

**SISTEM INFORMASI KEPEGAWAIAN PADA KEJAKSAAN NEGERI
LAMPUNG TIMUR**

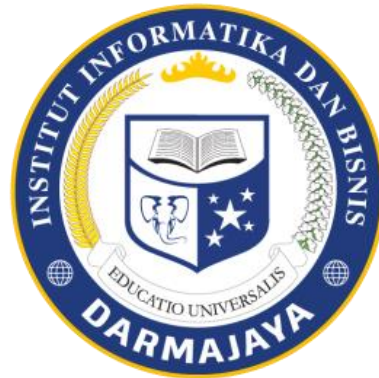
SKRIPSI

Diajukan sebagai salah satu syarat untuk mencapai gelar

SARJANA KOMPUTER

Pada Program Studi Sistem Informasi

IIB Darmajaya Bandar Lampung



Disusun oleh

Laura Desvia

1511050117

JURUSAN SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA

BANDAR LAMPUNG

2019



PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi yang saya ajukan ini adalah hasil karya saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi atau karya yang pernah ditulis atau diterbitkan orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Karya ini adalah milik saya dan pertanggungjawaban sepenuhnya berada di pundak saya.

Bandar Lampung, 16 September 2019



Laura Desvia
NPM. 1511050117

HALAMAN PERSETUJUAN

**JUDUL : SISTEM INFORMASI KEPEGAWAIAN PADA
KEJAKSAAN NEGERI LAMPUNG TIMUR**

NAMA : LAURA DESVIA

NPM : 1511050117

JURUSAN : S1 SISTEM INFORMASI



Pembimbing

Ketua Jurusan

Dona Yulianiwati, S.Kom., M.T.I
NIK. 00780204

Nurulko, S.Kom., M.T.I
NIK. 00440702

HALAMAN PENGESAHAN

Telah diuji dan dipertahankan didepan Tim Penguji SKRIPSI Program Studi Sistem Informasi IIB Darmajaya dan dinyatakan diterima untuk memenuhi syarat guna memperoleh gelar Sarjana Komputer

Mengesahkan,

1. Tim penguji

Tanda Tangan

Anggota 1

Neni Purwati, S.Kom., M.T.I



Anggota 2

Deppi Linda, S.Kom, M.T.I



2. Dekan Fakultas Ilmu Komputer IIB Darmajaya



Tanggal Lulus Ujian Skripsi : **16 SEPTEMBER 2019**

DAFTAR RIWAYAT HIDUP

1. Identitas

- a. Nama : Laura Desvia
- b. NPM : 1511050117
- c. Tempat/Tanggal Lahir : Sukadana, 30 Desember 1996
- d. Agama : Islam
- e. Alamat : Des. Surabaya Udik Kec.
Sukadana Lampung Timur
- f. Kewarganegaraan : Indonesia
- g. E-mail : lauradesvia2016@gmail.com
- h. HP : 085268931577

2. Riwayat Pendidikan

- a. Sekolah Dasar : SD N 5 Sukadana
- b. Sekolah Menengah Pertama : SMP N 1 Sukadana
- c. Sekolah Menengah Atas : SMA N1Sukadana

Bandar lampung, 16 September 2019

Laura Desvia
NPM 1511050117

MOTTO

*“Tidak ada kemudahan melainkan dengan
pertolongan Allah.”*

*Tidak akan berhenti sampai hari ini, yakin hari esok
akan lebih baik*

(penulis)

INTISARI

SISTEM INFORMASI KEPEGAWAIAN PADA KEJAKSAAN NEGERI LAMPUNG TIMUR

Oleh:

Laura Desvia
1511050117

Lauradesvia2016@gmail.com

Peningkatan efisiensi dan efektivitas Kejaksaan Negeri dapat dicapai melalui Sumber Daya Manusia (SDM) yang baik. Sumber Daya Manusia (SDM) yang baik dapat dilihat melalui data kepegawaian seperti presensi kerja, data kenaikan pangkat dan data mutasi pegawai. Pada penelitian yang dilakukan di Kejaksaan Negeri Lampung Timur data presensi pegawai, data mutasi dan kenaikan pangkat belum terkomputerisasi dan terintegrasi dengan baik. Hal tersebut dapat menyebabkan pencarian dan pengolahan data tidak optimal. Oleh karena itu dibutuhkan sistem yang terintegrasi agar data tersebut dapat diolah dengan baik.

Sistem yang dibangun terdiri dari sistem presensi pegawai Kejaksaan yang berbasis *Android* dengan pemanfaatan *QR code*. *QR code* tersebut dibaca menggunakan aplikasi *desktop*. Sistem mutasi dan kenaikan pangkat berbasis *desktop* dengan bahasa pemrograman yang digunakan adalah *Java*. Data presensi, kenaikan pangkat serta mutasi yang dikelola tersimpan dalam *database* menggunakan *MySQL*.

Dengan adanya sistem tersebut maka pengolahan data pegawai, data mutasi dan kenaikan pangkat dapat dikelola dengan baik. Adanya *QR code* pada presensi pegawai dapat mengurangi kecurangan data karena presensi dilakukan dengan cara membaca *QR code* melalui *smartphone* masing-masing pegawai. Pencarian dan pengolahan data mutasi dan kenaikan pangkat yang terintegrasi dapat memudahkan pengelola dalam mengelola data tersebut.

Kata Kunci : Kejaksaan Negeri Lampung Timur, *QR code*, *Android*, *Desktop*, *MySQL*

ABSTRACT

EMPLOYEE INFORMATION SYSTEMS IN THE PUBLIC PROSECUTOR'S OFFICE EAST LAMPUNG

By:

Laura Desvia

1511050117

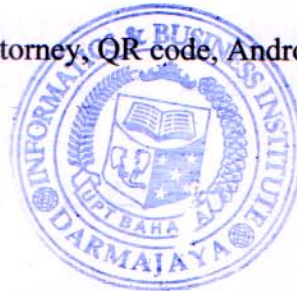
Lauradesvia2016@gmail.com

Increasing the efficiency and effectiveness of the Public Prosecutor's Office can be achieved through good Human Resources (HR). Good Human Resources (HR) can be seen through staffing data such as work attendance, promotion data and employee mutation data. In the study conducted at the East Lampung District Prosecutor's office, employee attendance data, mutation data and promotion were not computerized and well integrated. This can cause the search and data processing is not optimal. Therefore an integrated system is needed so that the data can be processed properly.

The system that was built consisted of the presence of the Attorney General's employee attendance system based on the use of QR codes. The QR code was read using the desktop application. The mutation and promotion system based on desktop with the programming language used was Java. The presence data, the promotions and managed mutations were stored in the database using MySQL.

With this system the processing of employee data, the mutation data and the promotion can be managed properly. The presence of QR codes in employee attendance can reduce data fraud because attendance is done by reading the QR code through the smartphone of each employee. Integrated data search and processing of mutations and promotions can make it easier for the managers to manage the data.

Keywords: East Lampung District Attorney, QR code, Android, Desktop, MySQL



PRAKATA

Assalamu'alaikum Wr, Wb

Segala puji kehadiran Allah SWT yang telah memberikan rahmat Taufiq dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul **SISTEM INFORMASI KEPEGAWAIAN PADA KEJAKSAAN NEGERI LAMPUNG TIMUR**. Penulisan Skripsi ini merupakan salah satu syarat untuk menyelesaikan studi pada program S1 Sistem Informasi di perguruan tinggi IIB Darmajaya Bandar Lampung. Penulis menyadari bahwa masih banyak kekurangan dalam penulisan skripsi ini dari bantuan berbagai pihak, karena itu dengan segala kerendahan hati penulis mengucapkan banyak terimakasih kepada

1. Puji syukur kehadiran Allah SWT. Yang telah memberikan rahmat dan hidayahnya sehingga saya dapat menyelesaikan skripsi ini.
2. Nabi Muhammad SAW. Yang telah menjadi sumber inspirasi dalam berbagai macam hal dan tindakan dalam hidup saya.
3. Untuk papi dan mami tercinta Idham Kholik dan Irma Yulia Sari yang terus selalu tanpa henti memberi motivasi, dukungan serta doa. Tanpa adanya kalian saya tidak akan pernah menjadi seperti ini.
4. Bapak Ir. Firmansyah YA. MBA,MSc., selaku Rektor IIB Darmajaya.
5. Bapak Sriyanto, S.kom, MM.,Ph.D selaku Dekan Fakultas Ilmu Komputer IIB Darmajaya Bandar Lampung.
6. Bapak Nurjoko, S.kom, M.T.I selaku Ketua Jurusan Sistem Informasi IIB Darmajaya.
7. Ibu Dona Yuliawati,S.Kom., M.T.I selaku dosen pembimbing yang selalu memberikan bimbingan dan motivasi kepada saya.
8. Untuk adik saya Rio Irawan yang telah memberi arti susahny menjadi kakak yang baik

9. Untuk kakak saya Lady Shella yang telah memberi arti susahnya dalam menyelesaikan skripsi ini.
10. Untuk nenek saya yang telah menemani dan menjaga saya selama kuliah ini.
11. Teman-teman seperjuangan Sistem Informasi.
12. Almamater IIB Darmajaya.

Semoga Allah SWT membalas kebaikan pihak yang telah membantu proses skripsi penulis.

Wassalamu'alaikum Wr, Wb.

Bandar Lampung, 16 September 2019
Penulis

Laura Desvia
1511050117

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERNYATAAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERSEMBAHAN	v
MOTTO	vi
DAFTAR RIWAYAT HIDUP	vii
ABSTRAK	viii
KATA PENGANTAR	x
DAFTAR ISI	xii
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan masalah	2
1.3 Ruang lingkup	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	4
2.1 Struktur Organisasi	4
2.2 Definisi Sistem	5
2.3 Definisi informasi	5
2.4 Definisi Sistem Informasi	6
2.5 Qr code	7
2.6 Metode Pengembangan Sistem	8

2.7	Metode Pengumpulan Data	8
2.8	Alat Pengembangan Sistem.....	9
2.9	<i>Database Manangement System</i>	12
2.10	<i>Java</i>	14
2.11	Android	14
2.12	Android Studio.....	15
2.13	<i>My SQL</i>	16
2.14	Penelitian Terkait.....	16
BAB III METODOLOGI PENELITIAN		18
3.1	Metode Pengumpulan Data	18
3.2	Metode Pengembangan Sistem	18
3.2.1	<i>Communication</i>	18
3.2.1.1	Analisis Sistem Berjalan.....	19
3.2.1.2	Analisis Permasalahan	20
3.2.2	<i>Quick Plan</i>	21
3.2.3	<i>Modelling Quick Design</i>	21
3.2.3.1	Arsitektur Sistem.....	21
3.2.3.2	<i>Use Case Diagram</i>	22
3.2.3.3	<i>Activity Diagram</i>	24
3.2.3.4	<i>Class Diagram</i>	30
3.2.3.5	Interface Sistem.....	30
3.2.3.5.1	Interface Sistem Berbasis Dekstop	31
3.2.3.5.2	Interface Sistem Berbasis Android	32
3.2.4	<i>Construction of prototype</i>	39
3.2.5	<i>Deployment delivery and Feedback</i>	40
BAB IV HASIL DAN PEMBAHASAN.....		41
4.1	Alat Pendukung Pembuatan Sistem	41
4.1.1	Perangkat Lunak(<i>Software</i>)	41
4.1.2	Perangkat Keras(<i>Hardware</i>)	41

4.2 Implementasi Sistem	41
4.2.1 Implementasi Sistem Desktop.....	42
4.2.2 Implementasi Sistem Android.....	48
BAB V KESIMPULAN DAN SARAN.....	50
5.1 Kesimpulan.....	50
5.2 Saran.....	50

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar	Halaman
2.1 Struktur Organisasi Kejaksaan Negeri Lampung Timur	5
2.2 Bagan Alur metode <i>prototype</i>	7
3.1 <i>Use Case Diagram</i> prosedur yang sedang berjalan	20
3.2 Perancangan Arsitektur yang diusulkan	22
3.3 Perancangan <i>Use case Diagram</i> Sistem yang diusulkan	23
3.4 Perancangan <i>Activity Diagram</i> login Kaur Kepegawaian	24
3.5 Perancangan <i>Activity Diagram</i> mengola Data Pegawai	25
3.6 Perancangan <i>Activity Diagram</i> mengola Data Mutasi	26
3.7 Perancangan <i>Activity Diagram</i> mengola Data Kenaikan pangkat	26
3.8 Perancangan <i>Activity Diagram</i> lihat data absen	27
3.9 Perancangan <i>Activity Diagram</i> laporan.....	27
3.10 Perancangan <i>Activity Diagram</i> login pegawai kejaksaan	29
3.11 Perancangan <i>Activity Diagram</i> absen.....	29
3.12 Perancangan <i>Class Diagram</i>	30
3.13 Perancangan <i>Interface</i> login kaur kepegawaian	31
3.14 Perancangan <i>Interface</i> Halaman utama kaur kepegawaian	32
3.15 Perancangan <i>Interface</i> Halaman tambah pegawai	32
3.16 Perancangan <i>Interface</i> Halaman mutasi pegawai	33
3.17 Perancangan <i>Interface</i> Halaman tambah data mutasi pegawai.....	34
3.18 Perancangan <i>Interface</i> Halaman kenaikan pangkat	35
3.19 Perancangan <i>Interface</i> tambah data kenaikan pangkat	36
3.20 Perancangan <i>Interface</i> lihat absen pegawai.....	36
3.21 Perancangan <i>Interface</i> kurva grafik absen	37
3.22 Perancangan <i>Interface</i> Halaman admin.....	37
3.23 Perancangan <i>Interface</i> Halaman tambah admin	38
3.24 Perancangan <i>Interface</i> Halaman login pegawai	38
3.25 Perancangan <i>Interface</i> Halaman utama absen pegawai	39
3.26 Perancangan <i>Interface</i> pembaca <i>qr qode</i>	39

4.1 Implementasi Halaman <i>login</i> kaur kepegawaian	42
4.2 Implementasi Halaman <i>input</i> pegawai	43
4.3 Implementasi Halaman utama kaur kepegawaian.....	43
4.4 Implementasi Halaman <i>output</i> data pegawai.....	44
4.5 Implementasi Halaman <i>input</i> data mutasi	44
4.6 Implementasi Halaman proses mutasi.....	45
4.7 Implementasi Halaman <i>output</i> data mutasi	45
4.8 Implementasi Halaman <i>input</i> data kenaikan pangkat	46
4.9 Implementasi Halaman proses data kenaikan pangkat.....	46
4.10 Implementasi Halaman output data kenaikan pangkat.....	47
4.11 Implementasi Halaman sebelum absen pegawai	47
4.12 Implementasi Halaman lihat absen pegawai	48
4.13 Implementasi <i>login</i> pegawai berbasis android.....	48
4.14 Implementasi Halaman utama absensi pegawai berbasis android	49
4.15 Implementasi Halaman Halaman pembaca <i>Qr Qode</i>	49

DAFTAR TABEL

Tabel	Halaman
2.1 Tabel Simbol <i>Use Case Diagram</i>	10
2.2 Tabel Simbol <i>Activity Diagram</i>	11
2.3 Tabel Simbol <i>Class Diagram</i>	12
2.4 Tabel Versi android.....	15

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah

Kejaksaan Negeri yang biasa disingkat Kejari adalah lembaga kejaksaan yang berkedudukan di ibu kota kabupaten/kota dan daerah hukumnya meliputi wilayah kekuasaan kabupaten/kota. Mengacu pada Undang-Undang No. 16 Tahun 2004 yang menggantikan UU No. 5 Tahun 1991 tentang Kejaksaan R.I., Kejaksaan sebagai salah satu lembaga penegak hukum dituntut untuk lebih berperan dalam menegakkan supremasi hukum, perlindungan kepentingan umum, penegakan hak asasi manusia, serta pemberantasan Korupsi, Kolusi, dan Nepotisme (KKN). Di dalam UU Kejaksaan yang baru ini, Kejaksaan RI sebagai lembaga negara yang melaksanakan kekuasaan negara di bidang penuntutan harus melaksanakan fungsi, tugas, dan wewenangnya secara merdeka, terlepas dari pengaruh kekuasaan pemerintah dan pengaruh kekuasaan lainnya (Pasal 2 ayat 2 Undang-Undang Nomor 16 Tahun 2004) (www.kejaksaan.go.id).

