Badan Kepegawaian dan Pengembangan Sumber Daya Manusia Kabupaten Bone,” *J. Teknol. Terpadu*, vol. 7, no. 2, pp. 108–112, 2021, doi: 10.54914/jtt.v7i2.437.
- [43] M. Alda, D. Y. Laily, D. Purwani, and N. C. Harahap, “Perancangan Aplikasi Sistem Informasi Absensi Online Berbasis Mobile (Studi Kasus : Sma Negeri1 Dolok Masihul),” *J. Progr. Stud. PGMI*, vol. 10, pp. 800–809, 2023.
- [44] P. Apriadi and E. Sutrisna, “Perancangan Aplikasi Absensi Karyawan Berbasis Mobile Menggunakan GPS (Studi Kasus PT. Trans Retail Indonesia),” *J. Autom. Comput. Inf. Syst.*, vol. 3, no. 1, pp. 1–9, 2023, doi: 10.47134/jacis.v3i1.54.

LAMPIRAN A

DAFTAR RIWAYAT HIDUP

Nama : Rama Rifki Fadli
NIM : 8020200003
Jurusan : Teknik Informatika
Jenis Kelamin : Laki-Laki
Status : Belum Menikah
Agama : Islam
Tempat/Tanggal Lahir : Sipin teluk duren, 24 04 2002
Alamat : Jl. Jambi Suak Kandis Km 27, Rt 09
Kel/desa Sipin teluk duren Kumpeh ulu
No Telp : 0822-8139-9397

N0	Nama Sekolah	Tamatan
1	SD N 47 Sipin Teluk Duren	Tahun 2008-2014
2	SMP N 16 Sipin Teluk Duren	Tahun 2014-2017
3	SMA 5 Arang Arang	Tahun 2017-2020
4	Universitas Dinamika Bangsa Jambi	Tahun 2020-Sekarang

Jambi, 5 Februari 2025

Rama Rifki Fadli

LAMPIRAN B
KARTU BIMBINGAN

LAMPIRAN B
KARTU BIMBINGAN

LAMPIRAN C
SURAT KETERANGAN



PEMERINTAH KABUPATEN MUARO JAMBI
PERUSAHAAN UMUM DAERAH AIR MINUM
TIRTA MUARO JAMBI
UNIT PELAYANAN METRO MENDALO



Muaro Jambi, Januari 2025

Nomor : 001/SP/PDAM-MM/I/2025
Lamp : -
Hal : Selesai Penelitian

Dengan hormat,

Schubungan dengan surat no : 389/SP/DK-FK/UNAMA/II/2025

Nama : Rama Rifki Fadli
Nim : 8020200003
Prodi : Teknik Informatika

Bahwa nama tersebut telah selesai melakukan penelitian untuk menyelesaikan tugas akhir dengan judul "Perancangan Aplikasi Presensi Berbasis Mobile Pada PDAM Muaro Jambi".

Demikian Surat Keterangan ini dibuat untuk di pergunakan sebagaimana semestinya.

Muaro Jambi, Januari 2025

Diketahui Oleh
Kepala Unit Metro Mendalo


ABUNJANI, SE
NPP. 200 205 016

LAMPIRAN D

TRANSKRIP WAWANCARA

Tempat Wawancara : PDAM MUARO JAMBI
Tanggal Wawancara : 10 Januari 2024
Waktu Wawancara : 10.00 Wib - Selesai
Narasumber : Asko
Jabatan : Pimpinan PDAM

No	Penulis(P) Narasumber(N)	
1	P	Jika boleh bertanya, bagaimana proses presensi pegawai dilakukan di perusahaan ini saat ini?
	N	Saat ini, proses presensi masih dilakukan secara manual. Pegawai mencatat kehadiran di buku presensi, dengan mencatat pada lembar absensi, akan membutuhkan waktu yang lama karena harus satu per satu data dimasukkan. Rentan juga adanya manipulasi data atau terjadi data yang ganda, sehingga tidak mendapatkan hasil yang akurat.
2	P	Baik, Pak. Berdasarkan penjelasan tersebut, apakah bapak tertarik untuk menggunakan sebuah sistem berbasis teknologi untuk mempermudah proses tersebut?
	N	Tentu saja, kami sangat tertarik menggunakan sistem yang lebih modern untuk meningkatkan efisiensi dan akurasi.
3	P	Sistem seperti apa yang bapak harapkan dari segi fungsionalitas?
	N	Saya berharap sistem tersebut dapat mencatat presensi pegawai secara otomatis, baik melalui perangkat digital atau aplikasi, Selain itu, sistem tersebut sebaiknya mudah digunakan oleh pegawai maupun admin.
4	P	Terima kasih, Pak, atas informasinya. Berdasarkan kebutuhan tersebut, saya ingin membantu bapak dengan mengembangkan sebuah aplikasi yang dirancang khusus untuk perusahaan ini. Tujuannya adalah untuk memberikan kemudahan dalam proses presensi.
	N	Ide ini sangat bagus, dan saya mendukung penuh upaya kamu untuk mengembangkan solusi ini.

Muaro Jambi, 10 Januari 2025

Asko

LAMPIRAN E

FOTO DOKUMENTASI DI KANTOR PDAM MUARO JAMBI



LAMPIRAN F

LISTING KODE PROGRAM

Flutter

1. Main.dart

```
import 'package:apppresensi/aplikasi/aplikasi.dart';
import 'package:apppresensi/auth/login.dart';
import 'package:apppresensi/util/session.dart';
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const SwitcherWidget()
    );
  }
}

class SwitcherWidget extends StatelessWidget {
  const SwitcherWidget({super.key});

  @override
  Widget build(BuildContext context) {
    return FutureBuilder(
      future: loadSession(),
      builder: (context, snapshot) {

        if(snapshot.connectionState == ConnectionState.waiting) {
```

```

        return const Scaffold(
          body: Center(
            child: CircularProgressIndicator(),
          ),
        );
      } else if(snapshot.connectionState == ConnectionState.done) {
        if(snapshot.hasData) {
          final SharedPreferences prefs = snapshot.data;
          if(prefs.getBool(IS_LOGIN) ?? false) {
            return const AplikasiWigdet();
          } else {
            return const LoginWidget();
          }
        }
      }
    }
  }
  return Container(child: const Text("Error"));
}
);
}
}

