

**PERANCANGAN SISTEM PRESENSI *QUICK RESPONSE CODE*  
BERBASIS *WEB* DAN *ANDROID CLIENT*  
CV. INDOSCOTS BABY UTAMA**

**KERJA PRAKTEK**

Diajukan untuk memenuhi salah satu syarat kelulusan  
Matakuliah Kerja Praktek Jenjang Strata Satu (S-1) pada Program Studi Teknik  
Informatika Konsentrasi Sistem Informasi  
STMIK Mardira Indonesia

Oleh :

**SANDHY PUTRA**

**17110127**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
MARDIRA INDONESIA  
2020**

Terakreditasi Menurut Keputusan Badan Akreditasi Perguruan Tinggi  
Menteri Riset, Teknologi dan Pendidikan Tinggi Republik Indonesia  
Nomor : 2887/SK/BAN - PT/Akred/S/X//2018

## **PERNYATAAN KEASLIAN**

Dengan ini saya menyatakan bahwa Laporan Kerja Praktek dengan judul **“PERANCANGAN SISTEM *PRESENSI QUICK RESPONSE CODE* BERBASIS *WEB* DAN *ANDROID CLIENT* CV. INDOSCOTS BABY UTAMA”** ini beserta isinya adalah benar-benar hasil karya saya, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung saksi yang dibebankan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Bandung, Februari 2021

Sandhy Putra

## LEMBAR PERSETUJUAN

JUDUL : PERANCANGAN SISTEM PRESENSI *QUICK RESPONSE CODE*  
BERBASIS *WEB* DAN *ANDROID CLIENT* CV. INDOSCOTS BABY  
UTAMA

PENYUSUN : SANDHY PUTRA

NIM : 17110127

Bandung, Februari 2021

Menyetujui,

Pembimbing

Ketua Program Studi

Teknik Informatika – S1

Rohmat Nur Ibrahim, S.Kom, M.T.

NIK : 94.01.014

Rohmat Nur Ibrahim, S.Kom., M.T

NIK : 94.01.014

## LEMBAR PENGESAHAN

Panitia Ujian Seminar Penelitian  
Jenjang Strata Satu (S-1) Pada Program Studi Teknik Informatika  
Sekolah Tinggi Manajemen Informatika dan Komputer  
Mardira Indonesia Bandung

Bandung, Februari 2021

Telah Menguji dan Menganalisa Penelitian Dengan Judul  
PERANCANGAN SISTEM PRESENSI *QUICK RESPONSE CODE* BERBASIS  
*WEB* DAN *ANDROID CLIENT* CV. INDOSCOTS BABY UTAMA

Disusun Oleh :

Nama : Sandhy Putra

NIM : 17110127

Mengesahkan,

Penguji I

Penguji II

---

NIK :

---

NIK :

## ABSTRAK

Sistem presensi merupakan hal penting bagi suatu instansi atau perusahaan untuk mengetahui tingkat prestasi kehadiran dan kedisiplinan dari setiap karyawannya. Sistem presensi di CV.Indoscots Baby Utama masih menggunakan cara yang bersifat semi *Electronic Data Processing*, dimana sistem presensi yang sedang berjalan belum terkomputerisasi secara menyeluruh karena dilakukan dengan cara tulis tangan pada sebuah formulir presensi oleh karyawan perusahaan tersebut. Hal tersebut beresiko menimbulkan kecurangan karyawan dalam mengisi formulir presensi. Selain itu, perusahaan tersebut tidak mengetahui informasi keberadaan karyawannya yang sedang melakukan kebijakan *work from home* atau bekerja dari rumah. Tujuan penyusunan laporan kerja praktek ini adalah untuk membuat Rancangan Sistem Presensi *Quick Response Code* berbasis *web* dan *android client* dimana ada dua *interface* yang akan dibuat, yakni *interface web* dan *interface android*. Sistem tersebut dibangun menggunakan *framework php codeigniter* dan *IDE MIT App Inventor*. Metode penelitian yang digunakan dalam penyusunan laporan kerja praktek ini adalah metode deskriptif dan metode pengembangan sistem yang digunakan adalah metode *Object Oriented Analysis and Design (OOAD)* dengan pemodelan sistem yang digambarkan menggunakan diagram *Unified Modelling Language (UML)*. Dengan dibuatnya sistem ini diharapkan kegiatan presensi dapat berjalan dengan tepat, akurat, dan dapat digunakan dengan praktis.