Peningkatan efisiensi dan efektivitas Kejaksaan Negeri dapat dicapai melalui Sumber Daya Manusia (SDM) yang baik. Sumber Daya Manusia (SDM) yang baik dapat dilihat melalui data kepegawaian seperti presensi kerja, data kenaikan pangkat dan data mutasi pegawai. Pada penelitian yang dilakukan di Kejaksaan Negeri Lampung Timur, presensi pegawai dilakukan dengan cara mengisi buku presensi setiap hari pada jam masuk kerja. Proses tersebut dapat menimbulkan kecurangan pada pengisian data presensi pegawai jika ada pegawai yang “menitip” presensi (pemalsuan tanda tangan jika pegawai tersebut tidak hadir) kepada pegawai lain. Disamping presensi pegawai, pengolahan data mutasi dan kenaikan pangkat juga belum tersistem dengan baik. Data mutasi dan kenaikan pegawai yang diterima oleh Pimpinan diolah menggunakan aplikasi *office* yang belum terkoneksi dengan basis data yang membuat pegawai sulit melakukan pencarian data yang dibutuhkan. Disamping itu, Surat Keputusan mutasi dan kenaikan pangkat pegawai yang diterima diarsipkan. Ketika Pimpinan membutuhkan nomor SK mutasi atau kenaikan pangkat untuk kebutuhan tertentu,

pencarian nomor SK sulit dilakukan dikarenakan arsip data SK tersebut tidak tersusun dengan rapi. Penyimpanan data yang tidak teratur dapat menyebabkan terjadinya kesalahan (*human error*) dalam penghapusan data. Untuk mengembalikan kondisi data yang terhapus seperti semula sulit dilakukan terlebih pada data yang tidak memiliki *back up* data. Oleh karena itu, pada penelitian ini peneliti mengangkat judul **“SISTEM INFORMASI KEPEGAWAIAN PADA KEJAKSAAN NEGERI LAMPUNG TIMUR”**. Penelitian ini diharapkan agar memudahkan dalam proses presensi pegawai dan pengolahan data pegawai seperti kenaikan pangkat serta mutasi di Kejaksaan Negeri Lampung Timur.

1.2 Perumusan Masalah

Dari latar belakang masalah yang telah diuraikan sebelumnya, maka dapat dirumuskan masalah “Bagaimana membangun sistem informasi kepegawaian Kejaksaan Negeri Lampung Timur?”

1.3 Ruang Lingkup Penelitian

Penelitian dilakukan pada Kejaksaan Negeri Lampung Timur mulai dari bulan November 2018 sampai dengan Januari 2019 dengan batasan penelitian sebagai berikut :

- a. Penelitian hanya membahas mengenai presensi pegawai, data mutasi pegawai dan data kenaikan pangkat pegawai.
- b. Presensi pegawai dilakukan dengan menggunakan *QR Code* berbasis Android, sedangkan pengolahan data mutasi dan kenaikan pangkat berbasis *desktop*.

1.4 Tujuan

Tujuan dari penelitian ini adalah membangun sebuah sistem yang dapat membantu dalam pengolahan data kenaikan pangkat, data mutasi serta presensi pegawai di Kejaksaan Negeri Lampung Timur.

1.5 Manfaat Penelitian

Manfaat dari penelitian yang dilakukan adalah sebagai berikut :

1. Memudahkan presensi pegawai Kejaksaan Negeri Lampung Timur.
2. Memudahkan Kaur Kepegawaian dalam melakukan pencarian data dan pembuatan laporan data pegawai mutasi serta kenaikan pangkat di Kejaksaan Negeri Lampung Timur.

1.6 Sistematika Penulisan

Secara garis besar penelitian ini terdiri dari 5 (lima) bab dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Pada bab ini diuraikan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II Tinjauan Pustaka

Pada bab ini berisi teori-teori yang berkaitan dan mendukung penelitian serta penulisan skripsi ini yang akan dilakukan oleh penulis/peneliti.

BAB III Metodologi Penelitian

Dalam bab ini dijelaskan tentang metodologi penelitian penyelesaian masalah yang dijelaskan di perumusan masalah yang meliputi metode pengumpulan data, metode pengembangan sistem, serta alat dan bahan pendukung.

BAB IV Hasil Penelitian dan Pembahasan

Pada bab ini akan dibahas tentang hasil penelitian yang berupa aplikasi yang dibangun, termasuk cara pengoperasiannya.

BAB V Simpulan dan Saran

Bab ini berisikan simpulan dari seluruh pembahasan dan saran yang diperlukan untuk perbaikan dimasa yang akan datang.

Daftar Pustaka

Lampiran

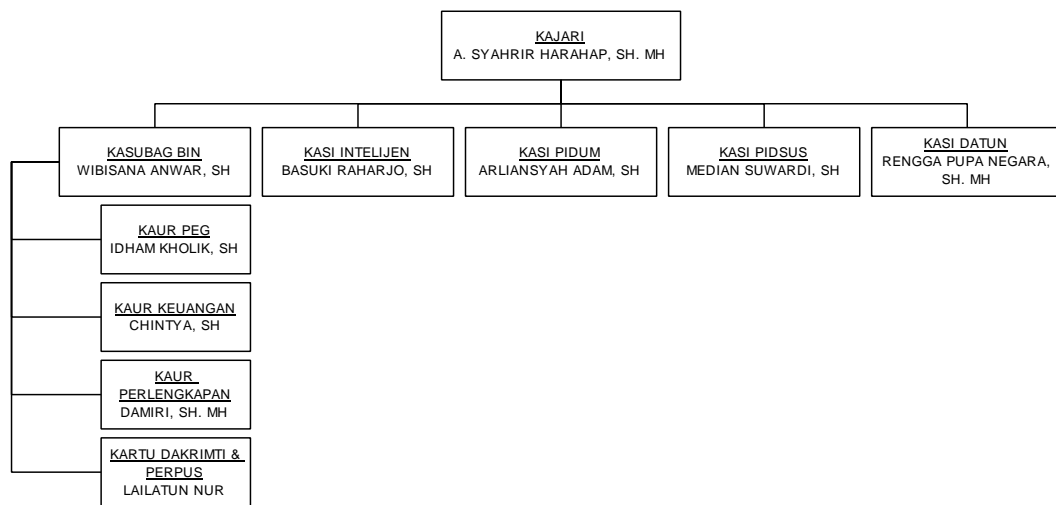
BAB II TINJAUAN PUSTAKA

2.1 Latar Belakang Kejaksaan Negeri Lampung Timur

Mengacu pada sistem Pemerintahan yang baik (Good Governance) Kejaksaan Negeri Lampung Timur adalah merupakan bagian dari Lembaga Pemerintah yang melaksanakan tugas kekuasaan Negara sebagai Penegak Hukum dibidang Penuntutan yang berkedudukan di wilayah Kabupaten Lampung Timur di Sukadana dan merupakan Instansi Vertikal di bawah Kejaksaan Tinggi Lampung dan Kejaksaan Agung Republik Indonesia.

Sebagai lembaga Penegak Hukum di daerah Kabupaten Lampung Timur yang mengendalikan pelaksanaan tugas dan wewenang Kejaksaan di daerah Hukumnya Kejaksaan Negeri Lampung Timur dari waktu ke waktu selalu berupaya untuk terus meningkatkan Profesionalisme, Integritas, dan Disiplin (PID) pegawainya agar dapat mengembang fungsi penyidikan dan penuntutan dalam rangka pembentukan Jaksa yang berintikan keadilan dan kebenaran pemerintah yang baik merupakan suatu prasyarat bagi setiap Instansi pemerintah untuk melaksanakan tugas pokok dan fungsi institusi dalam hal tersebut diperlukan pengembangan dan penerapan sistem yang tepat jelas sehingga penyenggaraan pemerintah dan pengembangan dapat berlangsung berdaya guna serta bebas dari KKN.

Dalam susunan ketatanegaraan, Kejaksaan adalah lembaga pemerintah yang melaksanakan kekuasaan Negara, khususnya dibidang penuntutan, Kejaksaan Negeri adalah lembaga Pemerintah yang berwenang dalam penegakan hukum dan keadilan. Kejaksaan Negeri dipimpin oleh seorang Kepala Kejaksaan Negeri dan bertanggung jawab kepada Kepala Kejaksaan Tinggi. Kejaksaan Negeri merupakan institusi pelaksana kekuasaan negara serta kewenangan lain yang berkedudukan dan pada lingkup daerah hukum di daerah tingkat II yang menjalankan tugas penegakan hukum khususnya dibidang penuntutan, yang berdomisili di kota/kabupaten berdasarkan Undang-Undang Nomor 16 tahun 2004. Adapun struktur organisasi Kejaksaan Negeri Lampung Timur adalah seperti pada Gambar 2.1.



Gambar 2.1 Struktur Organisasi Kejaksaan Negeri Lampung Timur

2.2 Sistem

Pada dasarnya, sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran, jika sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem (Abdul Kadir, 2014).

2.3 Informasi

McFadden, dan kawan-kawan mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Shannon dan Weaver, dua orang insinyur listrik melakukan pendekatan secara matematis untuk mendefinisikan informasi (Kroenke). Menurut mereka, informasi adalah jumlah ketidakpastian yang dikurangi ketika sebuah pesan diterima. Artinya, dengan adanya informasi, tingkat kepastian menjadi meningkat. Menurut Davis, informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang (Abdul Kadir, 2014).

2.4 Sistem Informasi

Sesungguhnya yang dimaksud sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (Computer Based Information System atau CBIS). Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa berbasis komputer, walaupun dalam kenyataannya komputer merupakan bagian yang penting. Di buku ini, yang dimaksudkan dengan sistem informasi adalah sistem informasi berbasis komputer. Ada beragam definisi sistem informasi, yaitu :

- a. Alter, sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
- b. Bodnar dan Hopwoo, sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
- c. Gelinas, Oram dan Wiggins, sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi keluaran kepada para pemakai.
- d. Hall, Sistem informasi adalah sebuah rangkaian prosedur formal, dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada para pemakai.
- e. Turban, McLean dan Wetherbe, Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi untuk tujuan yang spesifik.
- f. Wilkinson, Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia dan komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi) guna mencapai sasaran-sasaran perusahaan.

Berdasarkan berbagai definisi tersebut, dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi dan

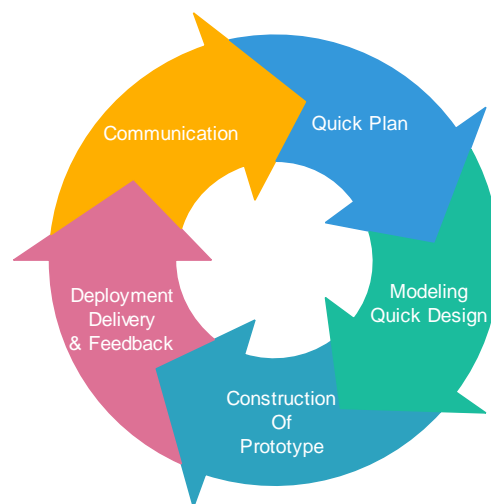
prosedur kerja), ada sesuatu yang diproses (data menjadi informasi) dan dimaksudkan untuk mencapai suatu sasaran atau tujuan (Abdul Kadir, 2014).

2.5 QR Code (*Quick Response Code*)

QR Code adalah sebuah kode batang dua dimensi yang ditemukan oleh sebuah perusahaan Jepang bernama Denso Wave pada tahun 1994. *QR Code* ini adalah pengembangan dari kode batang sebelumnya. Pada model barcode lama, data disimpan secara horizontal saja sedangkan pada *QR Code*, data disimpan baik secara vertikal maupun horizontal (Ermatita, 2017).

2.6 Metode Pengembangan Sistem

Metode *prototype* suatu proses pembuatan *software* yang bersifat berulang dan dengan perencanaan yang cepat yang dimana terdapat umpan balik yang memungkinkan terjadinya perulangan dan perbaikan *software* sampai dengan *software* tersebut memenuhi kebutuhan dari sisi pengguna. Siklus atau ilustrasi dari metode prototipe dapat dilihat pada Gambar 2.2.



Gambar 2.2 Ilustrasi Model Prototipe (Sumber : Pressman, 2017)

Pembuatan prototipe dimulai dengan dilakukannya komunikasi antara tim pengembang perangkat lunak dengan para pelanggan. Tim pengembang perangkat lunak akan melakukan pertemuan-pertemuan dengan para stakeholder untuk

mendefinisikan sasaran keseluruhan untuk perangkat lunak yang akan dikembangkan, mengidentifikasi spesifikasi kebutuhan apa pun yang saat ini diketahui, dan menggambarkan area-area dimana definisi lebih jauh pada iterasi selanjutnya merupakan keharusan. Iterasi pembuatan prototipe direncanakan dengan cepat dan pemodelan (dalam bentuk "rancangan cepat") dilakukan. Suatu rancangan cepat berfokus pada representasi semua aspek perangkat lunak yang akan dilihat oleh para pengguna akhir (misalnya rancangan antar muka pengguna (*user interface*) atau format tampilan). Rancangan cepat (*quick design*) akan memulai konstruksi pembuatan prototipe. Prototipe kemudian akan diserahkan kepada para stakeholder dan kemudian mereka akan melakukan evaluasi-evaluasi tertentu terhadap prototipe yang telah dilakukan sebelumnya, kemudian akhirnya akan memberikan umpan-balik yang akan digunakan untuk memperhalus spesifikasi kebutuhan. Iterasi akan terjadi saat prototipe diperbaiki untuk memenuhi kebutuhan dari para stakeholder, sementara pada saat yang sama memungkinkan kita untuk lebih memahami kebutuhan apa yang akan dikerjakan pada saat iterasi selanjutnya.

Idealnya, prototipe bertindak sebagai mekanisme untuk mengidentifikasi spesifikasi-spesifikasi kebutuhan perangkat lunak. Jika suatu prototipe yang dapat digunakan akan dikembangkan, kita bisa menggunakan program yang sudah ada sebelumnya atau dengan menerapkan penggunaan perangkat yang sudah ada misalnya perangkat pembentuk laporan (*report generator*) atau aplikasi untuk melakukan perancangan antarmuka (*window manager*) yang memungkinkan program yang dapat digunakan dapat dibuat dengan mudah dan cepat (Pressman, 2012).

2.7 Metode Pengumpulan Data

Menurut Rosa dan Shalahuddin (2018), hal pertama yang dilakukan dalam analisis sistem adalah melakukan pengumpulan data. Ada beberapa teknik pengumpulan data yang sering dilakukan, yaitu :

- a. Teknik Wawancara

Pengumpulan data dengan menggunakan wawancara memiliki beberapa keuntungan, yaitu :

1. Lebih mudah dalam menggali bagian sistem mana yang dianggap baik dan bagian mana yang dianggap kurang baik.
2. Jika ada bagian tertentu yang perlu digali lebih dalam, maka dapat menanyakannya langsung kepada narasumber.
3. Dapat menggali kebutuhan user secara lebih bebas.
4. *User* dapat mengungkapkan kebutuhannya secara lebih bebas.

b. Teknik Observasi

Pengumpulan data dengan menggunakan observasi mempunyai keuntungan, yaitu :

1. Ananlisis dapat melihat langsung bagaimana sistem lama berjalan.
2. Mampu menghasilkan gambaran lebih baik jika dibanding dengan teknik lainnya.

c. Teknik Studi Pustaka

Studi pustaka dilakukan untuk memperoleh data dan informasi dengan membaca berbagai bahan penulisan, karangan ilmiah serta sumber-sumber lain mengenai permasalahan yang berhubungan dengan penulisan.




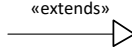
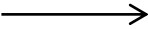
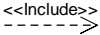
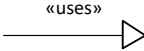
2.8 Alat Pengembangan Sistem

Alat pengembangan sistem yang digunakan menggunakan pemodelan *Unified Modeling Language (UML)* *use case diagram*, *activity diagram*, dan *class diagram*.

2.8.1 Use Case Diagram

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami (Rosa dan Shalahuddin, 2018). Adapun simbol-simbol *use case diagram* dapat di lihat pada Tabel 2.1.

Tabel 2.1 Simbol *Use Case Diagram*

Keterangan	Simbol	Deskripsi
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; dinyatakan dengan menggunakan kata kerja diawal-awal frase nama <i>use case</i>
Aktor		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
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.
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; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan <i>/include/uses</i>	 	<p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <ol style="list-style-type: none"> <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.

2.8.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa dan Shalahuddin, 2018). Simbol-simbol yang terdapat pada *activity diagram* adalah seperti pada Tabel 2.2.

Tabel 2.2 Simbol Diagram Aktivitas

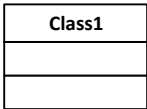
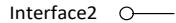



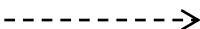

Keterangan	Simbol	Deskripsi
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.8.3 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan

variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa dan Shalahuddin, 2018). Simbol-simbol yang ada pada diagram kelas adalah seperti pada Tabel 2.3.