```

2. **Login.dart**

```

import 'dart:convert';

import 'package:apppresensi/aplikasi/aplikasi.dart';
import 'package:apppresensi/auth/authapi.dart';
import 'package:apppresensi/auth/requestlogin.dart';
import 'package:apppresensi/auth/responselogin.dart';
import 'package:apppresensi/util/message.dart';
import 'package:apppresensi/util/session.dart';
import 'package:flutter/material.dart';

class LoginWidget extends StatefulWidget {
  const LoginWidget({super.key});

  @override
  State<LoginWidget> createState() => _LoginWidgetState();
}

class _LoginWidgetState extends State<LoginWidget> {
  final TextEditingController username = TextEditingController();
  final TextEditingController password = TextEditingController();
  bool isVisible = true;

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.white,
    body: Padding(
      padding: const EdgeInsets.all(8.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Image.asset(
            "assets/images/logo.jpg",
            height: 140,
            scale: 1.5,
          ),
          const SizedBox(
            height: 16.0,
          ),
          const Text("Aplikasi Presensi", style: TextStyle(color: Colors.black54,
fontSize: 23)),
          const SizedBox(
            height: 16.0,
          ),
          TextFormField(
            controller: username,
            decoration: const InputDecoration(
              label: Text("Username"),
              hintText: "Username",
              border: OutlineInputBorder()
            ),
          ),
          const SizedBox(
            height: 16.0,
          ),
          TextFormField(
            controller: password,
            decoration: InputDecoration(
              suffix: GestureDetector(
                onTap: () {
                  setState(() {
                    isVisible = !isVisible;
                  });
                },
              ),
              child: Icon(isVisible ? Icons.visibility_off : Icons.visibility),
            ),
          ),

```

```

        label: const Text("Password"),
        hintText: "Password",
        border: const OutlineInputBorder()
    ),
    obscureText: isVisible,

),
const SizedBox(
    height: 16.0,
),
MaterialButton(
    onPressed: () {
        proseslogin(context, username.text, password.text);
    },
    minWidth: double.infinity,
    height: 40.0,
    textColor: Colors.white,
    color: Colors.blueAccent,
    child: const Text("Login"),
),
const SizedBox(
    height: 5.0,
),

],
),
),
);
}

```

```

void proseslogin(context, String vusername, String vpassword) async {
    try {
        final res = await AuthApi.apilogin(Requestlogin(username: vusername,
password: vpassword));
        // print(res.statusCode);
        if(res.statusCode == 200) {
            var responselogin =
Responselogin.fromJson(json.decode(json.encode(res.data)));
            if(responselogin.status == "berhasil") {
                createSession(responselogin);
                Navigator.pushReplacement(
                    context,
                    MaterialPageRoute(builder: (ctx) => const AplikasiWidget())
                );
                MessageWidget.snackbar(context, responselogin.pesan);
            } else {

```

```

        MessageWidget.snackbarperingatan(context, responselogin.pesan);
    }
    } else if(res.statusCode == 401) {
        MessageWidget.snackbarperingatan(context, "Maaf, anda gagal Login");
    }
    } on Exception catch (e) {
        MessageWidget.snackbarperingatan(context, "Maaf, anda gagal login");
    } catch (e) {
        print(e);
    }
    }
}

```

3. Aplikasi.dart

```

import 'package:apppresensi/aplikasi/akun/akun.dart';
import 'package:apppresensi/aplikasi/history/history.dart';
import 'package:apppresensi/aplikasi/izin/izin.dart';
import 'package:apppresensi/aplikasi/scanqr/scanqr.dart';
import 'package:apppresensi/auth/login.dart';
import 'package:apppresensi/util/session.dart';
import 'package:flutter/material.dart';

class AplikasiWiget extends StatefulWidget {
  const AplikasiWiget({super.key});

  @override
  State<AplikasiWiget> createState() => _AplikasiWigetState();
}

class _AplikasiWigetState extends State<AplikasiWiget> {
  int indexaktif = 0;
  List<Widget> daftarwidget = const [
    ScanqrWidget(),
    IzinWidget(),
    HistoryWidget(),
    AkunWidget()
  ];

  List<String> namamenu = const [
    "Scan QR Presensi", "Izin/Sakit", "History", "Akun"
  ];

  List<Icon> daftarikon = const [
    Icon(Icons.qr_code,color: Colors.white),

```

```

        Icon(Icons.file_copy_sharp,color: Colors.white),
        Icon(Icons.history,color: Colors.white),
        Icon(Icons.person,color: Colors.white)
    ];

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                leading: daftarikon[indexaktif],
                backgroundColor: Colors.blueAccent,
                title: Text(namamenu[indexaktif], style: const TextStyle(color:
Colors.white)),
                actions: [
                    IconButton(onPressed: () {
                        logout(context);
                    }, icon: const Icon(Icons.logout), color: Colors.white,)
                ],
            ),
            body: daftarwidget[indexaktif],
            bottomNavigationBar: BottomNavigationBar(
                currentIndex: indexaktif,
                items: [
                    BottomNavigationBarItem(icon: Icon(Icons.qr_code,color:
Colors.blue[400]), label: "Scan QR Presensi"),
                    BottomNavigationBarItem(icon: Icon(Icons.file_copy_sharp,color:
Colors.blue[400]), label: "Izin/Sakit"),
                    BottomNavigationBarItem(icon: Icon(Icons.history,color:
Colors.blue[400]), label: "History"),
                    BottomNavigationBarItem(icon: Icon(Icons.person,color:
Colors.blue[400]), label: "Akun")
                ],
                selectedItemColor: Colors.blue[200],
                onTap: (value) {
                    setState(() {
                        indexaktif = value;
                    });
                },
            ),
        );
    }

    void logout(BuildContext context) {
        clearSession();
        Navigator.pushReplacement(
            context, MaterialPageRoute(builder: (ctx) => const LoginWidget()));
    }