**Kata Kunci :** Sistem Presensi, *Qr-code*, *OOAD*, *UML*, *Codeigniter*, *App Inventor*

## **ABSTRACT**

*The presence system is important for an agency or company to see the level of attendance and discipline of each employee. The presence system at CV Indoscots Baby Utama still uses a semi-Electronic Data Processing method, where the presence of the current system has not been completely computerized because it is done by handwriting on an presence form by the company's employees. This has the risk of causing employee fraud in filling out presence forms. Besides, the company does not know the information on the whereabouts of its employees who are implementing a work from home policy. The purpose of preparing this practical work report is to make a Design of Web-based Quick Response Code Presence System and an android client where there are two interfaces to be made, namely the web interface and the Android interface. The system was built using the PHP CodeIgniter framework and the MIT App Inventor IDE. The research method used in the preparation of this practical work report is the descriptive method and the system method which is the Object-Oriented Analysis and Design (OOAD) method with system modeling depicted using the Unified Modeling Language (UML) diagrams. With the creation of this system, it is hoped that attendance activities can be carried out appropriately, accurately, and can be used practically.*

**Keywords:** Presence System, Qr-code, OOAD, UML, Codeigniter, App Inventor

## KATA PENGANTAR

Alhamdulillah segala puji bagi Allah SWT, yang telah melimpahkan kasih-Nya sehingga penulis dapat menyelesaikan laporan kerja praktek dengan judul “Perancangan Sistem Presensi *Quick Response Code* Berbasis *Web* Dan *Android Client* Cv. Indoscots Baby Utama” tepat pada waktunya. Laporan kerja praktek ini disusun untuk memenuhi salah satu syarat kelulusan pada Matakuliah Kerja Praktek Jenjang Strata Satu (S-1) Program Studi Teknik Informatika Konsentrasi Sistem Informasi STMIK Mardira Indonesia.

Dalam penulisan laporan kerja praktek ini, penulis mendapatkan bantuan dari berbagai pihak, penulis juga mendapatkan dukungan moril maupun materil serta mendapatkan bimbingan yang membantu dalam menyelesaikan laporan kerja praktek ini. Dalam kesempatan ini penulis tidak lupa mengucapkan terima kasih secara tulus khususnya kepada:

1. Orang tua tercinta, Ibunda Eutik Sumariah, Ayahanda Asep Ahmad , dan saudara Muhammad Fadil yang telah memberikan dukungan baik moril maupun material serta doa untuk keberhasilan penulis dalam meraih gelar yang diharapkan.
2. Bapak DR. Marjito, M.Pd. selaku Ketua STMIK Mardira Indonesia.
3. Bapak Rohmat Nur Ibrahim, S.Kom, M.T. selaku Ketua Prodi sekaligus pembimbing yang telah berkenan meluangkan waktu, pikiran, dan tenaganya untuk memberikan bimbingan serta arahan kepada penulis.
4. Dosen STMIK Mardira Indonesia yang telah memberikan ilmu pengetahuan kepada penulis

5. Sahabat tercinta Zaky yang telah memberikan dukungan dan motivasi selama berjalannya penyusunan laporan kerja praktek ini.
6. Teman-teman seperjuangan, Dinda Devianty, Erin Maurina, Putri Mutia Septiani, Rina Lianti, Rizal Muhammad Dhuha Zaini, dan Shofitry Sri Astuty yang telah membantu penulis dalam proses penyelesaian laporan kerja praktek ini.
7. Kepada semua pihak yang tidak bisa penulis sebutkan satu persatu, semoga kebaikan yang telah diberikan dapat dibalas oleh Allah SWT.