Tabel 2.3 Simbol *Class Diagram*

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

2.9 Database Management System (DBMS)

Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data

di maksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut *Database Management System* (DBMS). DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien. DBMS dapat digunakan untuk mengakomodasikan berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda-beda. Terdapat beberapa elemen basis data, yaitu (Abdul Kadir, 2014):

a. *Database*

Database atau basis data adalah kumpulan tabel yang mempunyai kaitan antara suatu tabel dengan tabel lainnya sehingga membentuk suatu bangunan data.

b. Tabel

Tabel adalah kumpulan *record-record* yang mempunyai panjang elemen yang sama dan atribut yang sama namun berbeda data valuenya.

c. Entitas

Entitas adalah sekumpulan objek yang terdefiniskan yang mempunyai karakteristik sama dan bisa dibedakan satu dengan lainnya. Objek dapat berupa barang, orang, tempat atau suatu kejadian.

d. Atribut

Atribut adalah deskripsi data yang bisa mengidentifikasi entitas yang membedakan entitas tersebut dengan entitas yang lain. Seluruh atribut harus cukup untuk menyatakan identitas objek atau dengan kata lain, kumpulan atribut dari setiap entitas dapat mengidentifikasi keunikan suatu individu.

e. *Data Value* (Nilai Data)

Data value adalah data aktual atau informasi yang disimpan pada tiap data, elemen atau atribut. Atribut nama pegawai menunjukkan tempat dimana informasi nama karyawan disimpan, nilai datanya misalnya adalah Anjang, Arif, Suryo dan lain-lain yang merupakan isi data nama pegawai tersebut.

f. *File*

File adalah kumpulan record sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda nilai datanya.

g. *Record/Tuple*

Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu record mewakili satu data atau informasi.

2.10 Java

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, desktop, web dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source* (Joyce, 2017).

2.11 Android

Android adalah sistem operasi dan *platform* pemrograman yang dikembangkan oleh Google untuk ponsel cerdas dan perangkat seluler lainnya (seperti tablet). Android bisa berjalan di beberapa macam perangkat dari banyak produsen yang berbeda. Android menyertakan *kit development* perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna Android. Android Juga menyediakan pasar untuk mendistribusikan aplikasi. secara keseluruhan, Android menyatakan ekosistem untuk aplikasi seluler (*Google Developer Training Team, 2016*).

Google menyediakan peningkatan versi bertahap utama untuk sistem operasi Android setiap enam hingga sembilan bulan, menggunakan nama bertema makanan. Adapun versi-versi sistem operasi Android adalah seperti pada Tabel 2.4.

Tabel 2.4 Versi Android

Nama Kode	Nomor Versi	Tanggal Rilis Awal	API Level
N/A	1.0	23 September 2008	1
N/A	1.1	9 Febuari 2009	2
Cupkace	1.5	27 April 2009	3
Donut	1.6	15 September 2009	4
Eclair	2.0-2.1	26 Oktober 2009	5-7
Froyo	2.2-2.2.3	20 Mei 2010	8
Gingerbread	2.3-2.3.7	6 Desember 2010	9-10
Honeycomb	3.0-3.2.6	22 Febuari 2011	11-13
Ice Cream Sandwitch	4.0-4.0.4	18 Oktober 2011	14-15
Jelly Bean	4.1-4.3.1	9 Juli 2012	16-18
KitKat	4.4-4.4.4	31 Oktober 2013	19-20
Lolipop	6.0-6.0.1	5 Oktober 2015	23
Nougat	7.0	22 Agustus 2016	24
Oreo	8.0	21 Agustus 2017	26
Pie	9.0	16 Agustus 2018	28

Untuk membantu mengembangkan aplikasi secara efisien, Google menawarkan Lingkungan *Development* Terintegrasi (IDE) Java Lengkap yang disebut Android Studio, dengan fitur lanjutan untuk pengembangan, *debug*, dan pemaketan aplikasi Android. Pendistribusian aplikasi Android dapat dilakukan dengan berbagai cara, yaitu melalui email, situs web, atau pasar aplikasi Google Play. Google Play adalah layanan distribusi digital yang dioperasikan dan dikembangkan oleh Google dan berfungsi sebagai toko aplikasi resmi untuk Android yang memungkinkan konsumen mengunduh dan menjelajah aplikasi yang dikembangkan dengan Android SDK dan dipublikasikan melalui Google.

2.12 Android Studio

Android Studio menyediakan alat untuk pengujian, dan mempublikasikan tahap proses development, serta lingkungan development terpadu untuk membuat

aplikasi bagi semua perangkat Android. Lingkungan development menyertakan kode template dengan kode contoh untuk fitur aplikasi umum, alat pengujian dan kerangka kerja yang banyak, dan sistem pembangunan yang fleksibel (*Google Developer Training Team, 2016*).

2.13 MySQL

Menurut Solichin (2016), MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak seperti Apache yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. Beberapa kelebihan MySQL antara lain : *free* (bebas di *download*), stabil dan tangguh, fleksibel dengan berbagai pemrograman, *security* yang baik, dukungan dari banyak komunitas, kemudahan *management database*, mendukung transaksi dan perkembangan *software* yang cukup cepat.

2.14 Penelitian Terkait

Penelitian terdahulu yang terkait dengan penelitian yang dilakukan saat ini adalah sebagai berikut :

- a. Ermatita (2017) dalam penelitiannya menyimpulkan bahwa sistem absensi menggunakan *QR Code* dapat membantu siswa untuk mengefisiensikan waktu dalam melakukan kegiatan absensi.

- b. Menurut MT Nuddin (2015) dalam penelitiannya menyimpulkan bahwa sistem absensi asisten dosen menggunakan *QR Code Scanner* menghasilkan informasi absen asisten dosen yang tepat karena tercantum waktu dan tidak bisa dirubah.

BAB III METODOLOGI PENELITIAN

3.1 Metode Pengumpulan Data

Pengumpulan data yang digunakan dalam menyusun serta melengkapi data adalah dengan cara sebagai berikut :

a. Observasi

Pengamatan langsung diadakan untuk memperoleh data yang dilakukan pada instansi terkait dengan penelitian yang dilakukan di Kejaksaan Negeri Lampung Timur.

b. Wawancara

Wawancara dilakukan dengan cara berkomunikasi langsung dengan Bapak Idham Kholik, SH selaku Kaur Kepegawaian Kejaksaan Negeri Lampung Timur mengenai presensi pegawai, dan pengolahan data mutasi serta kenaikan pangkat pegawai.

c. Studi Pustaka

Studi pustaka dilakukan untuk memperoleh data dan informasi dengan membaca berbagai bahan penulisan karangan ilmiah serta sumber-sumber lain mengenai permasalahan yang berhubungan dengan penulisan.

3.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam pembuatan sistem ini adalah prototipe yang terdiri dari 5 (lima) fase tahapan yaitu *communication*, *quick plan*, *modelling quick design*, *construction of prototype*, dan *deployment, delivery and feedback*.

3.2.1 Communication

Pada tahap ini, antara Kejaksaan Negeri Lampung Timur dengan peneliti berkomunikasi mengenai spesifikasi kebutuhan yang diinginkan. Peneliti melakukan pengumpulan data yang akan digunakan dalam pembuatan sistem yang dibutuhkan. Metode pengumpulan data yang digunakan guna mendapatkan data

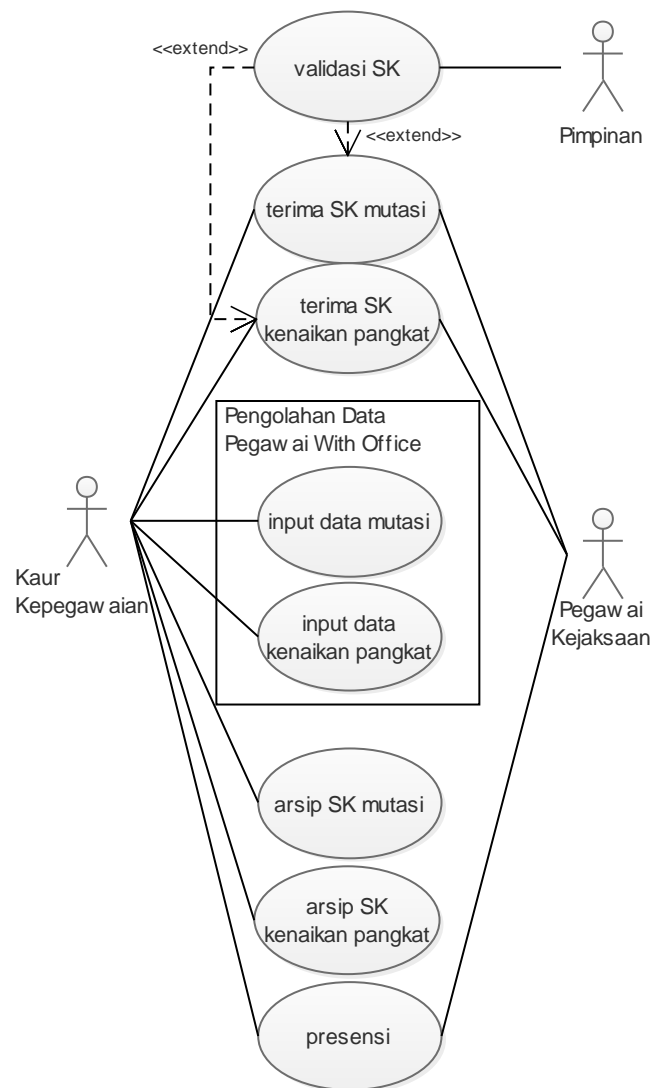
yang dibutuhkan adalah terdiri dari observasi, wawancara dan studi pustaka seperti yang telah dijelaskan di atas. Data yang diperoleh dari pengumpulan data dialisis. Adapun analisis yang didapat berupa prosedur sistem yang sedang berjalan saat ini mengenai presensi pegawai, dan pengolahan data mutasi serta kenaikan pangkat pegawai. Dari analisis sistem yang sedang berjalan saat ini, didapatlah analisis suatu permasalahan.

3.2.1.1 Analisis Sistem Berjalan

Analisis prosedur sistem yang sedang berjalan saat ini digambarkan dengan *use case diagram* seperti pada Gambar 3.1. Data mutasi dan kenaikan pangkat pegawai diolah oleh Kaur Kepegawaian menggunakan aplikasi *Office*, sedangkan presensi seluruh pegawai Kejaksaan dilakukan dengan cara mengisi buku presensi.

Pimpinan Kejaksaan Negeri Lampung Timur melakukan validasi SK mutasi dan SK kenaikan pangkat pegawai yang diperoleh dari Kejaksaan Agung. Kemudian SK yang sudah divalidasi tersebut diberikan kepada Kaur Kepegawaian untuk direkap datanya di dalam komputer menggunakan aplikasi Microsoft Excel. Setelah dilakukan perekapan data terhadap SK tersebut, maka Kaur Kepegawaian mengarsipkan SK tersebut. Salinan SK kenaikan pangkat dan mutasi pegawai yang telah divalidasi kemudian diberikan kepada pegawai yang bersangkutan.

Di dalam analisa sistem yang berjalan mengenai presensi pegawai dilakukan dengan cara pegawai menulis dan menandatangani buku presensi pegawai.



Gambar 3.1 Use Case Diagram Prosedur Sedang Berjalan Saat Ini

3.2.1.2 Analisis Permasalahan

Permasalahan yang terjadi pada sistem yang sedang berjalan saat ini adalah sebagai berikut :

- Data mutasi dan kenaikan pangkat pegawai Kejaksaan yang diolah menggunakan aplikasi *Office* tidak memiliki basis data sehingga tidak terdapat *back up* data. Jika terjadi kesalahan seperti “*human error*” (terjadi penghapusan data), maka data tersebut suli dikembalikan seperti semula.
- Presensi pegawai dilakukan dengan cara mengisi buku presensi setiap hari pada jam masuk kerja. Proses tersebut dapat menimbulkan kecurangan pada pengisian data presensi pegawai jika ada pegawai yang “menitip” presensi

(pemalsuan tanda tangan jika pegawai tersebut tidak hadir) kepada pegawai lain.

3.2.2 Quick Plan

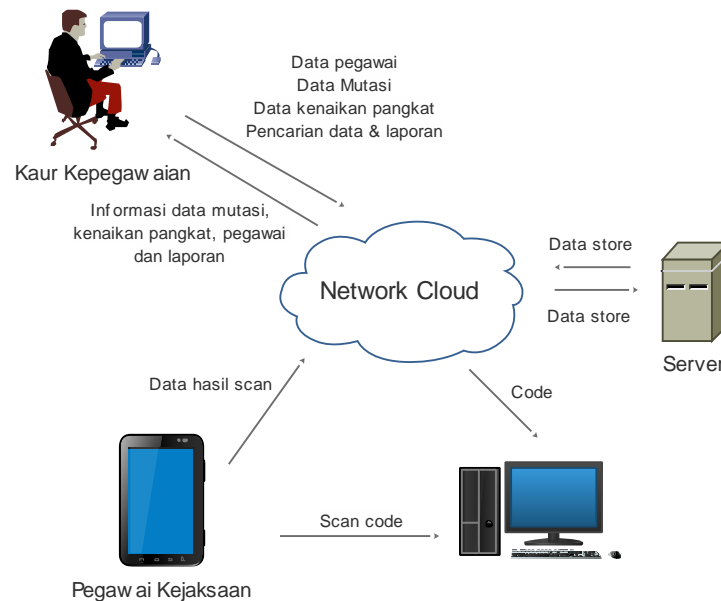
Dari analisis permasalahan yang telah diuraikan sebelumnya, maka untuk mengurangi permasalahan yang terjadi pada Kejaksaan Negeri Lampung Timur pada penelitian ini peneliti mengusulkan suatu sistem informasi kepegawaian yang di dalamnya terdapat presensi pegawai menggunakan QR Code berbasis Android dan pengolahan data mutasi serta kenaikan pangkat pegawai berbasis *Desktop*.

3.2.3 Modelling Quick Design

Setelah didapat ide mengenai sistem yang akan dibuat, maka pada tahap ini peneliti membuat perancangan/pemodelan sistem. Adapun perancangan sistem diusulkan terdiri dari perancangan arsitektur sistem, *use case diagram*, *activity diagram*, *class diagram* dan *interface* sistem. Perancangan ini dibuat menggunakan aplikasi *Edraw Max*.

3.2.3.1 Arsitektur Sistem

Perancangan arsitektur Sistem Informasi Kepegawaian pada Kejaksaan Negeri Lampung Timur yang diusulkan adalah seperti pada Gambar 3.2.



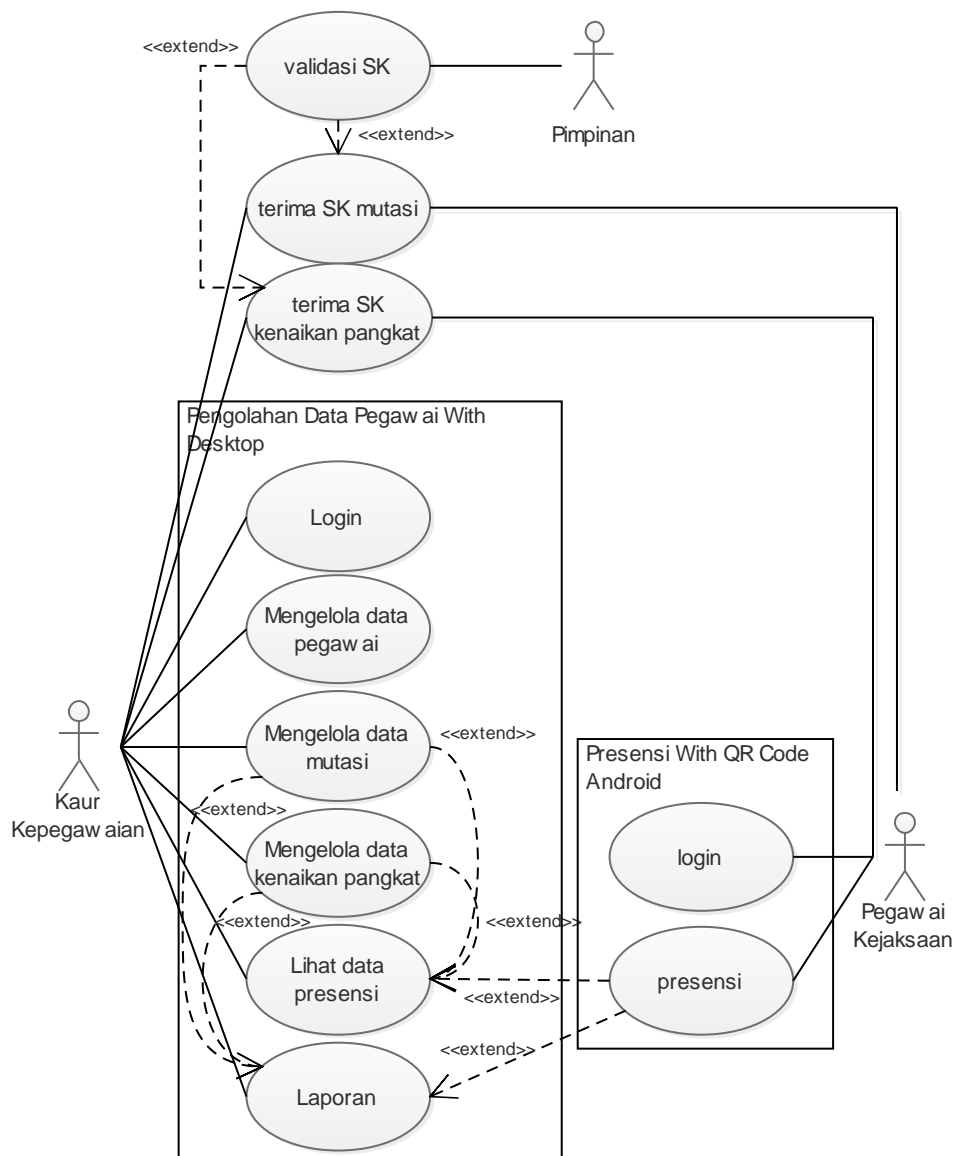
Gambar 3.2 Perancangan Arsitektur Sistem Diusulkan

Keterangan :

- Kaur Kepegawaian selaku admin memasukkan data pegawai, data mutasi dan kenaikan pangkat ke dalam sistem dan data tersebut tersimpan ke dalam basis data. Ketika Kaur Kepegawaian membutuhkan data-data tersebut dan melakukan pencarian, maka data yang dibutuhkan akan ditampilkan. Ketika Kaur Kepegawaian membutuhkan laporan, maka informasi data laporan tersebut akan tampil sesuai dengan yang dibutuhkan.
- Presensi pegawai dilakukan dengan cara men-*scan* code yang telah tampil di dalam komputer. Pegawai Kejaksanaan men-*scan* code yang telah tampil di dalam komputer menggunakan *smartphone* Android. Data hasil *scan* tersebut tersimpan di dalam *database*.