```

```
}  
}
```

4. Scanqrcode.dart

```
import 'package:apppresensi/aplikasi/apidio.dart';  
import 'package:flutter/material.dart';  
import 'package:mobile_scanner/mobile_scanner.dart';  
  
class ScanqrWidget extends StatefulWidget {  
  const ScanqrWidget({super.key});  
  
  @override  
  State<ScanqrWidget> createState() => _ScanqrWidgetState();  
}  
  
class _ScanqrWidgetState extends State<ScanqrWidget> {  
  Barcode? _barcode;  
  late Future<String> futureteks;  
  
  @override  
  void initState() {  
    super.initState();  
    setState(() {  
      futureteks = Apidio().postPresensiHadir("");  
    });  
  }  
  
  void _handleBarcode(BarcodeCapture barcodes) {  
    if (mounted) {  
      setState(() {  
        _barcode = barcodes.barcodes.firstOrNull;  
        futureteks = Apidio().postPresensiHadir(_barcode!.displayValue!);  
      });  
    }  
  }  
}  
  
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    backgroundColor: Colors.black,  
    body: Stack(  
      children: [  
        MobileScanner(  
          onDetect: _handleBarcode,  
        ),  
      ],  
    ),  
  );  
}
```

```

Align(
  alignment: Alignment.bottomCenter,
  child: Container(
    alignment: Alignment.bottomCenter,
    height: 100,
    color: Colors.black.withOpacity(0.4),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
        Expanded(
          child: Center(
            child: FutureBuilder(
              future: futureteks,
              builder: (context, snapshot) {
                if (snapshot.hasData) {
                  return Text(
                    snapshot.data!,
                    overflow: TextOverflow.fade,
                    style: const TextStyle(color: Colors.white),
                  );
                } else {
                  return const Text(
                    'Scan QR Code',
                    overflow: TextOverflow.fade,
                    style: TextStyle(color: Colors.white),
                  );
                }
              }
            )),
      ],
    ),
  ),
),
),
),
),
),
);
}
}

```

5. Izin.dart

```

import 'dart:convert';
import 'dart:io';

import 'package:apppresensi/aplikasi/apidio.dart';
import 'package:apppresensi/aplikasi/izin/responseizin.dart';
import 'package:apppresensi/util/message.dart';

```

```
import 'package:flutter/material.dart';
import 'package:flutter_image_compress/flutter_image_compress.dart';
import 'package:image_picker/image_picker.dart';
```

```
class IzinWidget extends StatefulWidget {
  const IzinWidget({super.key});

  @override
  State<IzinWidget> createState() => _IzinWidgetState();
}
```

```
class _IzinWidgetState extends State<IzinWidget> {
  final TextEditingController cirisepatu = TextEditingController();
```

```
List<String> jenisizin = ["Izin", "Sakit"];
String? _dataizin = "Izin";
File? _selectedImage;
String? _filebase64 = "";
```

```
@override
void initState() {
  super.initState();
  _loaddata();
}
```

```
void _loaddata() {
  setState(() {
    _filebase64 = "";
    _selectedImage = null;
  });
}
```

```
void postTambahIzin(context) async {
  final response = await Apidio().postPresensiIzin(
    _dataizin.toString(),
    _filebase64!
  );
  final resizin =
  Responseizin.fromJson(jsonDecode(jsonEncode(response.data)));
  if(resizin.status == "berhasil") {
    MessageWidget.snackbar(context, resizin.pesan);
    _loaddata();
  }
}
```

```
@override
```

```

Widget build(BuildContext context) {
  return Scaffold(
    body: Padding(
      padding: const EdgeInsets.all(2.0),
      child: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.start,
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              DropdownButtonFormField(
                decoration: const InputDecoration(
                  border: OutlineInputBorder(),
                  label: Text("Pilih Jenis Izin/Sakit"),
                  hintText: "Pilih Jenis Izin/Sakit"),
                items: jenisizin
                  .map((label) =>
                    DropdownMenuItem(value: label, child: Text(label)))
                  .toList(),
                onChanged: (value) {
                  _dataizin = value!;
                },
              ),
              const SizedBox(
                height: 8.0,
              ),
              Column(
                mainAxisAlignment: MainAxisAlignment.start,
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  const Text("Foto Surat Izin/Sakit"),
                  MaterialButton(
                    onPressed: () {
                      _pickImageCamera();
                    },
                    minWidth: double.infinity,
                    color: Colors.green,
                    textColor: Colors.white,
                    child: const Text("Foto Surat Izin/Sakit"))
                ],
              ),
              const SizedBox(
                height: 16.0,
              ),
              _selectedImage != null
            ],
          ),
        ),
      ),
    ),
  );
}

```

```

        ? Image.file(
            _selectedImage!,
            width: 400,
        )
        : const Text("Belum ada gambar"),
const SizedBox(
    height: 16,
),
MaterialButton(
    onPressed: () {
        postTambahIzin(context);
    },
    minWidth: double.infinity,
    height: 40.0,
    textColor: Colors.white,
    color: Colors.blueAccent,
    child: const Text("Simpan"),
),
],
),
))),
);
}

```

```

Future _pickImageCamera() async {
    final returnimage =
        await ImagePicker().pickImage(source: ImageSource.camera);
    var hasil = await
FlutterImageCompress.compressWithFile(returnimage!.path,
    quality: 70);

    List<int> imagebytes = hasil; //
File(returnimage!.path).readAsBytesSync();
    setState(() {
        _filebase64 = base64Encode(imagebytes);
        _selectedImage = File(returnimage.path);
    });
}
}