Penulis menyadari masih terdapat kekurangan dan bukan merupakan pembahasan yang sempurna dalam penulisan laporan kerja praktek ini, oleh sebab itu penulis pun sangat mengharapkan ide, saran dan masukan yang bersifat membangun untuk laporan kerja praktek ini. Semoga laporan kerja praktek ini dapat bermanfaat untuk lingkungan pendidikan khususnya untuk kesempurnaan penyusunan terhadap laporan penelitian nantinya.

Bandung,      Februari 2021

Penulis



## DAFTAR ISI

	Halaman
PERNYATAAN KEASLIAN.....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN .....	iii
ABSTRAK .....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR SIMBOL DIAGRAM <i>USE CASE</i> .....	xx
DAFTAR SIMBOL DIAGRAM <i>ACTIVITY</i> .....	xxii
DAFTAR SIMBOL DIAGRAM <i>SEQUENCE</i> .....	xxiii
DAFTAR SIMBOL DIAGRAM <i>CLASS</i> .....	xxiv
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Identifikasi Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan dan Kegunaan Penelitian.....	4

1.4.1 Tujuan Penelitian .....	4
1.4.2 Kegunaan Penelitian .....	4
1.5 Objek dan Waktu Penelitian .....	5
1.6 Metodologi Penelitian .....	5
1.6.1 Metode Penelitian .....	5
1.6.2 Metode Pengembangan Sistem.....	6
1.7 Sistematika Penulisan.....	8
<b>BAB II LANDASAN TEORI .....</b>	<b>10</b>
2.1 Definisi Perancangan.....	10
2.2 Definisi Sistem .....	10
2.2.1 Karakteristik Sistem .....	11
2.3 Definisi Presensi.....	12
2.4 Definisi Sistem Presensi .....	12
2.5 Pengertian <i>Quick Response Code (QR-Code)</i> .....	13
2.5.1 Struktur <i>Quick Response Code (QR-Code)</i> .....	13
2.5.2 Karakteristik <i>Quick Response Code (QR-Code)</i> .....	14
2.5.3 Alat Pemindai <i>Quick Response Code (QR-Code)</i> .....	17
2.6 Pengertian <i>Android</i> .....	17
2.6.1 Versi Sistem Android .....	18
2.7 Pengertian <i>Web</i> .....	23

2.7.1 Jenis-Jenis <i>Web</i> .....	24
2.7.2 Pemrograman <i>Web</i> .....	25
2.8 Definisi Basis Data ( <i>Database</i> ) .....	26
2.8.1 Keuntungan dan Kerugian Penggunaan Basis Data ( <i>Database</i> ) .....	27
2.9 Pendukung Perancangan.....	27
2.9.1 <i>Object Oriented Analysis Design (OOAD)</i> .....	27
2.9.2 <i>Unified Modelling Language (UML)</i> .....	28
2.10 Perangkat Lunak Pendukung.....	31
2.10.1 <i>Object Oriented Analysis Design (OOAD)</i> .....	32
2.10.2 Konsep MVC ( <i>Model View Controller</i> ) .....	33
2.10.3 <i>IDE MIT APP Inventor</i> .....	34
2.10.4 <i>MySql Database</i> .....	34
<b>BAB III ANALISIS DAN PERANCANGAN SISTEM.....</b>	<b>36</b>
3.1 CV. Indoscots Baby Utama .....	36
3.1.1 Struktur Organisasi .....	36
3.2 Analisis Proses Bisnis Sistem Lama.....	41
3.3 Analisis SWOT.....	42
3.4 Gambaran Umum Sistem Yang Diusulkan .....	43
3.5 <i>Unified Modelling Language (UML)</i> .....	43
3.5.1 <i>Usecase Diagram</i> .....	43