3.2.3.2 Use Case Diagram

Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Perancangan *use case diagram* Sistem Informasi Kepegawaian pada Kejaksanaan Negeri Lampung Timur yang diusulkan adalah seperti pada Gambar 3.3.



Gambar 3.3 Perancangan *Use Case* Diagram Sistem Diusulkan

Pimpinan Kejaksaan Negeri Lampung Timur melakukan validasi SK mutasi dan SK kenaikan pangkat pegawai yang diperoleh dari Kejaksaan Agung. Kemudian SK yang sudah divalidasi tersebut diberikan kepada Kaur Kepegawaian untuk direkap datanya di dalam sistem komputer yang terkoneksi dengan basis data. Sebelumnya Kaur Kepegawaian telah memasukkan data pegawai ke dalam sistem sehingga memudahkan dalam mengolah data SK mutasi dan kenaikan pangkat pegawai. Sistem juga dapat menampilkan laporan data presensi beserta data kenaikan pangkat dan mutasi pegawai. Setelah dilakukan perekapan data terhadap

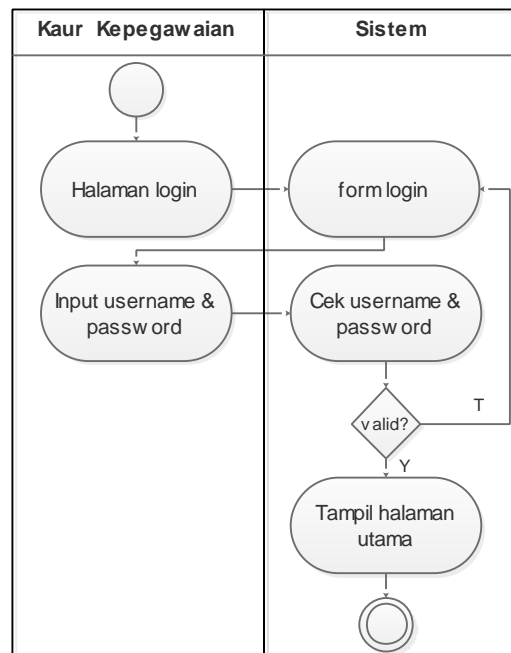
SK tersebut, maka Kaur Kepegawaian mengarsipkan SK tersebut. Salinan SK kenaikan pangkat dan mutasi pegawai yang telah divalidasi kemudian diberikan kepada pegawai yang bersangkutan. Untuk presensi pegawai dirancang terkomputerisasi antara *smartphone* Android dengan komputer menggunakan QR Code yang di dalam QR tersebut berisikan data masing-masing pegawai.

3.2.3.3 Activity Diagram

Activity diagram menggambarkan aliran kerja atau aktivitas sistem yang dibangun. Perancangan *activity diagram* Sistem Informasi Kepegawaian pada Kejaksaan Negeri Lampung Timur yang diusulkan adalah sebagai berikut :

a. Activity Diagram Login Kaur Kepegawaian (Desktop)

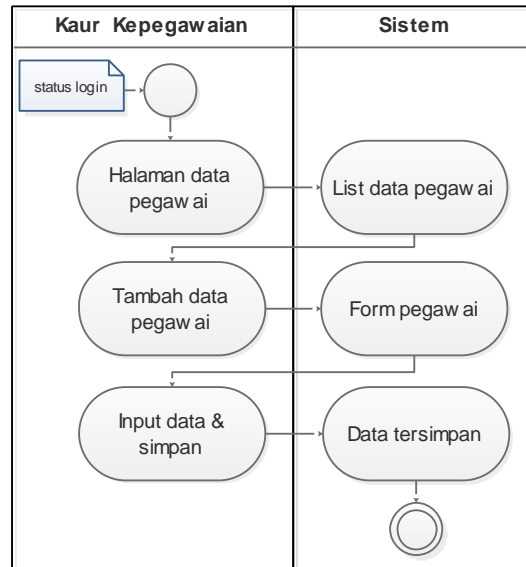
Activity diagram login menggambarkan aktivitas Kaur Kepegawaian untuk mengakses penuh sistem dengan memasukkan *username* dan *password*. Perancangan *activity diagram login* Kaur Kepegawaian adalah seperti pada Gambar 3.4.



Gambar 3.4 Perancangan *Activity Diagram Login* Kaur Kepegawaian (Desktop)

b. *Activity Diagram* Mengelola Data Pegawai

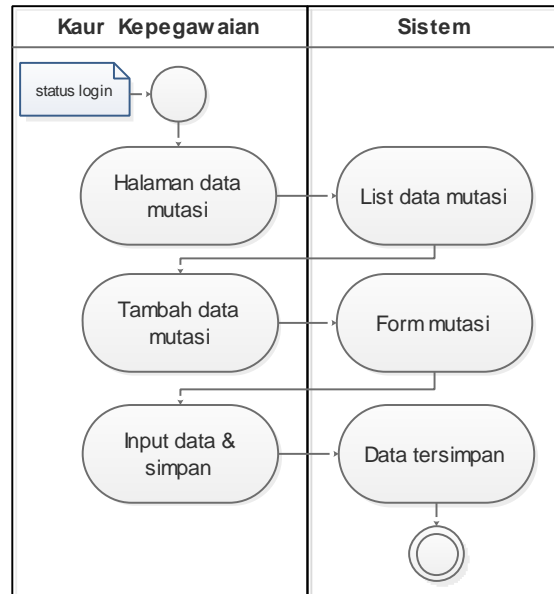
Activity diagram mengelola data pegawai menggambarkan aktivitas Kaur Kepegawaian dalam mengelola data pegawai ke dalam sistem. Perancangan *activity diagram* mengelola data pegawai adalah seperti pada Gambar 3.5.



Gambar 3.5 Perancangan *Activity Diagram* Mengelola Data Pegawai

c. *Activity Diagram* Mengelola Data Mutasi

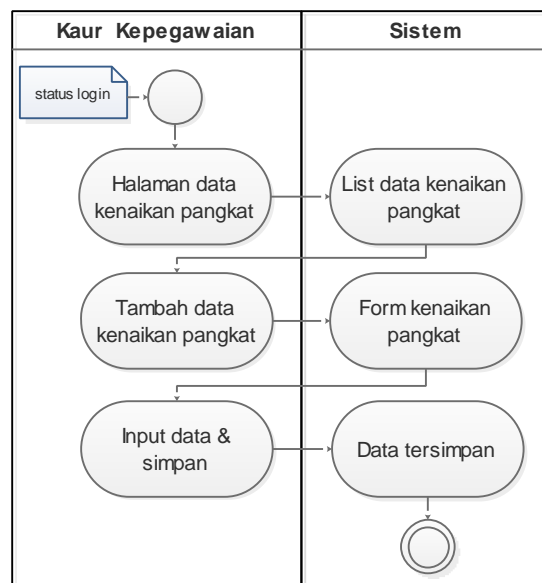
Activity diagram mengelola data mutasi menggambarkan aktivitas Kaur Kepegawaian dalam mengelola data mutasi pegawai ke dalam sistem. Perancangan *activity diagram* mengelola data mutasi adalah seperti pada Gambar 3.6.



Gambar 3.6 Perancangan *Activity Diagram* Mengelola Data Mutasi

d. *Activity Diagram* Mengelola Data Kenaikan Pangkat

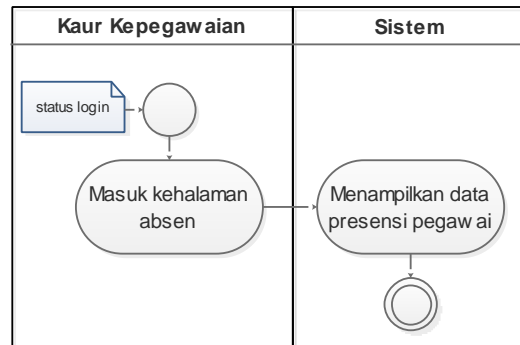
Activity diagram input data kenaikan pangkat menggambarkan aktivitas Kaur Kepegawaian dalam memasukkan data kenaikan pangkat pegawai ke dalam sistem. Perancangan *activity diagram* mengelola data kenaikan pangkat adalah seperti pada Gambar 3.7.



Gambar 3.7 Perancangan *Activity Diagram* Mengelola Data Kenaikan Pangkat

e. *Activity Diagram* Lihat Data Presensi

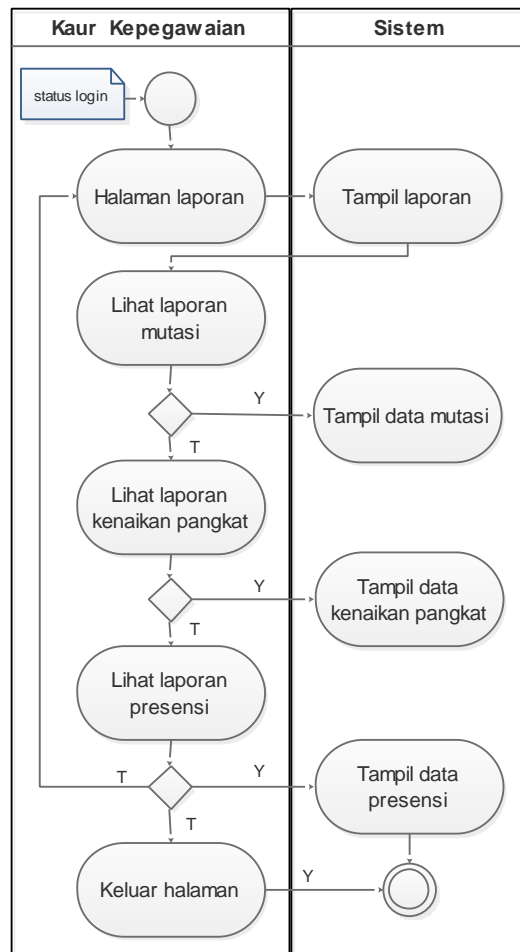
Activity diagram lihat data presensi menggambarkan aktivitas Kaur Kepegawaian melihat data presensi pegawai di dalam sistem. Perancangan *activity diagram* lihat data presensi adalah seperti pada Gambar 3.8.



Gambar 3.8 Perancangan *Activity Diagram* Lihat Data Presensi

f. *Activity Diagram* Laporan

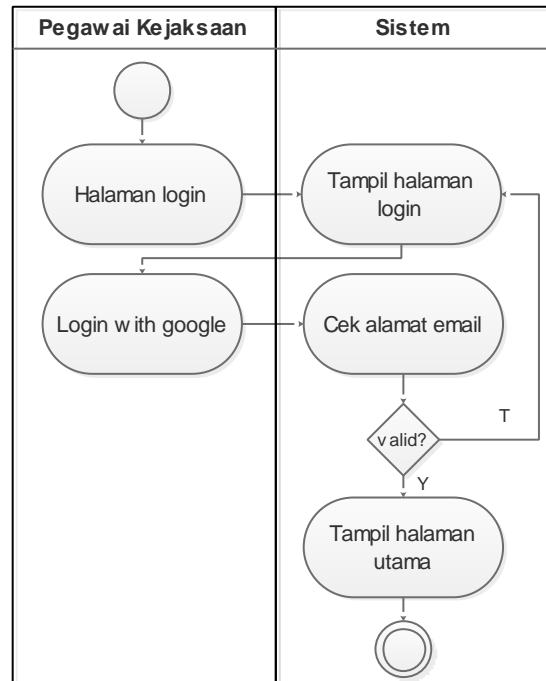
Activity diagram laporan menggambarkan aktivitas Kaur Kepegawaian dalam melihat laporan data mutasi, kenaikan pangkat pegawai dan presensi pegawai. Perancangan *activity diagram* laporan data adalah seperti pada Gambar 3.9.



Gambar 3.9 Perancangan Activity Diagram Laporan

g. Activity Diagram Login Pegawai Kejaksaan (Android)

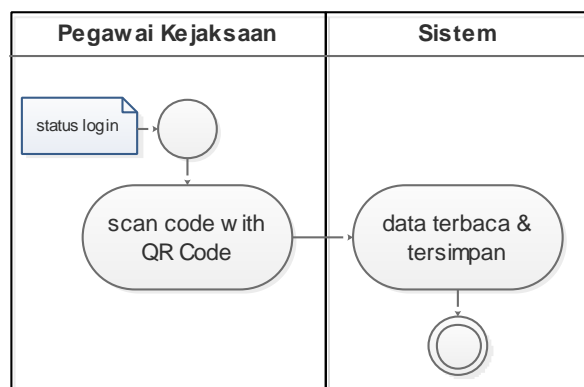
Activity diagram login menggambarkan aktivitas Pegawai Kejaksaan untuk mengakses penuh sistem dengan cara login berdasarkan akun gmail (alamat email) yang terdaftar di handphone pengguna. Perancangan activity diagram login Pegawai Kejaksaan adalah seperti pada Gambar 3.10.



Gambar 3.10 Perancangan *Activity Diagram Login* Pegawai Kejaksaan (Android)

h. *Activity Diagram* Presensi

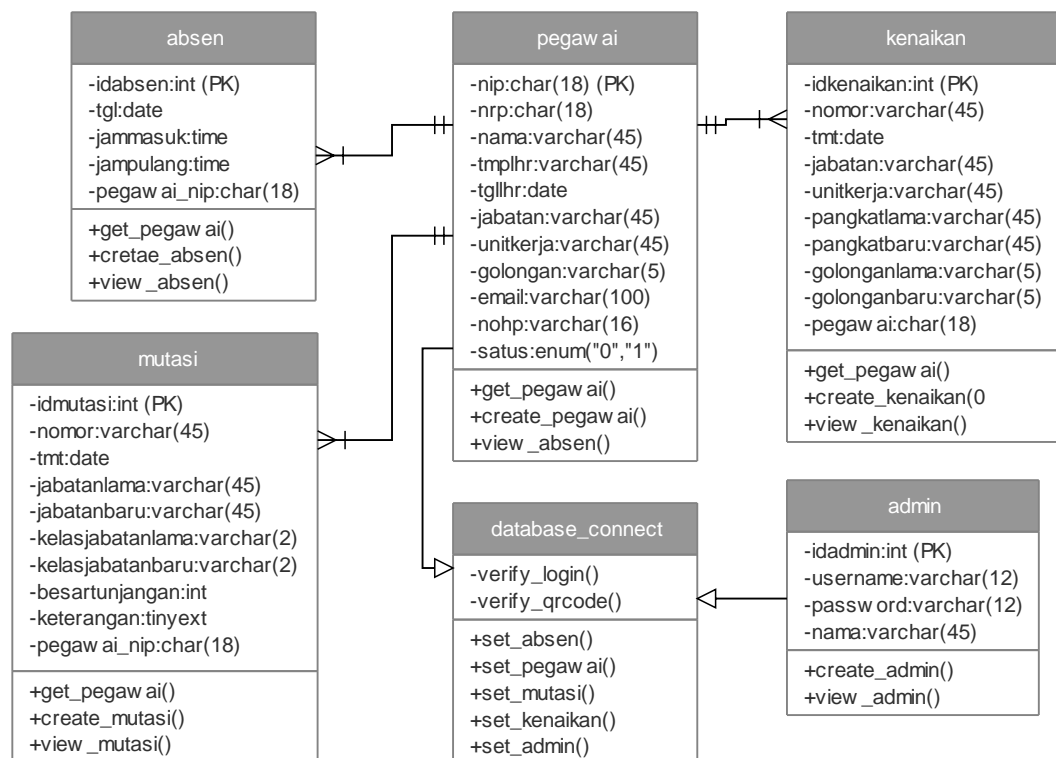
Activity diagram presensi menggambarkan aktivitas Pegawai Kejaksaan (keseluruhan) dalam melakukan presensi dengan cara *scan code* menggunakan *smartphone* Android (milik pribadi). Perancangan *activity diagram* presensi adalah seperti pada Gambar 3.11



Gambar 3.11 Perancangan *Activity Diagram* Presensi

3.2.3.4 Class Diagram

Perancangan *class diagram* menggambarkan aktivitas sistem dalam penyimpanan data berdasarkan kelas-kelas data yang disinpan yang terkoneksi satu sama lain, sehingga dalam pemrosesan data yang dilakukan oleh sistem dapat dilakukan dengan baik (terintegrasi). Adapun perancangan class diagram pada sistem yang diusulkan adalah seperti pada Gambar 3.12.