```

6. History.dart

```
import 'package:apppresensi/aplikasi/apidio.dart';
import 'package:apppresensi/aplikasi/model/Presensibulan.dart';
import 'package:apppresensi/util/listdata.dart';
import 'package:flutter/material.dart';

class HistoryWidget extends StatefulWidget {
  const HistoryWidget({super.key});

  @override
  State<HistoryWidget> createState() => _HistoryWidgetState();
}

class _HistoryWidgetState extends State<HistoryWidget> {
  late Future<List<Presensibulan>> futurepresensibulan;

  String bulan = "0";
  String tahun = "0000";

  @override
  void initState() {
    super.initState();
    _loaddata();
  }

  void _loaddata() {
    setState() {
      futurepresensibulan = Apidio().getHistoryBulan(bulan!, tahun!);
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Padding(
        padding: const EdgeInsets.all(2.0),
        child: Column(mainAxisAlignment: MainAxisAlignment.start, children: [
          Card(
            shape:
              RoundedRectangleBorder(borderRadius: BorderRadius.circular(5)),
            child: Padding(
              padding: const EdgeInsets.all(4.0),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.start,
                crossAxisAlignment: CrossAxisAlignment.start,
```

```

children: [
  DropdownButtonFormField(
    decoration: const InputDecoration(
      label: Text("Bulan"), hintText: "Bulan"),
    items: listbulan.entries
      .map((label) => DropdownMenuItem(
        value: label.key, child: Text(label.value)))
      .toList(),
    onChanged: (value) {
      setState(() {
        bulan = value!;
      });
    },
  ),
  const SizedBox(
    height: 8.0,
  ),
  DropdownButtonFormField(
    decoration: const InputDecoration(
      label: Text("Tahun"), hintText: "Tahun"),
    items: listtahun
      .map((label) => DropdownMenuItem(
        value: label, child: Text(label)))
      .toList(),
    onChanged: (value) {
      setState(() {
        tahun = value!;
      });
    },
  ),
  const SizedBox(
    height: 8.0,
  ),
  MaterialButton(
    onPressed: () {
      _loadaddata();
    },
    minWidth: double.infinity,
    height: 40.0,
    textColor: Colors.white,
    color: Theme.of(context).primaryColor,
    child: const Text("History Presensi"),
  )
],
),
)),
Expanded(
  flex: 1,

```

```

child: FutureBuilder(
  future: futurepresensibulan,
  builder: (ctx, snapshot) {
    if (snapshot.hasError) {
      return const Center(
        child: Text("Terjadi Error"),
      );
    } else if (snapshot.hasData) {
      var datapresensi = snapshot.data!;
      return ListView.builder(
        itemCount: datapresensi.length,
        itemBuilder: (context, index) {
          final presensi = datapresensi[index];
          return Card(
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(2)),
            child: Padding(
              padding: const EdgeInsets.all(8.0),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.start,
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Row(
                    mainAxisAlignment:
                      MainAxisAlignment.spaceBetween,
                    children: [
                      Text(presensi.tanggal,
                        style: const TextStyle(
                          color: Colors.blueGrey)),
                      Text(presensi.status,
                        style: const TextStyle(
                          color: Colors.greenAccent,      fontWeight:
FontWeight.bold)),
                    ],
                  ),
                  const SizedBox(
                    height: 5.0,
                  ),
                  Row(
                    crossAxisAlignment:
                      CrossAxisAlignment.center,
                    mainAxisAlignment:
                      MainAxisAlignment.spaceBetween,
                    children: [
                      Text("Jam Masuk : ${presensi.masuk ?? "-"}",
                        style: const TextStyle(

```

```

        color: Colors.indigoAccent)),
        Text("Jam Pulang : ${presensi.pulang ?? "-"}",
            style: const TextStyle(
                color: Colors.indigoAccent)),
    ],
  ),
  ],
),
));
});
} else {
  return const Center(
    child: CircularProgressIndicator(),
  );
}
}),
)
]),
));
}
}

```

7. Akun.dart

```

import 'package:apppresensi/aplikasi/akun/ubahindentitas.dart';
import 'package:apppresensi/aplikasi/akun/ubahpassword.dart';
import 'package:apppresensi/aplikasi/apidio.dart';
import 'package:apppresensi/aplikasi/model/pegawai.dart';
import 'package:flutter/material.dart';

class AkunWidget extends StatefulWidget {
  const AkunWidget({super.key});

  @override
  State<AkunWidget> createState() => _AkunWidgetState();
}

class _AkunWidgetState extends State<AkunWidget> {
  late Future<Pegawai>? futurepegawai;

  @override
  void initState() {
    super.initState();
    _loadakun();
  }
}

```

```

void _loadakun() async {
  setState(() {
    futurepegawai = Apidio().getPegawai();
  });
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: FutureBuilder(
      future: futurepegawai,
      builder: (context, snapshot) {
        if (snapshot.hasError) {
          return const Center(
            child: Text("Terjadi Error"),
          );
        } else if (snapshot.hasData) {
          var data = snapshot.data!;
          return Padding(
            padding: const EdgeInsets.all(3.0),
            child: Card(
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(2.0)),
              child: Padding(
                padding: const EdgeInsets.all(10.0),
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    // const Flexible(child: Text("Akun")),
                    buildUserInfoDisplay(data.nama, "Nama :"),
                    buildUserInfoDisplay(data.nik, "NIK :"),
                    buildUserInfoDisplay(data.jabatan, "Jabatan"),
                    buildUserInfoDisplay(
                      data.tgllahir, "Tanggal Lahir :"),
                    buildUserInfoDisplay(data.notelp, "No Telp :"),
                    buildUserInfoDisplay(data.email, "Email :"),
                    buildUserInfoDisplay(data.username, "Username :"),
                    const SizedBox(
                      height: 15,
                    ),
                    Row(
                      mainAxisAlignment: MainAxisAlignment.spaceBetween,
                      children: [
                        GestureDetector(
                          child: const Text(
                            "Ubah Identitas",

```

```

        style: TextStyle(color: Colors.green),
      ),
      onTap: () async {
        final bool? shouldRefresh =
          await Navigator.push<bool>(
            context,
            MaterialPageRoute(
              builder: (ctx) =>
                UbahidentitasWidget(
                  pegawai: data,
                )),
          );
        if (shouldRefresh != null) {
          if (shouldRefresh) {
            _loadakun();
          }
        }
      },
    ),
    GestureDetector(
      child: const Text(
        "Ubah Password",
        style: TextStyle(color: Colors.green),
      ),
      onTap: () async {
        Navigator.push<bool>(
          context,
          MaterialPageRoute(
            builder: (ctx) =>
              const UbahpasswordWidget()),
        ),
      ),
    ),
  ],
)
],
),
);
} else {
  return const Center(
    child: CircularProgressIndicator(),
  );
}
}));
}