3.5.2 <i>Usecase Spesification</i> .....	46
3.5.3 <i>Activity Diagram</i> .....	60
3.5.4 <i>Sequence Diagram</i> .....	77
3.5.4 <i>Class Diagram</i> .....	88
3.6 Struktur Basis Data .....	89
3.7 Pengkodean .....	90
3.8 Struktur Menu .....	91
3.8.1 Struktur Menu Antarmuka <i>Web</i> .....	91
3.8.1 Struktur Menu Antarmuka Ponsel <i>Android</i> .....	92
3.9 Perancangan Antarmuka .....	92
<b>BAB IV IMPLEMENTASI SISTEM</b> .....	<b>102</b>
4.1 Implementasi Hardware .....	102
4.2 Tampilan Struktur Tabel <i>Database</i> .....	102
4.2.1 Tampilan Struktur Tabel tb_karyawan .....	102
4.2.2 Tampilan Struktur Tabel tb_karyawan .....	102
4.2.1 Tampilan Struktur Tabel tb_divisi .....	103
4.2.1 Tampilan Struktur Tabel tb_presensi .....	103
4.2.1 Tampilan Struktur Tabel tb_user .....	103
4.3 Tampilan Program .....	104
<b>BAB V PENUTUP</b> .....	<b>114</b>

5.1 Kesimpulan .....	114
5.2 Saran .....	114
<b>DAFTAR PUSTAKA .....</b>	<b>115</b>

## DAFTAR TABEL

	Halaman
Tabel 3.1 Pendefinisian Aktor .....	44
Tabel 3.2 Pendefinisian <i>Usecase</i> .....	44
Tabel 3.3 <i>Usecase Spesification Login</i> .....	47
Tabel 3.4 <i>Usecase Spesification</i> Tambah Data Karyawan .....	47
Tabel 3.5 <i>Usecase Spesification Update</i> Data Karyawan .....	48
Tabel 3.6 <i>Usecase Spesification</i> Hapus Data Karyawan .....	49
Tabel 3.7 <i>Usecase Spesification</i> Tambah Data Divisi .....	50
Tabel 3.8 <i>Usecase Spesification Update</i> Data Divisi .....	51
Tabel 3.9 <i>Usecase Spesification</i> Hapus Data Divisi .....	51
Tabel 3.10 <i>Usecase Spesification</i> Tambah Data Presensi .....	52
Tabel 3.11 <i>Usecase Spesification Update</i> Data Presensi .....	53
Tabel 3.12 <i>Usecase Spesification</i> Hapus Data Presensi .....	54
Tabel 3.13 <i>Usecase Spesification</i> Tambah Data <i>User</i> .....	54
Tabel 3.14 <i>Usecase Spesification Update</i> Data <i>User</i> .....	55
Tabel 3.15 <i>Usecase Spesification</i> Hapus Data <i>User</i> .....	56
Tabel 3.16 <i>Usecase Spesification</i> Monitor Peta <i>Work From Home</i> .....	56
Tabel 3.17 <i>Usecase Spesification</i> Cetak Laporan Presensi .....	57

Tabel 3.18 <i>Usecase Spesification</i> Presensi Masuk .....	58
Tabel 3.19 <i>Usecase Spesification</i> Presensi Pulang .....	58
Tabel 3.20 <i>Usecase Spesification</i> Lihat Kartu Presensi .....	59
Tabel 3.21 <i>Usecase Spesification</i> Logout .....	60
Tabel 3.23 Struktur tb_divisi .....	89
Tabel 3.24 Struktur tb_presensi .....	89
Tabel 3.25 Struktur tb_user .....	90