Gambar 3.12 Perancangan *Class Diagram* Sistem Diusulkan

Keterangan :

PK = *Primary Key*

3.2.3.5 Interface Sistem

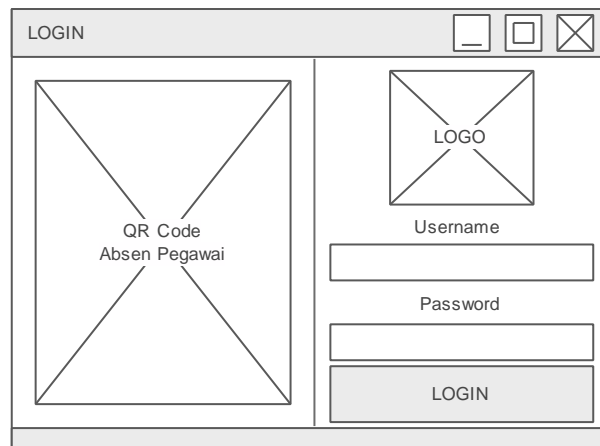
Interface sistem diusulkan terdiri dari 2 (dua) jenis, yaitu *interface* sistem berbasis *desktop* dengan hak akses Kaur Kepegawaian, dan *interface* berbasis Android yang diakses oleh Pegawai Kejaksaan.

3.2.3.5.1 *Interface Sistem Berbasis Desktop*

Perancangan *interface* sistem yang diusulkan berbasis *desktop* ini digunakan oleh Kaur Kepegawaian dalam pengolahan data pegawai, mutasi, kenaikan pangkat, melihat data presensi pegawai, dan laporan. Sistem ini dibuat dengan bahasa pemrograman Java menggunakan aplikasi IntelliJIDEA. Adapun perancangan *interface* dari sistem ini adalah sebagai berikut :

a. *Interface Login* Kaur Kepegawaian

Interface ini dirancang untuk *login* Kaur Kepegawaian. Perancangan *interface login* Kaur Kepegawaian adalah seperti pada Gambar 3.13.



Gambar 3.13 Perancangan *Interface Login* Kaur Kepegawaian

b. *Interface* Halaman Utama Kaur Kepegawaian

Interface halaman utama dirancang berisi menu-menu pendukung yang dapat digunakan oleh Kaur kepegawaian dalam mengolah data mutasi, data pegawai, data admin dan data kenaikan pangkat. Adapun perancangan *interface* halaman utama Kaur Kepegawaian adalah seperti pada Gambar 3.14.

Gambar 3.14 Perancangan *Interface* Halaman Utama Kaur Kepegawaian

c. *Interface* Halaman Tambah Pegawai

Interface ini dirancang dan dipergunakan oleh Kaur Kepegawaian untuk dapat menambah data pegawai ke dalam sistem. Perancangan *interface* halaman tambah pegawai adalah seperti pada Gambar 3.15.

Gambar 3.15 Perancangan *Interface* Halaman Tambah Pegawai

d. *Interface* Halaman Mutasi

Interface ini dirancang dan dipergunakan oleh Kaur Kepegawaian untuk dapat mengolah (*input*, edit, hapus, *download*) data mutasi pegawai ke dalam sistem.

Perancangan *interface* halaman mutasi pegawai adalah seperti pada Gambar 3.16.

The image shows a web application window titled "MUTASI". The window has a standard title bar with minimize, maximize, and close buttons. The main content area is divided into two columns of input fields. The left column contains fields for NIP, NAMA, NRP, JABATAN, and UNIT KERJA. The right column contains fields for NOMOR, TMT, JABATAN LAMA, JABATAN BARU, KELAS JABATAN LAMA, KELAS JABATAN, BESAR TUNJANGAN, and KETERANGAN. Below the input fields, there is a table with two columns: "NOMOR" and "TMT". The table is currently empty. To the right of the table, there is a "DOWNLOAD SK" button. At the bottom of the window, there are three buttons: "TAMBAH", "EDIT", and "HAPUS".

Gambar 3.16 Perancangan *Interface* Halaman Mutasi Pegawai

Jika akan menambah data mutasi, maka tekan tombol “tambah” dan sistem akan menampilkan halaman tambah data mutasi seperti pada Gambar 3.17.

The image shows a web form titled "MUTASI" (Transfer) with the following fields and controls:

- NIP**: Input field
- NAMA**: Input field
- NRP**: Input field
- JABATAN**: Input field
- UNIT KERJA**: Input field
- NOMOR**: Input field
- TMT**: Input field with a calendar icon
- JABATAN LAMA**: Input field
- JABATAN BARU**: Input field
- KELAS JABATAN LAMA**: Input field
- KELAS JABATAN BARU**: Input field
- BESAR TUNJANGAN**: Input field
- KETERANGAN**: Large text area for notes
- SK**: Input field with a search icon
- Buttons**: SIMPAN (Save), ULANGI (Repeat), BATAL (Cancel)

Gambar 3.17 Perancangan *Interface* Tambah Data Mutasi Pegawai

e. *Interface* Halaman Kenaikan Pangkat

Interface ini dirancang dan dipergunakan oleh Kaur Kepegawaian untuk dapat mengolah (*input*, edit, hapus, *download*) data kenaikan pangkat pegawai ke dalam sistem. Perancangan *interface* halaman kenaikan pangkat pegawai adalah seperti pada Gambar 3.18.

KENAIKAN

NIP

NAMA

NRP

JABATAN

UNIT KERJA

NOMOR

TMT

JABATAN

UNIT KERJA

PANGKAT LAMA

PANGKAT BARU

GOLONGAN LAMA

GOLONGAN BARU

NOMOR	TMT

Gambar 3.18 Perancangan *Interface* Halaman Kenaikan Pangkat Pegawai

Jika akan menambah data kenaikan pangkat, maka tekan tombol “tambah” dan sistem akan menampilkan halaman tambah data kenaikan pangkat seperti pada Gambar 3.19.

KENAIKAN

NIP

NAMA

NRP

JABATAN

UNIT KERJA

NOMOR

TMT

JABATAN

UNIT KERJA

PANGKAT LAMA PANGKAT BARU

GOLONGAN LAMA GOLONGAN

SK

Gambar 3.19 Perancangan *Interface* Halaman Tambah Data Kenaikan Pangkat Pegawai

f. *Interface* Halaman Absen

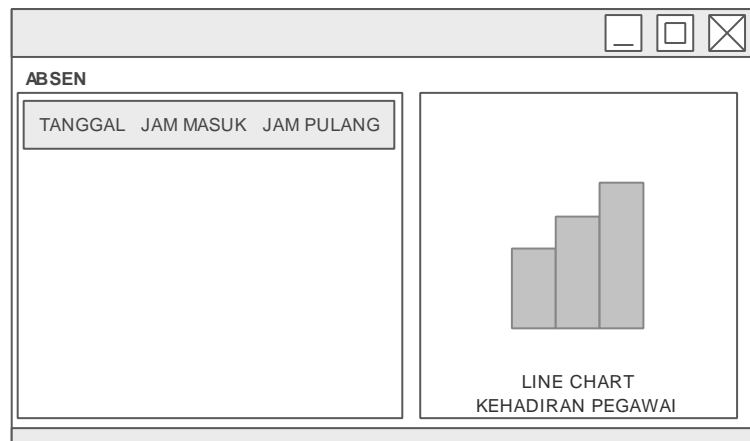
Halaman absen dirancang menjadi 2 (dua) bagian, yaitu halaman absen yang menampilkan data presensi pegawai seperti pada Gambar 3.20, dan halaman absen yang menampilkan kurva grafik presensi pegawai seperti pada Gambar 3.21.

Kepegawaian Kejaksaan

ABSEN

NIP	Nama	Tanggal	Jam Masuk	Jam Pulang
-----	------	---------	-----------	------------

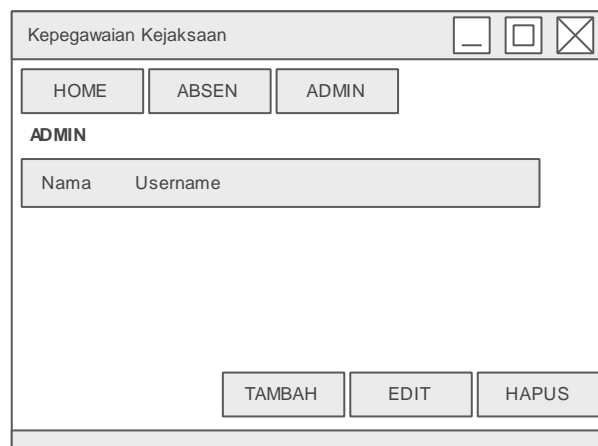
Gambar 3.20 Perancangan *Interface* Lihat Presensi Pegawai



Gambar 3.21 Perancangan *Interface* Kurva Grafik Presensi Pegawai

g. *Interface* Halaman Admin

Interface halaman admin dirancang untuk digunakan oleh Kaur Kepegawaian dalam mengolah data admin. Perancangan *interface* halaman admin adalah seperti pada Gambar 3.22.



Gambar 3.22 Perancangan *Interface* Halaman Admin

Jika akan menambah data admin, maka tekan tombol “tambah” dan sistem akan menampilkan halaman tambah data admin seperti pada Gambar 3.23.

Gambar 3.23 Perancangan *Interface* Halaman Tambah Data admin

3.2.3.5.2 *Interface* Sistem Berbasis Android

Perancangan *interface* sistem yang diusulkan berbasis Android ini digunakan oleh seluruh pegawai Kejaksaan Negeri Lampung Timur dalam melakukan presensi pegawai menggunakan *QR Code*. Adapun perancangan *interface* presensi pegawai adalah sebagai berikut :

a. *Interface* Halaman *Login* Pegawai

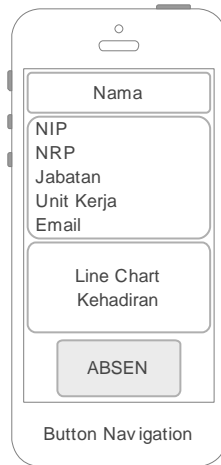
Interface halaman *login* dirancang berbasiskan *email Google* pegawai yang digunakan di dalam *smartphone Android*. Perancangan halaman *login* pegawai berbasiskan Android adalah seperti pada Gambar 3.24.



Gambar 3.24 Perancangan *Interface Login* Pegawai Berbasis Android

b. *Interface* Halaman Utama Presensi

Interface halaman utama presensi pegawai setelah berhasil *login* adalah seperti pada Gambar 3.25.



Gambar 3.25 Perancangan *Interface* Halaman Utama Prsesensi Berbasis Android

c. Interface Halaman Presensi QR Code

Interface presensi QR Code dirancang sebagai media pembaca code QR yang tampil pada sistem presensi yang ada pada desktop. Perancangan interface presensi QR Code adalah seperti pada Gambar 3.26.



Gambar 3.26 Perancangan *Interface* Halaman Presensi QR Code

3.2.4 Construction Of Prototype

Setelah tahap pemodelan, maka peneliti mulai melakukan pengkodean program. Bahasa pemograman yang digunakan dalam pembuatan sistem presensi adalah *Java* menggunakan aplikasi Android Studio dengan *database* MySQL

menggunakan *tools* PHPMYAdmin. Pengolahan data mutasi dan kenaikan pangkat pegawai dibuat berbasis *desktop* dengan menggunakan aplikasi IntelliJ IDEA.

3.2.5 Deployment, Delivery and Feedback

Tahap pembuatan program dibarengi oleh tahapan implementasi dan pengujian sistem. Sistem yang baru diberikan kepada Kejaksaan Negeri Lampung Timur. Jika ada kekurangan atau penambahan kebutuhan sistem, maka Kejaksaan Negeri Lampung Timur akan mengkomunikasikan kembali dengan peneliti.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4.1 Alat Pendukung Pembuatan Sistem

Alat pendukung pembuatan Sistem Informasi Kepegawaian pada Kejaksaan Negeri Lampung Timur terdiri dari beberapa perangkat lunak dan perangkat keras. Adapun alat pendukung pembuatan sistem yaitu sebagai berikut :

4.1.1 Perangkat Lunak (*Software*)

Perangkat lunak yang dibutuhkan Sistem Informasi Kepegawaian pada Kejaksaan Negeri Lampung Timur adalah sebagai berikut :

- a. Sistem Operasi : *Windows* 10 64bit dan *Android* Versi 5.0 (*Lollipop*).
- b. *Database* : *MySQL*.
- c. *Tools* : *Android Studio*, *IntelliJ IDEA*, *Edraw Max*, *Xampp* (*MySQL*), *Ms. Word* dan *Notepad*.

4.1.2 Perangkat Keras (*Hardware*)

Perangkat keras dengan rekomendasi minimum yang digunakan dalam pembuatan Sistem Informasi Kepegawaian pada Kejaksaan Negeri Lampung Timur adalah sebagai berikut :

- a. Komputer atau laptop, dengan spesifikasi minimal :
 1. *Processor Core i3*.
 2. *Hardisk 500 GB*.
 3. *RAM 4 GB*.
- b. *Smartphone* *Android*.

4.2 Implementasi Sistem

Implementasi dari sistem yang telah dirancang sebelumnya adalah terdiri dari 2 (dua) hak akses sistem, yaitu implementasi sistem berbasis *desktop* dengan hak akses Kaur Kepegawaian, dan implementasi berbasis *Android* yang diakses

oleh Pegawai Kejaksaan. Adapun penjelasan implementasi sistem tiap akses adalah sebagai berikut.

4.2.1 Implementasi Sistem (*Desktop*) Akses Kaur Kepegawaian

Implementasi dari sistem yang diusulkan berbasis *desktop* ini digunakan oleh Kaur Kepegawaian dalam pengolahan data pegawai, mutasi, kenaikan pangkat, melihat data presensi pegawai, dan laporan. Sistem ini dibuat dengan bahasa pemrograman Java menggunakan aplikasi IntelliJ IDEA. Adapun implementasi dari sistem ini adalah sebagai berikut :

a. Implementasi Halaman *Login* Kaur Kepegawaian

Halaman ini digunakan untuk *login* Kaur Kepegawaian beserta akses absensi pegawai jaksa lainnya termasuk Kaur Kepegawaian menggunakan QR code. Adapun implementasi sistem halaman *login* Kaur kepegawaian adalah seperti ppada Gambar 4.1.



Gambar 4.1 Implementasi Halaman *Login* Kaur Kepegawaian

b. Implementasi Halaman Input Pegawai

Implementasi ini dipergunakan oleh Kaur Kepegawaian untuk dapat menambah data pegawai ke dalam sistem. Implementasi sistem halaman tambah pegawai adalah seperti pada Gambar 4.2

The screenshot shows a web form titled 'PEGAWAI'. It includes the following fields and controls:

- NIP: A single-line text input field.
- NRP: A single-line text input field.
- NAMA: A single-line text input field.
- TEMPAT, TGL LAHIR: Two single-line text input fields, with a calendar icon next to the second field.
- JABATAN: A single-line text input field.
- UNIT KERJA: A single-line text input field.
- GOLONGAN: A single-line text input field.
- EMAIL: A single-line text input field.
- NO HP: A single-line text input field.
- STATUS: Two radio buttons labeled 'Aktif' (selected) and 'Tidak Aktif'.
- Buttons at the bottom: 'SIMPAN' (Save), 'ULANGI' (Refresh), and 'BATAL' (Cancel).

Gambar4.2 Implementasi Halaman Input Pegawai Akses Kaur Kepegawaian

c. Implementasi Proses Halaman Utama

Implementasi halaman utama berisi menu-menu pendukung yang dapat digunakan oleh Kaur kepegawaian dalam mengolah data mutasi, data pegawai, data admin dan data kenaikan pangkat seperti gambar4.3

The screenshot shows the main dashboard of the 'DATA KEPEGAWAIAN KEJAKSAAN NEGERI LAMPUNG TIMUR'. It includes the following elements:

- Navigation bar: 'HOME', 'ABSEN', and 'ADMIN' buttons.
- Search bar: A search input field with a magnifying glass icon.
- Table: A table with columns: NIP, NRP, NAMA, TEMPAT LAHIR, TANGGAL LAHIR, JABATAN, and UNIT KERJA. The table is currently empty, displaying 'No content in table'.
- Secondary navigation bar: 'ABSEN', 'KENAIKAN', 'MUTASI', 'PEGAWAI', 'EDIT', and 'HAPUS' buttons.