```

```

Widget buildUserInfoDisplay(String getValue, String title) => Padding(
  padding: const EdgeInsets.only(bottom: 10),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        title,
        style: const TextStyle(
          fontSize: 15,
          fontWeight: FontWeight.w500,
          color: Colors.grey,
        ),
      ),
      const SizedBox(
        height: 1,
      ),
      Container(
        width: double.infinity,
        decoration: const BoxDecoration(
          border: Border(
            bottom: BorderSide(
              color: Colors.black12,
              width: 1,
            ),
          ),
        ),
        child: Padding(
          padding: const EdgeInsets.only(bottom: 5),
          child: Text(getValue,
            style: const TextStyle(fontSize: 16, height: 1.4)),
        ),
      ),
    ],
  ));
}

```

Website

1. Env

APP_NAME=Laravel

APP_ENV=local

APP_KEY=base64:7emg4eoN0Jw5YSrbr1V9pzLCy21u/R4WQyAN9x6Jy

RM=

APP_DEBUG=true

APP_URL=http://localhost

LOG_CHANNEL=stack

LOG_DEPRECATED_CHANNEL=null

LOG_LEVEL=debug

DB_DATABASE_SQLITE=fileapp/database/database.sqlite

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=ta

DB_USERNAME=root

DB_PASSWORD=

2. Route

<?php

```
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Admin\HomeController;
use App\Http\Controllers\Admin\JabatanController;
use App\Http\Controllers\Admin\KoordinatController;
use App\Http\Controllers\Admin\LaporanController;
use App\Http\Controllers\Admin\PegawaiController;
use App\Http\Controllers\Admin\PengaturanController;
use App\Http\Controllers\Admin\PenggunaController;
use App\Http\Controllers\Admin\PresensiController;
use App\Http\Controllers>LoginController;
use App\Http\Controllers\QrcodeController;
```

```
Route::get('/',[QrcodeController::class,'index']);
```

```
// auth admin
```

```
Route::get('/auth/login',[LoginController::class,'index'])->name('auth.login');
```

```
Route::post('/auth/login/proses',[LoginController::class,'proseslogin'])->name('auth.login.proses');
```

```
Route::get('/auth/logout',[LoginController::class,'proseslogout'])->name('auth.logout');
```

```
Route::prefix('adminweb')->middleware('auth:admin')->group(function(){
    Route::get('/home/admin',[HomeController::class,'index'])->name('home.admin');
```

```
Route::get('/home/ubahstatuspresensi',[HomeController::class,'ubahstatuspresensi'])->name('home.ubahstatus.presensi');
```

```
Route::get('/home/profil',[HomeController::class,'profil'])->name('home.profil');
```

```
Route::resource('/pegawai',PegawaiController::class);
```

```
Route::resource('/pengaturan',PengaturanController::class);
```

```
Route::resource('/jabatan',JabatanController::class);
```

```

Route::resource('/admin',PenggunaController::class);
Route::resource('/koordinat',KoordinatController::class);

Route::prefix('presensi')->name('presensi.')->group(function(){

    Route::get('/hariini',[PresensiController::class,'hariini'])-
>name('hariini');
    Route::get('/semuanya',[PresensiController::class,'semuanya'])-
>name('semuanya');

    Route::get('/perbulan',[PresensiController::class,'perbulan'])-
>name('perbulan');

Route::get('/perbulanloaddata',[PresensiController::class,'perbulanloaddata']
)->name('perbulan.loaddata');

    Route::get('/lihatpresensi',[PresensiController::class,'lihatpresensi'])-
>name('lihatpresensi');

    Route::get('/formpresensi',[PresensiController::class,'formpresensi'])-
>name('formpresensi');
    });

Route::prefix('laporan')->name('laporan.')->group(function(){
    Route::get('/pegawai',[LaporanController::class,'pegawai'])-
>name('pegawai');
    Route::get('/pegawaicetak',[LaporanController::class,'pegawaicetak'])-
>name('pegawai.cetak');

    Route::get('/presensi',[LaporanController::class,'presensi'])-
>name('presensi');

Route::get('/presensiloaddata',[LaporanController::class,'presensiloaddata'])-
>name('presensi.loaddata');

    Route::get('/perbulan',[LaporanController::class,'perbulan'])-
>name('perbulan');

Route::get('/perbulanloaddata',[LaporanController::class,'perbulanloaddata']
)->name('perbulan.loaddata');
    });

Route::get('/konfirmasihapus',[PresensiController::class,'konfirmasihapus'])-
>name('konfirmasihapus.view');

```

```
Route::get('/konfirmasihapusproses',[PresensiController::class,'konfirmasihapusproses']->name('konfirmasihapus.hapus'));
});
```

3. Pengguna

```
<?php
```

```
namespace App\Http\Controllers\Admin;
```

```
use App\Http\Controllers\Controller;
use App\Models\Admin;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Str;
```

```
class PenggunaController extends Controller
{
    private $title = 'Admin';
    private $namamenu = 'Pengguna';

    public function index()
    {
        $data = [
            'title' => $this->namamenu,
            'linkform' => route(Str::lower($this->title).'create'),
            'linkdestroy' => Str::lower($this->title).'destroy',
            'linkedit' => Str::lower($this->title).'edit',
            'tabel' => Admin::paginate(10)
        ];
        return view('admin.'.Str::lower($this->title).'tabel',$data);
    }

    public function create()
    {
        $data = [
            'title' => 'Tambah '. $this->namamenu,
            'linkback' => route(Str::lower($this->title).'index'),
            'route' => Str::lower($this->title).'store',
            'method' => 'POST',
            'data' => new Admin(),
            'level' => \Info::listlevel()
        ];
        return view('admin.'.Str::lower($this->title).'form',$data);
    }
}
```

```

public function store(Request $request)
{
    $request->request->add([
        'password' => Hash::make($request->passwordnya)
    ]);

    Admin::create($request->except('passwordnya'));
    return redirect()
        ->route(Str::lower($this->title).'index')
        ->with('success','Berhasil simpan '.$this->title);
}

public function show($id)
{
    //
}

public function edit($id)
{
    $data = [
        'title' => 'Ubah '. $this->namamenu,
        'linkback' => route(Str::lower($this->title).'index'),
        'route' => Str::lower($this->title).'update',
        'method' => 'PATCH',
        'data' => Admin::findOrFail($id),
        'level' => \Info::listlevel()
    ];
    return view('admin.'.Str::lower($this->title).'form',$data);
}

public function update(Request $request, $id)
{
    if($request->passwordnya != "") {
        $request->request->add([
            'password' => Hash::make($request->passwordnya)
        ]);
    }

    Admin::findOrFail($id)
        ->fill($request->except('passwordnya'))
        ->save();

    return redirect()
        ->route(Str::lower($this->title).'index')
        ->with('success','Berhasil ubah '. $this->namamenu);
}