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Struktur <i>qr-code</i> .....	13
Gambar 2.2 Pembacaan <i>qr-code</i> dari segala arah .....	15
Gambar 2.3 Kerusakan pada <i>QR-Code</i> .....	16
Gambar 3.1 Struktur Organisasi CV. Indoscots Baby Utama .....	36
Gambar 3.2 Proses bisnis sistem yang sedang berjalan .....	42
Gambar 3.3 <i>Usecase Diagram</i> .....	46
Gambar 3.4 Activity Diagram Login .....	61
Gambar 3.5 <i>Activity Diagram</i> Tambah Data Karyawan .....	62
Gambar 3.6 <i>Activity Diagram Update</i> Data Karyawan.....	63
Gambar 3.7 <i>Activity Diagram</i> Hapus Data Karyawan .....	64
Gambar 3.8 <i>Activity Diagram</i> Tambah Data Divisi .....	65
Gambar 3.9 <i>Activity Diagram Update</i> Data Divisi .....	66
Gambar 3.10 <i>Activity Diagram</i> Hapus Data Divisi.....	67
Gambar 3.11 <i>Activity Diagram</i> Tambah Data Presensi .....	68
Gambar 3.12 <i>Activity Diagram Update</i> Data Presensi.....	69
Gambar 3.13 <i>Activity Diagram</i> Hapus Data Presensi .....	70
Gambar 3.14 <i>Activity Diagram</i> Tambah Data User .....	71
Gambar 3.15 <i>Activity Diagram Update</i> Data User .....	72






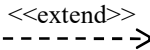
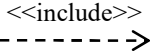
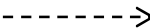

Gambar 3.16 <i>Activity Diagram</i> Hapus Data User .....	73
Gambar 3.17 <i>Activity Diagram</i> Monitor Peta <i>Work From Home</i> .....	73
Gambar 3.18 <i>Activity Diagram</i> Cetak Laporan Presensi .....	74
Gambar 3.19 <i>Activity Diagram</i> Presensi Masuk .....	74
Gambar 3.20 <i>Activity Diagram</i> Presensi Pulang .....	75
Gambar 3.21 <i>Activity Diagram</i> Lihat Kartu Presensi .....	76
Gambar 3.22 <i>Activity Diagram</i> Logout .....	77
Gambar 3.23 <i>Sequence Diagram</i> Login .....	78
Gambar 3.24 <i>Sequence Diagram</i> Tambah Data Karyawan .....	78
Gambar 3.25 <i>Sequence Diagram</i> Update Data Karyawan .....	79
Gambar 3.26 <i>Sequence Diagram</i> Hapus Data Karyawan .....	79
Gambar 3.27 <i>Sequence Diagram</i> Tambah Data Divisi .....	80
Gambar 3.28 <i>Sequence Diagram</i> Update Data Divisi .....	80
Gambar 3.29 <i>Sequence Diagram</i> Hapus Data Divisi .....	81
Gambar 3.30 <i>Sequence Diagram</i> Tambah Data Presensi .....	81
Gambar 3.31 <i>Sequence Diagram</i> Update Data Presensi .....	82
Gambar 3.32 <i>Sequence Diagram</i> Hapus Data Presensi .....	82
Gambar 3.33 <i>Activity Diagram</i> Tambah Data User .....	83
Gambar 3.34 <i>Sequence Diagram</i> Update Data User .....	83
Gambar 3.35 <i>Sequence Diagram</i> Hapus Data User .....	84