Gambar 4.3 Implementasi Halaman Utama Kaur Kepegawaian Akses Kaur Kepegawaian

d. Implementasi Output Data Kepegawaian

Implementasi ini hasil output dari data kepegawaian ke dalam sistem adalah seperti pada Gambar 4.4

NIP	NRP	NAMA	TEMPAT LAHIR	TANGGAL LAHIR	JABATAN	UNIT KERJA
196303091984092001	3846312	Lailatunnur	Sukadana	March 3, 1963	Muda wira tu	Kejaksaan Negeri Lampung T
197604022002121001	40376298	Damiri	Jepara	April 2, 1976	Muda Wira	Kejaksaan Negeri Lampung T
198097640186517865	60880171	Neli putri	Bandar lampung	September 24, 2009	Jaksa utama	Kejaksaan Negeri Lampung T
198112312002121010	89789769	Adiostama	Jakarta	September 11, 2019	Jaksa muda	Kejaksaan Negeri Lampung T
198306202009122001	61083422	Eva Marwathy	jakarta	September 2, 2019	Jaksa pratama	Kejaksaan Negeri Lampung T
198307302008122003	609837378	Dede Tri Anggraini	Bandar Lampung	July 30, 1983	Jaksa Pratama	Kejaksaan Negeri Lampung T
19835679101019189	23678998	fatimah	metro	September 1, 2019	kepala jaksa	kejaksanaan negeri lampung ti
198411022002122001	40384029	Nopiyana	Metro	November 2, 1984	Madya wira	Kejaksaan Negeri Lampung T
198809252014031002	6148885	Ab bayu purwo	Manggelang	September 25, 1988	Ajun Jaksa	Kejaksaan Negeri Lampung T
199208212015022001	61592137	Laura Desvia	Bandar Lampung	December 30, 1996	Kepala Jaksa	Kejaksaan negeri Lampung T
199208252001022001	61789778	Afina putri	Jakarta	August 25, 1992	Ajun jaksa	Kejaksaan Negeri Lampung T
199212232001122001	6789064	Putra utama	jakarta	September 11, 2019	ajun jaksa	Kejaksaan Negeri Lampung T

Gambar 4.4 Implementasi Sistem Halaman Output Data Pegawai Akses Kaur Kepegawaian

e. Implementasi Input data mutasi

Implementasi input data mutasi akan menambah data mutasi, maka sistem menampilkan halaman input data mutasi seperti pada gambar 4.5

NOMOR

TMT

JABATAN LAMA JABATAN BARU

KELAS JABATAN LAMA KELAS JABATAN BARU

BESAR TUNJANGAN

KETERANGAN

SK

Gambar 4.5 Implementasi Halaman Input Data Mutasi Akses Kaur Kepegawaian

f. Implementasi Proses Halaman Mutasi

Halaman ini dipergunakan oleh Kaur Kepegawaian untuk dapat mengolah (*input*, edit, hapus, *download*) data mutasi pegawai ke dalam sistem. Implementasi halaman mutasi pegawai adalah seperti pada Gambar 4.6.

Gambar 4.6 Implementasi Halaman Proses Mutasi Pegawai Akses Kaur Kepegawaian

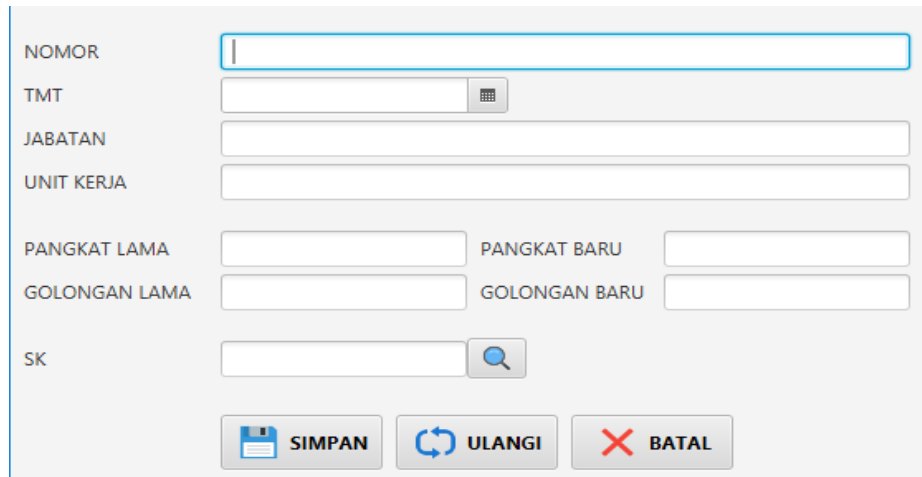
g. Implementasi output mutasi

Implementasi ini hasil output dari data mutasi ke dalam sistem. Implementasi sistem halaman output mutasi adalah seperti pada Gambar 4.7

Gambar 4.7 Implementasi Halaman output Mutasi Pegawai Akses Kaur Kepegawaian

h. Implementasi input data kenaikan pangkat

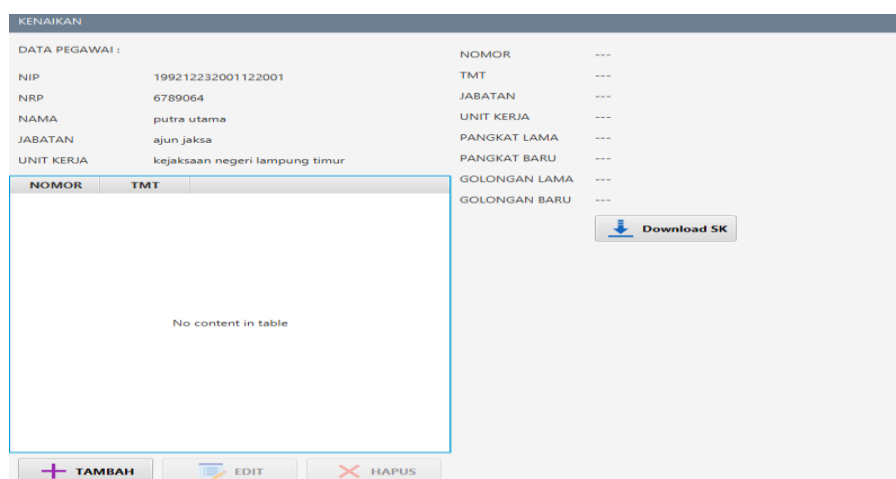
Implementasi input data mutasi akan menambah data mutasi, maka sistem menampilkan halaman input data mutasi seperti pada gambar 4.8



Gambar 4.8 Implementasi Halaman Input Data Kenaikan Pangkat Akses Kaur Kepegawaian

i. Implementasi Proses Halaman Kenaikan Pangkat

Halaman ini dipergunakan oleh Kaur Kepegawaian untuk dapat mengolah (*input, edit, hapus, download*) data kenaikan pangkat pegawai ke dalam sistem. Implementasi sistem halaman kenaikan pangkat pegawai adalah seperti pada Gambar 4.9



Gambar 4.9 Implementasi Halaman Proses Kenaikan Pangkat Pegawai Akses Kaur Kepegawaian

j. Implementasi Output Data Kenaikan Pangkat

Implementasi ini hasil output dari data kenaikan pangkat ke dalam sistem. Implementasi sistem halaman output kenaikan pangkat adalah seperti pada Gambar 4.10

KENAIKAN																			
DATA PEGAWAI :																			
NIP	199212232001122001	NOMOR	1155																
NRP	6789064	TMT	October 1, 2018																
NAMA	Putra utama	JABATAN	jaksa madya																
JABATAN	ajun jaksa	UNIT KERJA	kejaksaan negeri lampung timur																
UNIT KERJA	Kejaksaan Negeri Lampung Timur	PANGKAT LAMA	ajun jaksa																
		PANGKAT BARU	penata muda																
		GOLONGAN LAMA	III/a																
		GOLONGAN BARU	III/b																
<table border="1"> <thead> <tr> <th>NOMOR</th> <th>TMT</th> </tr> </thead> <tbody> <tr> <td>1155</td> <td>October 1, ...</td> </tr> <tr> <td>1667</td> <td>October 7, ...</td> </tr> <tr> <td>1166</td> <td>August 5, 2...</td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </tbody> </table>			NOMOR	TMT	1155	October 1, ...	1667	October 7, ...	1166	August 5, 2...									Download SK
NOMOR	TMT																		
1155	October 1, ...																		
1667	October 7, ...																		
1166	August 5, 2...																		

Gambar 4.10 Implementasi Halaman Proses Output Kenaikan Pangkat Pegawai Akses Kaur Kepegawaian

k. Implementasi Halaman Absen

Halaman absen dibagi menjadi 2 (dua) bagian, yaitu halaman absen yang menampilkan halaman sebelum melakukan absensi pegawai seperti pada Gambar 4.11 dan halaman absen yang menampilkan jika pegawai sudah melakukan absensi pegawai seperti pada Gambar 4.12.

ABSEN				
NIP	NAMA	TANGGAL	JAM MASUK	JAM PULA...
No content in table				

Gambar 4.11 Implementasi Halaman sebelum absensi Pegawai Akses Kaur Kepegawaian

ABSEN				
NIP	NAMA	TANGGAL	JAM MASUK	JAM PULANG
19961252002172021	Laura desvia	October 20, 2019	01:15:15	
19961252002172021	Laura desvia	October 18, 2019		22:04:13

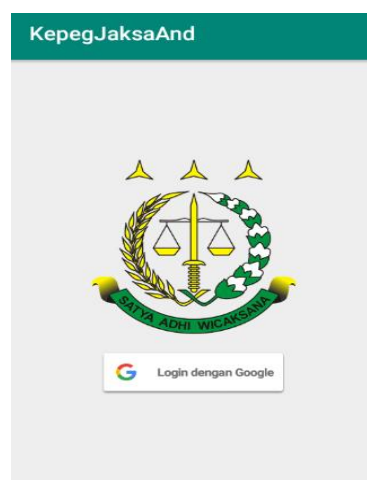
Gambar 4.12 Implementasi Halaman Lihat Presensi Pegawai Akses Kaur Kepegawaian

4.2.2 Implementasi Sistem (Android) Pegawai Kejaksaan

Implementasi sistem yang diusulkan berbasis Android ini digunakan oleh seluruh pegawai Kejaksaan Negeri Lampung Timur (termasuk Kaur Kepegawaian) dalam melakukan presensi pegawai menggunakan *QR Code*. Adapun implementasi sistem presensi pegawai adalah sebagai berikut :

a. Implementasi Halaman *Login* Pegawai

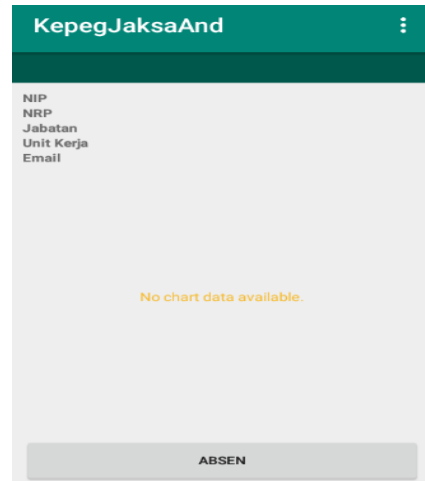
Halaman *login* pegawai berbasiskan *email Google* pegawai yang digunakan di dalam *smartphone Android*. Pegawai sebelumnya harus login dengan menggunakan email pegawai. Implementasi halaman *login* pegawai berbasiskan Android adalah seperti pada Gambar 4.13.



Gambar 4.13 Perancangan *Interface Login* Pegawai Berbasis Android

b. Implementasi Halaman Utama Presensi

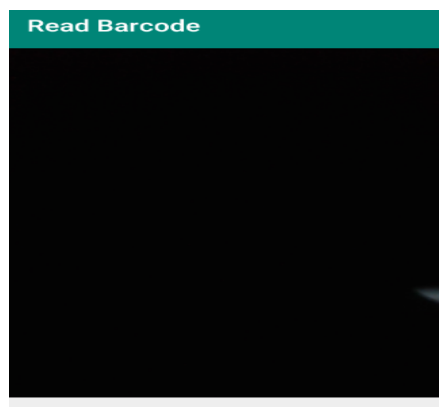
Implementasi halaman utama presensi pegawai setelah berhasil *login* adalah seperti pada Gambar 4.14. Di dalam halaman ini terdapat tombol absen yang digunakan untuk melakukan presensi kehadiran.



Gambar 4.14 Implementasi Halaman Utama Presensi Pegawai Berbasis Android

c. Implementasi Halaman Presensi QR Code

Halaman presensi QR Code digunakan sebagai media pembaca *code* QR yang tampil pada sistem presensi yang ada pada *desktop*. Implementasi presensi QR Code adalah seperti pada Gambar 4.15.



Gambar 4.15 Implementasi Halaman Presensi Pegawai Menggunakan QR Code

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Adapun kesimpulan dari Sistem Informasi Kepegawaian pada Kejaksaan Negeri Lampung Timur adalah sebagai berikut :

- a. Sistem yang dibangun dapat memudahkan presensi kehadiran Pegawai Kejaksaan dengan pemanfaatan *smartphone* Android yang didalamnya terdapat QR Code presensi.
- b. Dapat memudahkan Kaur Kepegawaian dalam melakukan pengolahan data mutasi dan kenaikan pangkat pegawai termasuk mengolah data laporannya.

5.2 Saran

Adapun saran yang diajukan untuk penelitian yang akan dilakukan selanjutnya adalah dengan menambahkan grafik data mutasi dan kenaikan pangkat pegawai.

DAFTAR PUSTAKA

A.S, Rosa dan Shalahuddin, M. 2018. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek* Edisi Revisi. Informatika, Bandung.

Avestro, Joyce. 2007. *Pengenalan Pemograman 1 : Java Education Network Indonesia (JENI)*.

Ermatita., dkk. 2017. *Pengembangan Sistem Absensi Menggunakan QR Code Reader Berbasis Android (Studi Kasus: Fakultas Ilmu Komputer Jurusan Sistem Informasi Unsri)*. Jurusan Sistem Informasi, Fakultas Ilmu Komputer Universitas Sriwijaya, Palembang.

Google Developer Training Team. 2016. Android Developer Fundamentals Course-Learn to Develop Android Application-Concept Reference by Google Developer Training Team.

Kadir, Abdul. 2014. *Pengenalan Sistem Informasi Edisi Revisi*. Andi Offset, Yogyakarta.

KOMINFO. *Indonesia Raksasa Teknologi Digital Asia*. Dikutip dari (https://kominfo.go.id/content/detail/6095/indonesia-raksasa-teknologi-digital-asia/0/sorotan_media), Tanggal 09 Desember 2018, Pukul 11.28 WIB.

Pengertian Kejaksaan. Diakses dari https://www.kejaksaan.go.id/profil_kejaksaan.php?id=1, Tanggal 01 November 2018, Pukul 06:17 WIB.

S, Roger Pressman. 2012. *Rekatyasa Perangkat Lunak*. Andi, Yogyakarta.

Solichin, Achmad. 2016. Pemograman Web dengan PHP dan MySQL. *E-Book*
diunduh dari
https://www.researchgate.net/publication/236885805_Pemrograman_Web_dengan_PHP_dan_MySQL.