```



```

<div class="col">
  <div class="card">
    <div class="card-body">
      {!! Form::model($data, ['route' => [$route, $data->id],
'method' => $method]) !!}
      <div class="row">
        <div class="col-md-6">
          <div class="form-group">
            <label>Nama Pengguna</label>
            {{ Form::text('nama', $data->nama, ['class' => 'form-
control', 'placeholder' => 'Nama Pengguna', 'autofocus', 'required']) }}
          </div>

          <div class="form-group">
            <label>Username</label>
            {{ Form::text('username', $data->username, ['class' =>
'form-control', 'placeholder' => 'Username', 'autofocus', 'required']) }}
          </div>
          <div class="form-group">
            <label>Level</label>
            {{ Form::select('level', $level, $data->level, [
'class' => 'form-control',
'placeholder' => 'Pilih...',
'autofocus',
'required',
]) }}
          </div>
          <div class="form-group">
            <label>Password</label>
            @if ($data->id == "")
              {{ Form::password('passwordnya', ['class' => 'form-
control', 'placeholder' => 'Password', 'autofocus', 'required']) }}
            @else
              {{ Form::password('passwordnya', ['class' => 'form-
control', 'placeholder' => 'Password']) }}
              <span>Kosongkan saja jika tidak diganti</span>
            @endif
          </div>

        </div>
      </div>

      <button type="submit" class="btn btn-primary btn-
sm">Simpan</button>
      {!! Form::close() !!}
    </div>
  </div>

```

```

        </div>
    </div>
</div>
</div>
@endsection

```

4. Pegawai

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\Jabatan;
use App\Models\Pegawai as model;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;

class PegawaiController extends Controller
{
    private $folder = 'admin.pegawai';
    private $title = 'Pegawai';
    private $route = 'pegawai';

    public function index(Request $request)
    {
        $data = [
            'title' => $this->title,
            'linkform' => route($this->route . '.create'),
            'linkdestroy' => $this->route . '.destroy',
            'linkedit' => $this->route . '.edit',
            'tabel' => model::all(),
        ];
        return view($this->folder . '.tabel', $data);
    }

    public function create()
    {
        $data = [
            'title' => 'Tambah ' . $this->title,
            'linkback' => route($this->route . '.index'),
            'route' => $this->route . '.store',
            'method' => 'POST',
            'data' => new model(),
            'jabatan' => Jabatan::pluck('jabatan','jabatan')

```

```

];
return view($this->folder . '.form', $data);
}

public function store(Request $request)
{
    $request->request->add([
        'password' => Hash::make($request->passwordnya)
    ]);

    model::create($request->all());
    return redirect()
        ->route($this->route . '.index')
        ->with('success', 'Berhasil simpan ' . $this->title);
}

public function show($id)
{
}

public function edit($id)
{
    $data = [
        'title' => 'Ubah ' . $this->title,
        'linkback' => route($this->route . '.index'),
        'route' => $this->route . '.update',
        'method' => 'PATCH',
        'data' => model::findOrFail($id),
        'jabatan' => Jabatan::pluck('jabatan','jabatan')
    ];
    return view($this->folder . '.form', $data);
}

public function update(Request $request, $id)
{
    $model = model::findOrFail($id);

    if ($request->password != "") {
        $request->request->add([
            'password' => Hash::make($request->password)
        ]);
    } else {
        $request->request->add([
            'password' => $model->password
        ]);
    }
}

```



```

        <th>TANGGAL LAHIR</th>
        <th>NO TELP</th>
        <th>USERNAME</th>
        <th width="80">AKSI</th>
    </tr>
</thead>
<tbody>
    @foreach ($tabel as $rs)
        <tr>
            <td>{{ $loop->iteration }}</td>
            <td>{{ $rs->nama }}</td>
            <td>{{ $rs->jabatan }}</td>
            <td>{{ $rs->nik }}</td>
            <td>{{ $rs->tgllahir }}</td>
            <td>{{ $rs->notelp }}</td>
            <td>{{ $rs->username }}</td>
            <td>
                <a href="{{ route($linkedit, $rs->id) }}"><i
class="fa fa-edit"></i></a> |
                <a data-toggle="modal" data-
target="#modalHapus" data-id="{{ $rs->id }}"
href="#"><i class="fa fa-trash text-
danger"></i></a>
            </td>
        </tr>
    @endforeach
</tbody>
</table>
</div>
</div>
</div>
</div>

    @include('admin.jshapusdata')
@endsection

@extends('admin.app')

@section('content')
    <div class="container-fluid">

        <div class="row">
            <div class="col-4">

```

```

        <h1 class="h3 mb-4 text-gray-800">{{ $title }}</h1>
    </div>
    <div class="col-8 text-right">
        <a href="{{ $linkback }}" class="btn btn-sm btn-outline-
secondary">Kembali</a>
    </div>
</div>
<div class="row">
    <div class="col">
        <div class="card">
            <div class="card-body">
                {!! Form::model($data, ['route' => [$route, $data->id],
'method' => $method]) !!}
                <div class="row">
                    <div class="col-md-6">
                        <div class="form-group">
                            <label>Nama</label>
                            {{ Form::text('nama', $data->nama, ['class' => 'form-
control', 'placeholder' => 'Nama', 'autofocus', 'required']) }}
                        </div>
                        <div class="form-group">
                            <label>NIK</label>
                            {{ Form::text('nik', $data->nip, ['class' => 'form-
control', 'placeholder' => 'NIK', 'autofocus', 'required']) }}
                        </div>
                        <div class="form-group">
                            <label>Tanggal Lahir</label>
                            {{ Form::date('tglahir', $data->tglahir, ['class' =>
'form-control', 'placeholder' => 'Tanggal Lahir', 'autofocus', 'required']) }}
                        </div>
                        <div class="form-group">
                            <label>Jabatan</label>
                            {{ Form::select('jabatan', $jabatan, $data->jabatan, [
'class' => 'form-control',
'placeholder' => 'Pilih...',
'autofocus',
'required',
]) }}
                        </div>
                    </div>
                </div>
                <div class="col-md-6">
                    <div class="form-group">
                        <label>No HP</label>
                        {{ Form::text('notelp', $data->notelp, ['class' => 'form-
control', 'placeholder' => 'No HP', 'autofocus', 'required']) }}
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
        <div class="form-group">
            <label>Username</label>
            {{ Form::text('username', $data->username, ['class' =>
'form-control', 'placeholder' => 'Username', 'autofocus', 'required']) }}
        </div>