Gambar 3.36 <i>Sequence Diagram</i> Monitor Peta <i>Work From Home</i> .....	84
Gambar 3.37 <i>Sequence Diagram</i> Cetak Laporan Presensi .....	85
Gambar 3.38 <i>Sequence Diagram</i> Presensi Masuk .....	85
Gambar 3.39 <i>Sequence Diagram</i> Presensi Pulang .....	86
Gambar 3.40 <i>Sequence Diagram</i> Lihat Kartu Presensi.....	86
Gambar 3.41 <i>Sequence Diagram</i> Logout .....	87
Gambar 3.42 <i>Class Diagram</i> .....	88
Gambar 3.43 Struktur menu antarmuka <i>web</i> .....	91
Gambar 3.44 Struktur menu antarmuka ponsel <i>android</i> .....	92
Gambar 3.45 Perancangan tampilan halaman <i>login</i> .....	92
Gambar 3.46 Perancangan tampilan halaman utama .....	93
Gambar 3.47 Perancangan tampilan menu master karyawan.....	93
Gambar 3.48 Perancangan tampilan <i>form</i> tambah data karyawan .....	94
Gambar 3.49 Perancangan tampilan <i>form update</i> karyawan.....	94
Gambar 3.50 Perancangan tampilan menu master divisi .....	95
Gambar 3.51 Perancangan tampilan <i>form</i> tambah divisi .....	95
Gambar 3.52 Perancangan tampilan <i>form update</i> divisi .....	96
Gambar 3.53 Perancangan tampilan menu master presensi .....	96
Gambar 3.54 Perancangan tampilan <i>form</i> tambah presensi .....	97
Gambar 3.55 Perancangan Tampilan <i>Form Update</i> Presensi .....	97

Gambar 3.56 Perancangan tampilan menu master <i>user</i> .....	98
Gambar 3.57 Perancangan tampilan <i>form</i> tambah <i>user</i> .....	98
Gambar 3.58 Perancangan tampilan form <i>update user</i> .....	99
Gambar 3.59 Perancangan peta sebaran <i>work from home</i> .....	99
Gambar 3.60 Perancangan tampilan <i>scan</i> presensi .....	100
Gambar 3.61 Perancangan tampilan menu utama aplikasi presensi .....	100
Gambar 3.62 Perancangan tampilan menu <i>scan</i> presensi .....	101
Gambar 3.63 Perancangan menu kartu presensi .....	101
Gambar 4.1 Struktur tabel tb_karyawan.....	102
Gambar 4.2 Struktur tabel tb_divisi .....	103
Gambar 4.3 Struktur tabel tb_presensi .....	103
Gambar 4.4 Struktur tabel tb_user .....	103
Gambar 4.5 Tampilan <i>login</i> .....	104
Gambar 4.6 Tampilan halaman utama .....	104
Gambar 4.7 Tampilan menu master karyawan .....	105
Gambar 4.8 Tampilan <i>form</i> tambah data karyawan .....	105
Gambar 4.9 Tampilan <i>Form update</i> Karyawan .....	106
Gambar 4.10 Tampilan menu master divisi .....	106
Gambar 4.11 Tampilan <i>form</i> tambah divisi .....	107
Gambar 3.12 Tampilan <i>form update</i> divisi .....	107

Gambar 4.13 Tampilan menu master presensi .....	108
Gambar 4.14 Tampilan <i>form</i> tambah presensi .....	108
Gambar 4.15 Tampilan <i>form update</i> presensi .....	109
Gambar 4.16 Tampilan menu master <i>user</i> .....	109
Gambar 4.17 Tampilan <i>form</i> tambah <i>user</i> .....	110
Gambar 4.18 Tampilan <i>form update user</i> .....	110
Gambar 4.19 Tampilan peta sebaran <i>work from home</i> .....	111
Gambar 4.20 Tampilan halaman <i>scan</i> presensi .....	111
Gambar 4.21 Tampilan menu utama aplikasi presensi.....	112
Gambar 4.22 Tampilan menu <i>scan</i> presensi .....	112
Gambar 4.23 Tampilan menu kartu presensi.....	113