Taqwa, M Nuddin dan Laily D Fithri. 2015. Sistem Absensi Asisten Dosen Menggunakan *QR Code Scanner* Berbasis Android Pada Program Studi Sistem Informasi Universitas Muria Kudus. Prosiding SNATIF, ISBN: 978-602-1180-21-1.

```

package dtc;

import mdl.AbsenMdl;
import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class AbsenDtc {
    private String partUrl;
    private String method;
    private JSONObject jsonObject;

    private Dtc dtc = new Dtc();

    private void setPartUrlMethodObject(String instruction, AbsenMdl mdl) {
        jsonObject = new JSONObject();

        switch (instruction) {
            case "add":
                partUrl = "absen";
                method = "POST";

                jsonObject.put("tgl", mdl.tgl);
                if (mdl.jammasuk != null && !mdl.jammasuk.equals(""))
                    jsonObject.put("jammasuk", mdl.jammasuk);
                if (mdl.jampulang != null && !mdl.jampulang.equals(""))
                    jsonObject.put("jampulang", mdl.jampulang);
                jsonObject.put("pegawai_nip", mdl.pegawai_nip);
                break;
            case "upd":
                partUrl = "absen/" + mdl.idabsen;
                method = "PUT";

                jsonObject.put("tgl", mdl.tgl);
                if (mdl.jammasuk != null && !mdl.jammasuk.equals(""))
                    jsonObject.put("jammasuk", mdl.jammasuk);
                if (mdl.jampulang != null && !mdl.jampulang.equals(""))
                    jsonObject.put("jampulang", mdl.jampulang);
                jsonObject.put("pegawai_nip", mdl.pegawai_nip);
                break;
            case "del":
                partUrl = "absen/" + mdl.idabsen;
                method = "DELETE";
                break;
            case "get":
                if (mdl == null) {
                    partUrl = "absen";
                    method = "GET";
                } else {
                    partUrl = "absen/some";
                    method = "POST";

                    jsonObject.put("pegawai_nip", mdl.pegawai_nip);
                }
                break;
        }
    }

    private AbsenMdl setMdl(JSONObject jo) {
        AbsenMdl mdl = new AbsenMdl();
        mdl.idabsen = Integer.toString(jo.getInt("idabsen"));
    }
}

```

```

        mdl.tgl = jo.getString("tgl");
        if (!jo.isNull("jammasuk")) mdl.jammasuk = jo.getString("jammasuk");
        if (!jo.isNull("jampulang")) mdl.jampulang = jo.getString("jampulang");
        mdl.pegawai_nip = jo.getString("pegawai_nip");
        mdl.nrp = jo.getString("nrp");
        mdl.nama = jo.getString("nama");

        return mdl;
    }

    public int aud(String instruction, AbsenMdl mdl) {
        setPartUrlMethodObject(instruction, mdl);

        return dtc.aud(partUrl, method, jsonObject);
    }

    public List<AbsenMdl> get(AbsenMdl mdl) {
        List<AbsenMdl> mdls = new ArrayList<>();

        setPartUrlMethodObject("get", mdl);

        String inputLine = dtc.get(partUrl, method, jsonObject);
        if (inputLine != null) {
            JSONArray jsonArray = new JSONArray(inputLine);
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject jo = jsonArray.getJSONObject(i);

                mdls.add(setMdl(jo));
            }
        }

        return mdls;
    }
}
package dtc;

import mdl.AdminMdl;
import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class AdminDtc {
    private String partUrl;
    private String method;
    private JSONObject jsonObject;

    private Dtc dtc = new Dtc();

    private void setPartUrlMethodObject(String instruction, AdminMdl mdl) {
        jsonObject = new JSONObject();

        switch (instruction) {
            case "add":
                partUrl = "admin";
                method = "POST";

                jsonObject.put("username", mdl.username);
                jsonObject.put("password", mdl.password);
                jsonObject.put("nama", mdl.nama);
                break;
            case "upd":

```

```

        partUrl = "admin/" + mdl.idadmin;
        method = "PUT";

        jsonObject.put("username", mdl.username);
        jsonObject.put("password", mdl.password);
        jsonObject.put("nama", mdl.nama);
        break;
    case "del":
        partUrl = "admin/" + mdl.idadmin;
        method = "DELETE";
        break;
    case "get":
        if (mdl == null) {
            partUrl = "admin";
            method = "GET";
        } else {
            partUrl = "admin/some";
            method = "POST";

            jsonObject.put("idadmin", mdl.idadmin);
        }
        break;
    case "login":
        partUrl = "admin/login";
        method = "POST";

        jsonObject.put("username", mdl.username);
        jsonObject.put("password", mdl.password);
        break;
    }
}

private AdminMdl setMdl(JSONObject jo) {
    AdminMdl mdl = new AdminMdl();
    mdl.idadmin = Integer.toString(jo.getInt("idadmin"));
    mdl.username = jo.getString("username");
    mdl.password = jo.getString("password");
    mdl.nama = jo.getString("nama");

    return mdl;
}

public int aud(String instruction, AdminMdl mdl) {
    setPartUrlMethodObject(instruction, mdl);

    return dtc.aud(partUrl, method, jsonObject);
}

public List<AdminMdl> get(AdminMdl mdl) {
    List<AdminMdl> mdls = new ArrayList<>();

    setPartUrlMethodObject("get", mdl);

    String inputLine = dtc.get(partUrl, method, jsonObject);
    if (inputLine != null) {
        JSONArray jsonArray = new JSONArray(inputLine);
        for (int i = 0; i < jsonArray.length(); i++) {
            JSONObject jo = jsonArray.getJSONObject(i);

            mdls.add(setMdl(jo));
        }
    }
}

```

```

        return mdls;
    }

    public AdminMdl login(AdminMdl mdl) {
        AdminMdl m = null;

        setPartUrlMethodObject("login", mdl);

        String inputLine = dtc.get(partUrl, method, jsonObject);
        if (inputLine != null) {
            JSONObject jo = new JSONObject(inputLine);

            m = setMdl(jo);
        }

        return m;
    }
}

package dtc;

import mdl.ChartMdl;
import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class ChartDtc {
    private String partUrl;
    private String method;
    private JSONObject jsonObject;

    private Dtc dtc = new Dtc();

    private void setPartUrlMethodObject(String instruction, ChartMdl mdl) {
        jsonObject = new JSONObject();

        switch (instruction) {
            case "chartpegawai":
                partUrl = "absen/chartpegawai";
                method = "POST";

                jsonObject.put("pegawai_nip", mdl.pegawai_nip);
                break;
        }
    }

    public List<ChartMdl> getChartPegawai(ChartMdl mdl) {
        List<ChartMdl> mdls = new ArrayList<>();

        setPartUrlMethodObject("chartpegawai", mdl);

        String inputLine = dtc.get(partUrl, method, jsonObject);
        if (inputLine != null) {
            JSONArray jsonArray = new JSONArray(inputLine);
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject jo = jsonArray.getJSONObject(i);

                ChartMdl m = new ChartMdl();
                m.month = Integer.toString(jo.getInt("month"));
                m.year = Integer.toString(jo.getInt("year"));
                m.count = Integer.toString(jo.getInt("count"));
                m.pegawai_nip = jo.getString("pegawai_nip");
            }
        }
    }
}

```



```

        mdls.add(m);
    }
}

return mdls;
}
}

package dtc;

public class CVDtc {
    private AdminDtc adminDtc = new AdminDtc();
}

package dtc;

import cnn.Cnn;
import com.enwif.hcrr.HCRR;
import hlp.InfoDialog;
import org.json.JSONObject;

import java.awt.*;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.nio.file.Paths;

class Dtc {
    private String audget(String partUrl, String method, JSONObject jsonObject) {
        HttpResponse<String> httpResponse = HCRR.sendSync(Cnn.url + partUrl, method,
        jsonObject.toString());
        String r = httpResponse.body();

        System.out.println("Dtc : " + r);

        return r;
    }

    private void adFile(String partUrl, String method, JSONObject jsonObject, String
    pathFile, String nameFile) {
        HCRR.sendSyncFile(Cnn.url + partUrl, method, jsonObject.toString(), pathFile,
        nameFile);
    }

    int aud(String partUrl, String method, JSONObject jsonObject) {
        String inputLine = audget(partUrl, method, jsonObject);
        if (!inputLine.isEmpty()) {
            if (inputLine.equals("invalid") || inputLine.equals("err")) {
                InfoDialog.showDlg("Error", inputLine);

                return 0;
            }
            else return Integer.parseInt(inputLine);
        } else return 0;
    }

    String get(String partUrl, String method, JSONObject jsonObject) {
        String inputLine = audget(partUrl, method, jsonObject);
        if (!inputLine.isEmpty()) {
            if (inputLine.equals("invalid") || inputLine.equals("err")) {

```

```

        InfoDialog.showDlg("Error", inputLine);

        return null;
    }
    else return inputLine;
} else return null;
}

void getFile(String partUrl, String method, JSONObject jsonObject, String
pathFile, String nameFile) {
    adFile(partUrl, method, jsonObject, pathFile, nameFile);

    try {
        Desktop.getDesktop().open(new File(pathFile + nameFile));
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public void test(String partUrl, String method, JSONObject jsonObject) {
    String path = System.getProperty("user.home") + "/Downloads/";
    String file = "xxx.pdf";

    HttpClient httpClient = HttpClient.newHttpClient();

    HttpRequest httpRequest = null;
    try {
        httpRequest = HttpRequest.newBuilder()
            .uri(URI.create(Cnn.url + partUrl))
            .header("Content-Type", "multipart/form-data")
            .method(method, HttpRequest.BodyPublishers.ofFile(Paths.get(path +
file)))
            .build();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }

    try {
        HttpResponse<String> httpResponse = httpClient.send(httpRequest,
HttpResponse.BodyHandlers.ofString());
        String r = httpResponse.body();
        System.out.println("FileFileFile : " + r);
    } catch (IOException | InterruptedException e) {
        e.printStackTrace();
    }
}
}

package dtc;

import mdl.FileMdl;
import org.json.JSONObject;

import java.time.LocalDate;

public class FileDtc {
    private String partUrl;
    private String method;
    private JSONObject jsonObject;
    private String pathFile;
    private String nameFile;

    private Dtc dtc = new Dtc();
}

```

```

private void setPartUrlMethodObject(String instruction, FileMdl mdl) {
    JSONObject = new JSONObject();

    switch (instruction) {
        case "addkenaikan":
            partUrl = "file/kenaikan";
            method = "POST";

            JSONObject.put("id", mdl.id);
            JSONObject.put("file", mdl.fileByte64);
            break;

        case "delkenaikan":
            partUrl = "file/kenaikan/test/" + mdl.id;
            method = "DELETE";
            break;

        case "getkenaikan":
            partUrl = "file/kenaikan/" + mdl.id;
            method = "GET";

            pathFile = System.getProperty("user.home") + "/Downloads/";
            nameFile = "Kenaikan_" + LocalDate.now() + ".pdf";
            break;

        case "addmutasi":
            partUrl = "file/mutasi";
            method = "POST";

            JSONObject.put("id", mdl.id);
            JSONObject.put("file", mdl.fileByte64);
            break;

        case "getmutasi":
            partUrl = "file/mutasi/" + mdl.id;
            method = "GET";

            pathFile = System.getProperty("user.home") + "/Downloads/";
            nameFile = "Mutasi_" + LocalDate.now() + ".pdf";
            break;
    }
}

public int aud(String instruction, FileMdl mdl) {
    setPartUrlMethodObject(instruction, mdl);

    return dtc.aud(partUrl, method, JSONObject);
}

public void get(String instruction, FileMdl mdl) {
    setPartUrlMethodObject(instruction, mdl);

    dtc.getFile(partUrl, method, JSONObject, pathFile, nameFile);
}
}
package dtc;

import mdl.KenaikanMdl;
import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;

```

```

import java.util.List;

public class KenaikanDtc {
    private String partUrl;
    private String method;
    private JSONObject jsonObject;

    private Dtc dtc = new Dtc();

    private void setPartUrlMethodObject(String instruction, KenaikanMdl mdl) {
        jsonObject = new JSONObject();

        switch (instruction) {
            case "add":
                partUrl = "kenaikan";
                method = "POST";

                jsonObject.put("nomor", mdl.nomor);
                jsonObject.put("tmt", mdl.tmt);
                jsonObject.put("jabatan", mdl.jabatan);
                jsonObject.put("unitkerja", mdl.unitkerja);
                jsonObject.put("pangkatlama", mdl.pangkatlama);
                jsonObject.put("pangkatbaru", mdl.pangkatbaru);
                jsonObject.put("golonganlama", mdl.golonganlama);
                jsonObject.put("golonganbaru", mdl.golonganbaru);
                jsonObject.put("pegawai_nip", mdl.pegawai_nip);
                break;
            case "upd":
                partUrl = "kenaikan/" + mdl.idkenaikan;
                method = "PUT";

                jsonObject.put("nomor", mdl.nomor);
                jsonObject.put("tmt", mdl.tmt);
                jsonObject.put("jabatan", mdl.jabatan);
                jsonObject.put("unitkerja", mdl.unitkerja);
                jsonObject.put("pangkatlama", mdl.pangkatlama);
                jsonObject.put("pangkatbaru", mdl.pangkatbaru);
                jsonObject.put("golonganlama", mdl.golonganlama);
                jsonObject.put("golonganbaru", mdl.golonganbaru);
                jsonObject.put("pegawai_nip", mdl.pegawai_nip);
                break;
            case "del":
                partUrl = "kenaikan/" + mdl.idkenaikan;
                method = "DELETE";
                break;
            case "get":
                if (mdl == null) {
                    partUrl = "kenaikan";
                    method = "GET";
                } else {
                    partUrl = "kenaikan/some";
                    method = "POST";

                    jsonObject.put("pegawai_nip", mdl.pegawai_nip);
                }
                break;
        }
    }

    private KenaikanMdl setMdl(JSONObject jo) {
        KenaikanMdl mdl = new KenaikanMdl();
        mdl.idkenaikan = Integer.toString(jo.getInt("idkenaikan"));
        mdl.nomor = jo.getString("nomor");
    }
}

```

```

        mdl.tmt = jo.getString("tmt");
        mdl.jabatan = jo.getString("jabatan");
        mdl.unitkerja = jo.getString("unitkerja");
        mdl.pangkatlama = jo.getString("pangkatlama");
        mdl.pangkatbaru = jo.getString("pangkatbaru");
        mdl.golonganlama = jo.getString("golonganlama");
        mdl.golonganbaru = jo.getString("golonganbaru");
        mdl.pegawai_nip = jo.getString("pegawai_nip");

        return mdl;
    }

    public int aud(String instruction, KenaikanMdl mdl) {
        setPartUrlMethodObject(instruction, mdl);

        return dtc.aud(partUrl, method, jsonObject);
    }

    public List<KenaikanMdl> get(KenaikanMdl mdl) {
        List<KenaikanMdl> mdls = new ArrayList<>();

        setPartUrlMethodObject("get", mdl);

        String inputLine = dtc.get(partUrl, method, jsonObject);
        if (inputLine != null) {
            JSONArray jsonArray = new JSONArray(inputLine);
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject jo = jsonArray.getJSONObject(i);

                mdls.add(setMdl(jo));
            }
        }

        return mdls;
    }
}

package dtc;

import mdl.MutasiMdl;
import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class MutasiDtc {
    private String partUrl;
    private String method;
    private JSONObject jsonObject;

    private Dtc dtc = new Dtc();

    private void setPartUrlMethodObject(String instruction, MutasiMdl mdl) {
        jsonObject = new JSONObject();

        switch (instruction) {
            case "add":
                partUrl = "mutasi";
                method = "POST";

                jsonObject.put("nomor", mdl.nomor);
                jsonObject.put("tmt", mdl.tmt);
                jsonObject.put("jabatanlama", mdl.jabatanlama);

```

```

        jsonObject.put("jabatanbaru", mdl.jabatanbaru);
        jsonObject.put("kelasjabatanlama", mdl.kelasjabatanlama);
        jsonObject.put("kelasjabatanbaru", mdl.kelasjabatanbaru);
        if (mdl.besartunjangan != null && !mdl.besartunjangan.equals(""))
jSONObject.put("besartunjangan", mdl.besartunjangan);
        if (mdl.keterangan != null && !mdl.keterangan.equals(""))
jSONObject.put("keterangan", mdl.keterangan);
        jsonObject.put("pegawai_nip", mdl.pegawai_nip);
        break;
    case "upd":
        partUrl = "mutasi/" + mdl.idmutasi;
        method = "PUT";

        jsonObject.put("nomor", mdl.nomor);
        jsonObject.put("tmt", mdl.tmt);
        jsonObject.put("jabatanlama", mdl.jabatanlama);
        jsonObject.put("jabatanbaru", mdl.jabatanbaru);
        jsonObject.put("kelasjabatanlama", mdl.kelasjabatanlama);
        jsonObject.put("kelasjabatanbaru", mdl.kelasjabatanbaru);
        if (mdl.besartunjangan != null && !mdl.besartunjangan.equals(""))
jSONObject.put("besartunjangan", mdl.besartunjangan);
        if (mdl.keterangan != null && !mdl.keterangan.equals(""))
jSONObject.put("keterangan", mdl.keterangan);
        jsonObject.put("pegawai_nip", mdl.pegawai_nip);
        break;
    case "del":
        partUrl = "mutasi/" + mdl.idmutasi;
        method = "DELETE";
        break;
    case "get":
        if (mdl == null) {
            partUrl = "mutasi";
            method = "GET";
        } else {
            partUrl = "mutasi/some";
            method = "POST";

            jsonObject.put("pegawai_nip", mdl.pegawai_nip);
        }
        break;
    }
}

private MutasiMdl setMdl(JSONObject jo) {
    MutasiMdl mdl = new MutasiMdl();
    mdl.idmutasi = Integer.toString(jo.getInt("idmutasi"));
    mdl.nomor = jo.getString("nomor");
    mdl.tmt = jo.getString("tmt");
    mdl.jabatanlama = jo.getString("jabatanlama");
    mdl.jabatanbaru = jo.getString("jabatanbaru");
    mdl.kelasjabatanlama = jo.getString("kelasjabatanlama");
    mdl.kelasjabatanbaru = jo.getString("kelasjabatanbaru");
    if (!jo.isNull("besartunjangan")) mdl.besartunjangan =
Integer.toString(jo.getInt("besartunjangan"));
    if (!jo.isNull("keterangan")) mdl.keterangan = jo.getString("keterangan");
    mdl.pegawai_nip = jo.getString("pegawai_nip");

    return mdl;
}

public int aud(String instruction, MutasiMdl mdl) {
    setPartUrlMethodObject(instruction, mdl);
}