        <div class="form-group">
            <label>Password</label>
            {{ Form::password('password', ['class' => 'form-
control', 'placeholder' => 'Password']) }}
            @if ($data->id != "")
                <span>Kosongkan saja jika tidak diganti</span>
            @endif
        </div>

    </div>
</div>

    <button type="submit" class="btn btn-primary btn-
sm">Simpan</button>
    {!! Form::close() !!}
</div>
</div>
</div>
</div>
</div>
@endsection

```

5. Presensi

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\Pegawai;
use App\Models\Presensi;
use Illuminate\Http\Request;
use Illuminate\Support\Carbon;

class PresensiController extends Controller
{
    private $folder = 'admin.presensi';
    private $title = 'Presensi';
    private $route = 'presensi';
}

```

```

public function hariini(Request $request)
{
    $data = [
        'title' => 'Presensi hari ini '.date('d-M-Y'),
        'tabel' => Pegawai::all(),
        'tanggal' => date('Y-m-d')
    ];
    return view($this->folder . '.hariini', $data);
}

public function perbulan(Request $request)
{
    $data = [
        'title' => 'Presensi Perbulan',
    ];
    return view($this->folder . '.perbulan', $data);
}

public function perbulanloaddata(Request $request)
{
    $bulan = $request->bulan;
    $tahun = $request->tahun;

    $data = [
        'bulan' => $bulan,
        'tahun' => $tahun,
        // 'tabel' => Presensi::perbulan($bulan,$tahun)
        'pegawai' => Pegawai::all(),
        'tanggal' => Presensi::presensibulangroup($bulan,$tahun)
    ];
    return view($this->folder . '.perbulanloaddata', $data);
}

public function semuanya(Request $request)
{
    $data = [
        'title' => 'Semua Presensi',
        'pegawai' => Pegawai::all(),
        'tanggal' => Presensi::presensitanggalgroup()
    ];
    return view($this->folder . '.semuanya', $data);
}

public function lihatpresensi(Request $request)
{

```

```

$pegawai = $request->idpegawai;
$tanggal = $request->tanggal;
$menu = $request->menu;

$linkback = "";
if($menu == 'hariini') {
    $linkback = route('presensi.hariini');
} else if($menu == 'perbulan') {
    $linkback = route('presensi.perbulan');
} else if($menu == 'semuanya') {
    $linkback = route('presensi.semuanya');
}
$data = [
    'title' => 'Informasi Presensi',
    'tanggal' => $tanggal,
    'pegawai' => Pegawai::find($pegawai),
    'tanggalindo' => Carbon::parse($tanggal)->format('d M Y'),
    'masuk' => Presensi::presensidetil($tanggal,'Masuk',$pegawai),
    'pulang' => Presensi::presensidetil($tanggal,'Pulang',$pegawai),
    'izin' => Presensi::presensidetilizin($tanggal,$pegawai),
    'linkback' => $linkback
];
return view($this->folder . '.lihatpresensi', $data);
}

public function konfirmasihapus() {
    return view('admin.pengaturan.konfirmasihapus');
}

public function konfirmasihapusproses() {
    Presensi::truncate();
    return redirect()->route('pengaturan.index')->with('danger','Berhasil
menghapus data presensi');
    // return view('admin.pengaturan.konfirmasihapus');
}

public function formpresensi(Request $request)
{
    $data = [
        'title' => 'Presensi ',
        'pegawai' => Pegawai::pluck('nama','id'),
        'linkback' => route('presensi.hariini')
    ];
    return view($this->folder . '.formpresensi', $data);
}

```



```

<tr>
  <th width="40">NO</th>
  <th>Nama</th>
  <th width="130">NIP</th>
  <th>Jabatan</th>
  <th>Masuk / Keterangan</th>
  <th>Pulang / Keterangan</th>
  <th>Izin</th>
  <th width="100">Aksi</th>
</tr>
</thead>
<tbody>
  @foreach ($tabel as $rs)
    <tr>
      <td>{{ $loop->iteration }}</td>
      <td>{{ $rs->nama }}</td>
      <td>{{ $rs->nip }}</td>
      <td>{{ $rs->jabatan }}</td>
      <td>{{
\
App\Models\Presensi::presensimasuk(date('Y-m-d'),'Masuk',$rs-
>id,'Hadir') }}</td>
      <td>{{
\
App\Models\Presensi::presensipulang(date('Y-m-d'),'Pulang',$rs-
>id,'Hadir') }}</td>
      <td>{{
\
App\Models\Presensi::presensiizin(date('Y-m-d'),$rs->id,'Izin') }}</td>
      <td>
        <a href="{{ {
route('presensi.lihatpresensi',['idpegawai' => $rs->id,'tanggal' =>
$tanggal,'menu' => 'hariini']) }}" class="btn btn-sm btn-primary"><i
class="fa fa-eye"></i> Lihat</a>
      </td>
    </tr>
  @endforeach
</tbody>
</table>
</div>
</div>
</div>
</div>

</div>

@endsection

```

6. AndroidController

```
<?php

namespace App\Http\Controllers;

use App\Models\Koordinat;
use App\Models\Pegawai;
use App\Models\Presensi;
use Illuminate\Http\Request;
use Illuminate\Support\Carbon;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Hash;

class AndroidController extends Controller
{
    public function loginPengguna(Request $request)
    {
        $data = [
            'status' => "gagal",
            'pesan' => 'Anda Gagal Login',
            'data' => [
                'id' => '0',
                'nama' => '0',
                'jabatan' => "",
                'jarak' => '0',
                'latitude' => "",
                'longitude' => ""
            ]
        ];

        $pegawai = Pegawai::where('username', $request->username)->first();
        if ($pegawai) {

            if (Hash::check($request->password, $pegawai->password)) {
                $idkoordinat = 1;
                $koordinat = Koordinat::find($idkoordinat);

                $data['status'] = "berhasil";
                $data['pesan'] = 'Berhasil Login';
                $data['data'] = [
                    'id' => $pegawai->id,
                    'nama' => $pegawai->nama,
                    'jabatan' => $pegawai->jabatan,
                    'jarak' => $koordinat->jarak,
                    'latitude' => $koordinat->latitude,
```

```

        'longitude' => $koordinat->longitude,
    ];
}
}

return $data;
}

public function getpegawai(Request $request)
{
    $idpegawai = $request->iddata;
    $pegawai = Pegawai::find($idpegawai);
    return $pegawai;
}

public function ubahidentitas($id, Request $request) {
    $pegawai = Pegawai::find($id);
    $pegawai->fill($request->all());
    $pegawai->save();
    return $pegawai;
}

public function ubahpassword($iddata, Request $request) {
    $data = [
        'status' => 'gagal',
        'pesan' => 'Gagal ganti password, password lama anda salah'
    ];
    $pbaru = $request->passwordbaru;
    $plama = $request->passwordlama;

    $pegawai = Pegawai::find($iddata);
    $cek = Hash::check($plama, $pegawai->password);

    if($cek) {
        $pegawai->password = Hash::make($pbaru);
        $pegawai->save();
        $data['status'] = 'berhasil';
        $data['pesan'] = 'Password berhasil diganti';
    }

    return $data;
}

//

function cekpresensi($idpegawai, $status) {

```

```

$q = "SELECT * FROM presensi
      WHERE idpegawai = '$idpegawai'
      AND presensi = '$status'
      AND DATE(tanggal) = CURDATE()";
return collect(DB::select(DB::raw($q)));
}

```

```

public function presensihadir(Request $request)
{

```

```

    $idpegawai = $request->idpegawai;
    $kodeqr = $request->kodeqr;

```

```

    $idkoordinat = 1;
    $koordinat = Koordinat::find($idkoordinat);

```

```

    $tanggal = Carbon::now();
    $status = 'Hadir';

```

```

    $latitude = $koordinat->latitude;
    $longitude = $koordinat->longitude;

```

```

    $presensi = Presensi::cekstatuspresensi(); //cek status berdasarkan jam
    pulang/masuk

```

```

    if($kodeqr != \Info::buatkodeqr()) {
        $data = [
            'status' => "berhasil",
            'pesan' => 'Gagal melakukan presensi, kode Qr salah',
            'presensi' => [
                'tanggal' => $tanggal,
                'status' => "",
                'idpegawai' => strval($idpegawai),
                'latitude' => "",
                'longitude' => "",
                'presensi' => "",
                'foto' => $kodeqr
            ]
        ];
    }

```

```

    return $data;

```

```

}

```

```

if($this->cekpresensi($idpegawai,$presensi)->count()) {
    $data = [
        'status' => "berhasil",

```

```

        'pesan' => 'Anda sudah melakukan Presensi ' . ucfirst($presensi),
        'presensi' => [
            'tanggal' => $tanggal,
            'status' => '',
            'idpegawai' => $idpegawai,
            'latitude' => '',
            'longitude' => '',
            'presensi' => '',
            'foto' => $kodeqr
        ]
    ];
    return $data;
}

if ($kodeqr == \Info::buatkodeqr()) {
    $create = Presensi::create([
        'tanggal' => $tanggal,
        'status' => $status,
        'idpegawai' => strval($idpegawai),
        'latitude' => $latitude,
        'longitude' => $longitude,
        'presensi' => $presensi,
        'foto' => $kodeqr
    ]);

    // update status
    $idpresensi = $create->id;
    $statuspresensi = Presensi::cekuntukupdate($idpresensi);
    $presensi = Presensi::find($idpresensi);
    $presensi->presensi = $statuspresensi;
    $presensi->save();

    // return json
    $data = [
        'status' => "berhasil",
        'pesan' => 'Berhasil melakukan presensi ' . ucfirst($statuspresensi),
        'presensi' => [
            'tanggal' => $tanggal,
            'status' => $create->status,
            'idpegawai' => strval($idpegawai),
            'latitude' => $create->latitude,
            'longitude' => $create->longitude,
            'presensi' => $create->presensi,
            'foto' => $kodeqr
        ]
    ];
}

```

```

        return $data;
    }
}

public function presensitidakmasuk(Request $request)
{
    $idpegawai = $request->idpegawai;
    $status = $request->status;
    $foto = $request->foto;

    $idkoordinat = 1;
    $koordinat = Koordinat::find($idkoordinat);

    $tanggal = Carbon::now();

    $latitude = $koordinat->latitude;
    $longitude = $koordinat->longitude;

    $teksstatus = ucfirst(strtolower($status));

    $create = Presensi::create([
        'tanggal' => $tanggal,
        'status' => $teksstatus,
        'idpegawai' => strval($idpegawai),
        'latitude' => $latitude,
        'longitude' => $longitude,
        'presensi' => $status,
        'foto' => $foto
    ]);

    // return json
    $data = [
        'status' => "berhasil",
        'pesan' => 'Data ' . $teksstatus. ' berhasil di simpan',
        'presensi' => [
            'tanggal' => $tanggal,
            'status' => $create->status,
            'idpegawai' => strval($idpegawai),
            'latitude' => $create->latitude,
            'longitude' => $create->longitude,
            'presensi' => $create->presensi,
            'foto' => '-'
        ]
    ];
}

```

```

        return $data;
    }

    public function presensiharian(Request $request)
    {
        $idpegawai = $request->idpegawai;
        $presensiharian = Presensi::presensiharianpegawai($idpegawai);
        return $presensiharian;
    }

    public function presensibulanan(Request $request)
    {
        $idpegawai = $request->iddata;
        $tahun = $request->tahun;
        $bulan = $request->bulan;

        $presensibulanan = Presensi::presensibulanpegawai($idpegawai,
        $bulan, $tahun);
        return $presensibulanan;
    }

    public function loadpegawai()
    {
        return Pegawai::all();
    }

    public function generatekodeqr()
    {
    }
}

```