```

```

        return dtc.aud(partUrl, method, jsonObject);
    }

    public List<MutasiMdl> get(MutasiMdl mdl) {
        List<MutasiMdl> mdls = new ArrayList<>();

        setPartUrlMethodObject("get", mdl);

        String inputLine = dtc.get(partUrl, method, jsonObject);
        if (inputLine != null) {
            JSONArray jsonArray = new JSONArray(inputLine);
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject jo = jsonArray.getJSONObject(i);

                mdls.add(setMdl(jo));
            }
        }

        return mdls;
    }
}

package dtc;

import mdl.PegawaiMdl;
import org.json.JSONArray;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

public class PegawaiDtc {
    private String partUrl;
    private String method;
    private JSONObject jsonObject;

    private Dtc dtc = new Dtc();

    private void setPartUrlMethodObject(String instruction, PegawaiMdl mdl) {
        jsonObject = new JSONObject();

        switch (instruction) {
            case "add":
                partUrl = "pegawai";
                method = "POST";

                jsonObject.put("nip", mdl.nip);
                jsonObject.put("nrp", mdl.nrp);
                jsonObject.put("nama", mdl.nama);
                jsonObject.put("tmplhr", mdl.tmplhr);
                jsonObject.put("tgllhr", mdl.tgllhr);
                jsonObject.put("jabatan", mdl.jabatan);
                jsonObject.put("unitkerja", mdl.unitkerja);
                jsonObject.put("golongan", mdl.golongan);
                jsonObject.put("email", mdl.email);
                jsonObject.put("nohp", mdl.nohp);
                jsonObject.put("status", mdl.status);
                break;
            case "upd":
                partUrl = "pegawai/" + mdl.old_nip;
                method = "PUT";

                jsonObject.put("nip", mdl.nip);
                jsonObject.put("nrp", mdl.nrp);

```

```

        jsonObject.put("nama", mdl.nama);
        jsonObject.put("tmplhr", mdl.tmplhr);
        jsonObject.put("tgllhr", mdl.tgllhr);
        jsonObject.put("jabatan", mdl.jabatan);
        jsonObject.put("unitkerja", mdl.unitkerja);
        jsonObject.put("golongan", mdl.golongan);
        jsonObject.put("email", mdl.email);
        jsonObject.put("nohp", mdl.nohp);
        jsonObject.put("status", mdl.status);
        break;
    case "del":
        partUrl = "pegawai/" + mdl.nip;
        method = "DELETE";
        break;
    case "get":
        if (mdl == null) {
            partUrl = "pegawai";
            method = "GET";
        } else {
            partUrl = "pegawai/some";
            method = "POST";

            jsonObject.put("nip", mdl.nip);
        }
        break;
    case "src":
        partUrl = "pegawai/src";
        method = "POST";

        jsonObject.put("nip", mdl.nip);
        jsonObject.put("nama", mdl.nama);
        break;
    }
}

private PegawaiMdl setMdl(JSONObject jo) {
    PegawaiMdl mdl = new PegawaiMdl();
    mdl.nip = jo.getString("nip");
    mdl.nrp = jo.getString("nrp");
    mdl.nama = jo.getString("nama");
    mdl.tmplhr = jo.getString("tmplhr");
    mdl.tgllhr = jo.getString("tgllhr");
    mdl.jabatan = jo.getString("jabatan");
    mdl.unitkerja = jo.getString("unitkerja");
    mdl.golongan = jo.getString("golongan");
    mdl.email = jo.getString("email");
    mdl.nohp = jo.getString("nohp");
    mdl.status = jo.getString("status");

    return mdl;
}

public int aud(String instruction, PegawaiMdl mdl) {
    setPartUrlMethodObject(instruction, mdl);

    return dtc.aud(partUrl, method, jsonObject);
}

public List<PegawaiMdl> get(PegawaiMdl mdl) {
    List<PegawaiMdl> mdls = new ArrayList<>();

    setPartUrlMethodObject("get", mdl);

```



```

String inputLine = dtc.get(partUrl, method, jsonObject);
if (inputLine != null) {
    JSONArray jsonArray = new JSONArray(inputLine);
    for (int i = 0; i < jsonArray.length(); i++) {
        JSONObject jo = jsonArray.getJSONObject(i);

        mdls.add(setMdl(jo));
    }
}

return mdls;
}

public List<PegawaiMdl> src(PegawaiMdl mdl) {
    List<PegawaiMdl> mdls = new ArrayList<>();

    setPartUrlMethodObject("src", mdl);

    String inputLine = dtc.get(partUrl, method, jsonObject);
    if (inputLine != null) {
        JSONArray jsonArray = new JSONArray(inputLine);
        for (int i = 0; i < jsonArray.length(); i++) {
            JSONObject jo = jsonArray.getJSONObject(i);

            mdls.add(setMdl(jo));
        }
    }

    return mdls;
}
}
package dtc;

import mdl.QRAbsenMdl;
import org.json.JSONObject;

public class QRAbsenDtc {
    private String partUrl;
    private String method;
    private JSONObject jsonObject;

    private Dtc dtc = new Dtc();

    private void setPartUrlMethodObject(String instruction, QRAbsenMdl mdl) {
        jsonObject = new JSONObject();

        switch (instruction) {
            case "init":
                partUrl = "qrabsen/init";
                method = "POST";
                break;
            case "qr":
                partUrl = "qrabsen/qr";
                method = "POST";

                jsonObject.put("qr", mdl.qr);
                break;
            case "get":
                partUrl = "qrabsen/getverify";
                method = "POST";
                break;
        }
    }
}

```

```

private QRAbsenMdl setMdl(JSONObject jo) {
    QRAbsenMdl mdl = new QRAbsenMdl();
    mdl.nip = jo.getString("nip");
    mdl.nrp = jo.getString("nrp");
    mdl.nama = jo.getString("nama");
    mdl.jabatan = jo.getString("jabatan");
    mdl.unitkerja = jo.getString("unitkerja");

    return mdl;
}

public int aud(String instruction, QRAbsenMdl mdl) {
    setPartUrlMethodObject(instruction, mdl);

    return dtc.aud(partUrl, method, jsonObject);
}

public QRAbsenMdl get() {
    QRAbsenMdl mdl = null;

    setPartUrlMethodObject("get", mdl);

    String inputLine = dtc.get(partUrl, method, jsonObject);
    if (inputLine != null && !inputLine.equals("0")) {
        JSONObject jo = new JSONObject(inputLine);

        mdl = setMdl(jo);
    }
    return mdl;
}
}
package mdl;

import java.time.Instant;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.time.format.FormatStyle;

public class AbsenMdl {
    public String idabsen, tgl, jammasuk, jampulang, pegawai_nip;
    public String nrp, nama;

    public String getIdabsen() {
        return idabsen;
    }

    public String getTgl() {
        return tgl;
    }

    public String getJammasuk() {
        return jammasuk;
    }

    public String getJampulang() {
        return jampulang;
    }

    public String getPegawai_nip() {
        return pegawai_nip;
    }
}

```

```

    public String getNrp() {
        return nrp;
    }

    public String getNama() {
        return nama;
    }

    public String getVw_tgl() {
        return
DateTimeFormatter.ofLocalizedDate(FormatStyle.LONG).format(Instant.parse(tgl).atZone(ZoneId.systemDefault()));
    }
}
package mdl;

import java.time.Instant;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.time.format.FormatStyle;

public class KenaikanMdl {
    public String idkenaikan, nomor, tmt, jabatan, unitkerja, pangkatlama,
pangkatbaru, golonganlama, golonganbaru, pegawai_nip;

    public String getIdkenaikan() {
        return idkenaikan;
    }

    public String getNomor() {
        return nomor;
    }

    public String getTmt() {
        return tmt;
    }

    public String getJabatan() {
        return jabatan;
    }

    public String getUnitkerja() {
        return unitkerja;
    }

    public String getPangkatlama() {
        return pangkatlama;
    }

    public String getPangkatbaru() {
        return pangkatbaru;
    }

    public String getGolonganlama() {
        return golonganlama;
    }

    public String getGolonganbaru() {
        return golonganbaru;
    }

    public String getPegawai_nip() {
        return pegawai_nip;
    }
}

```

```

    }

    public String getVw_tmt() {
        return
DateTimeFormatter.ofLocalizedDate(FormatStyle.LONG).format(Instant.parse(tmt).atZone(ZoneId.systemDefault()));
    }
}
package mdl;

import java.time.Instant;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.time.format.FormatStyle;

public class MutasiMdl {
    public String idmutasi, nomor, tmt, jabatanlama, jabatanbaru, kelasjabatanlama,
kelasjabatanbaru, besartunjangannya, keterangan, pegawai_nip;

    public String getIdmutasi() {
        return idmutasi;
    }

    public String getNomor() {
        return nomor;
    }

    public String getTmt() {
        return tmt;
    }

    public String getJabatanlama() {
        return jabatanlama;
    }

    public String getJabatanbaru() {
        return jabatanbaru;
    }

    public String getKelasjabatanlama() {
        return kelasjabatanlama;
    }

    public String getKelasjabatanbaru() {
        return kelasjabatanbaru;
    }

    public String getBesartunjangannya() {
        return besartunjangannya;
    }

    public String getKeterangan() {
        return keterangan;
    }

    public String getPegawai_nip() {
        return pegawai_nip;
    }

    public String getVw_tmt() {
        return
DateTimeFormatter.ofLocalizedDate(FormatStyle.LONG).format(Instant.parse(tmt).atZone(ZoneId.systemDefault()));
    }
}

```

```
    }  
}  
package mdl;  
  
import java.time.Instant;  
import java.time.ZoneId;  
import java.time.format.DateTimeFormatter;  
import java.time.format.FormatStyle;  
  
public class PegawaiMdl {  
    public String nip, nrp, nama, tmlhr, tglhr, jabatan, unitkerja, golongan, email,  
    nohp, status;  
    public String old_nip;  
  
    public String getNip() {  
        return nip;  
    }  
  
    public String getNrp() {  
        return nrp;  
    }  
  
    public String getNama() {  
        return nama;  
    }  
  
    public String getTmlhr() {  
        return tmlhr;  
    }  
  
    public String getTglhr() {  
        return tglhr;  
    }  
  
    public String getJabatan() {  
        return jabatan;  
    }  
  
    public String getUnitkerja() {  
        return unitkerja;  
    }  
  
    public String getGolongan() {  
        return golongan;  
    }  
  
    public String getEmail() {  
        return email;  
    }  
  
    public String getNohp() {  
        return nohp;  
    }  
  
    public String getStatus() {  
        return status;  
    }  
  
    public String getOld_nip() {  
        return old_nip;  
    }  
  
    public String getVw_tglhr() {
```

```
        return
DateTimeFormatter.ofLocalizedDate(FormatStyle.LONG).format(Instant.parse(tgl1hr).atZone(ZoneId.systemDefault()));
    }

    public String getVw_status() {
        switch (status) {
            case "1":
                return "Aktif";
            case "0":
                return "Tidak Aktif";
            default:
                return "Tidak Aktif";
        }
    }
}
package mdl;

public class QRAbsenMdl {
    public String qr;
    public String nip, nrp, nama, jabatan, unitkerja;
}
```



SURAT KEPUTUSAN
REKTOR IIB DARMAJAYA
NOMOR : SK.0605/DMJ/DFIK/BAAK/XII-18

Tentang
Dosen Pembimbing Skripsi
Program Studi S1 Sistem Informasi

REKTOR IIB DARMAJAYA

- Memperhatikan :** 1. Bahwa dalam rangka usaha peningkatan mutu dan peranan IIB Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam Skripsi.
- Menimbang :** 1. Laporan dan usulan Ketua Program Studi S1 Sistem Informasi.
1. Bahwa untuk mengefektifkan tenaga pengajar dalam Skripsi mahasiswa perlu ditetapkan **Dosen Pembimbing Skripsi**.
2. Bahwa untuk maksud tersebut dipandang perlu menerbitkan Surat Keputusan Rektor.
- Mengingat :** 1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi
3. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya
4. STATUTA IBI Darmajaya
5. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi
6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.
- Menetapkan**
- Pertama :** Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Sistem Informasi.
- Kedua :** Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telah ditetapkan.
- Ketiga :** Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma penggajian dan honorarium IBI Darmajaya.
- Keempat :** Surat Keputusan ini berlaku sejak tanggal ditetapkan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

Ditetapkan di : Bandar Lampung
Pada tanggal : 10 Desember 2018
a.n. Rektor IIB Darmajaya,
Dekan Fakultas Ilmu Komputer

Sriyanto S. Kom., M.M.
NIK. 00210800

1. Ketua Jurusan S1 Sistem Informasi
2. Yang bersangkutan
3. Arsip

Lampiran : Surat Keputusan Rektor IIB Darmajaya
 Nomor : SK. 0605/DMJ/DFIK/BAAK/XII-18
 Tanggal : 10 Desember 2018
 Perihal : Pembimbing Penulisan Skripsi
 Program Studi Strata Satu (S1) Sistem Informasi

JUDUL SKRIPSI DAN DOSEN PEMBIMBING
 PROGRAM STUDI STRATA SATU (S1) SISTEM INFORMASI

No	NAMA	NPM	JUDUL	PEMBIMBING
17	Muhammad Farhan Ramadhan	1511050105	Sistem Informasi Monitoring Hujan Di Kota Bandar Lampung	Deppi Linda, S.Kom, M.Ti
18	Laura Desvia	1511050117	Sistem Informasi Kepegawaian pada Kejaksaan Negeri Lampung Timur	Dinda Yulawati, S.Kom, M.Ti
19	ENI SETIAWATI	1611059038	Sistm Informasi Presensi Siswa Berbasis Finger Print Dan SMS Gateway Di SMP Negeri 3 Katibung	
20	Tiara Utami	1511050026	Sistem Informasi Klinik Kecantikan Pada Kartika Aesthetic Sdi Bandar Lampung Berbasis Desktop Dan SMS Gateway	
21	DENDY RAMADHANI PRANATA	1411050072	Perancangan Sistem Informasi E-Document Kepegawaian Pada Kantor Balai Besar Wilayah Mesuji Sekampung Bandar Lampung	Halimah, S.Kom., M.T.I
22	Achmad Ramadhan Adfa	1511050051	Perancangan Sistem Informasi Museum Lampung Berbasis Website	
23	Felista Klaralia	1511050047	Penerapan Algoritma Naive Bayes Dalam Mengidentifikasi Pola Penyakit Malaria Pada Puskesmas Hanura	
24	Intan Wareka	1511050118	Penerapan Metode Naive Bayes Untuk Penentuan Siswa/i Terbaik di SMAN 1 Pardasuka	
25	Tita Herawati	1511050099	Sistem Informasi Geografis Pemilihan Perumahan Menggunakan Metode Weighted Product (WP) Berbasis Android	
26	Anggi Asmara Putri	1511050164	Sistem Informasi Pengolahan E-Document Penelitian Dosen Pada Jurusan Sistem Informasi IIB Darmaja	Hendra Kurniawan, S.Kom, M.Ti
27	Sasiya Nadira	1511050082	Penerapan Knowledge Management System Berbasis Web (Studi Kasus: Spesialis Penyakit Jantung dan Spesialis Penyakit Dalam RSUD Dr. H. Abdul Moeloek)	
28	Ayu Venilia	1511050022	Sistem Informasi Pemetaan Lokasi Peternakan dengan Pendekatan Metode TOPSIS Berbasis Android	
29	M. Zaky Fanani	1611059005P	Rancang Bangun Sistem Informasi Presensi Mahasiswa Menggunakan QR Code Berbasis Android Pada UIN Raden Intan Bandar Lampung	
30	Aditya Edgar Ramadhan	1511050078	Perancangan Mobile E-Learning Untuk Asistensi Dan Media Pembelajaran Menggunakan Rapid Application Development) berbasis Android Study Kasus di Bimbingan Belajar San Education	Hernanto, S.Kom, M.Ti
31	Meliza Anjika Jaya	1511059017p	Rancang Bangun Sistem Informasi Tracer Study Alumni Pada Fakultas Tarbiyah Universitas Islam Negeri Raden Intan Lampung	Indera, S.Kom, M.Ti



FORMULIR

BIRO ADMINISTRASI AKADEMIK KEMAHASISWAAN (BAAK)

FORM KONSULTASI/BIMBINGAN SKRIPSI/TUGAS AKHIR *)

NAMA : LAURA DESVIA
 NIM : 1511050117
 PEMBIMBING I : Dona Yulianiwati, S.kom., M.T.I
 PEMBIMBING II :
 JUDUL LAPORAN : Sistem Informasi Kepegawain Pada Kejaksaan
 Negeri Lampung Timur
 WAKTU : s.d (6+2 bulan)

No	HARI/TANGGAL	HASIL KONSULTASI	PARAF
1	Selasa 14/10 2018	Revisi bab 1	
2	Senin 29/10 2018	"	
3	Selasa 09/11 2018	ACC Bab 1 dan Revisi 2	
4	Senin 12/11 2018	ACC bab 2	
5	Jumat 19/12 2018	Revisi bab 3	
6	Rabu 24/01 2019	ACC bab 3 dan Revisi bab 4	
7	Senin 05/02 2019	Revisi bab 4 dan Program	
8	Senin 19/02 2019	ACC Program Revisi bab 5	
9	Jumat 29/02 2019	ACC bab 5	
10	Jumat 30/02 2019	ACC sidang	

*) Coret yang tidak perlu

Bandar Lampung, 03-09-2019
Ketua Jurusan

(Nurjoko, S.kom., M.T.I)
NIK. 00490702