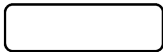


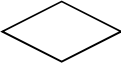

### DAFTAR SIMBOL DIAGRAM *USE CASE*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi meskipun simbol dari aktor ialah gambar orang, tapi aktor belum tentu merupakan orang. Biasanya penamaan aktor dinamakan menggunakan kata benda di awal frase nama aktor.
3		Asosiasi/ <i>Associaton</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
4		<i>Extends</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri meski tanpa <i>use case</i> tambahan itu.
5		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan membutuhkan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri ( <i>independent</i> ).
7		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu merupakan fungsi yang lebih umum dari lainnya

8		<i>System Boundary</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
---	---	------------------------	--

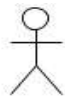

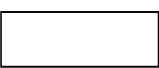
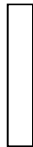
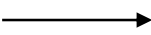
(Sumber: Rosa dan Salahudin, 2018)

### DAFTAR SIMBOL DIAGRAM *ACTIVITY*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Aktivitas yang dilakukan system
2		<i>Initial Node</i>	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
3		<i>Final Node</i>	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
4		<i>Decision / Percabangan</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
5		<i>Fork / Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

(Sumber: Rosa dan Salahudin, 2018)

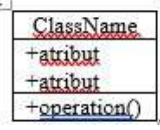


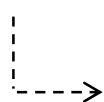
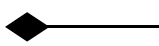
### DAFTAR SIMBOL DIAGRAM *SEQUENCE*

NO	GAMBAR	NAMA	KETERANGAN
1		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi meskipun simbol dari aktor ialah gambar orang, tapi aktor belum tentu merupakan orang. Biasanya penamaan aktor dinamakan menggunakan kata benda di awal frase nama aktor.
2		<i>Lifeline</i>	Menyatakan kehidupan suatu objek
3		<i>Object</i>	Menyatakan objek yang berinteraksi dengan pesan
4		<i>Activation</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi dengan pesan.
5		<i>Message</i>	Mengindikasikan komunikasi antara objek

(Sumber: Rosa dan Salahudin, 2018)



## DAFTAR SIMBOL DIAGRAM CLASS

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Class</i>	Blok pembangun pada pemrograman berorientasi objek. Bagian atas adalah bagian dari <i>class</i> . Bagian tengah mendefinisikan <i>property</i> /atribut <i>class</i> . Bagian akhir mendefinisikan <i>method</i> dari sebuah <i>class</i> .
2		<i>Asosiasi/Associaton</i>	<i>Relationship</i> paling umum antara 2 <i>class</i> , dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hokum-hukum multiplisitas pada sebuah <i>relationship</i> .
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada diatas obejk induk ( <i>ancestor</i> ).
4		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempegaruhi elemen yang bergantung padanya
5		<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>composition</i> terhadap <i>class</i> tempat dia bergantung tersebut.

(Sumber: Rosa dan Salahudin, 2018)

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Pada era digital dimasa kini perkembangan teknologi informasi terjadi begitu pesat, setiap individu berlomba-lomba menciptakan teknologi baru karena adanya tuntutan dari kebutuhan setiap individu untuk dapat memfasilitasi setiap aktifitas, proses, ataupun kegiatan-kegiatan dengan lebih mudah dan praktis dan dapat meminimalisir kesalahan yang dilakukan oleh manusia atau lebih sering dikenal dengan *human error*.

Didalam suatu instansi atau perusahaan terdapat istilah yang dikenal dengan sistem presensi yang berfungsi sebagai pencatatan kehadiran karyawan. Adapun peran dari sistem presensi ini merupakan hal paling penting bagi suatu instansi atau perusahaan untuk mengetahui tingkat prestasi kehadiran dan kedisiplinan dari setiap karyawannya. Tentunya apabila sistem presensi ini masih dilakukan dengan cara-cara yang bersifat semi *Electronnic Data Processing (EDP)*, kebutuhan perusahaan untuk mengetahui tingkat prestasi kehadiran dan kedisiplinan dari setiap karyawan tentunya tidak akan maksimal karena adanya celah kecurangan pencatatan kehadiran yang dilakukan oleh karyawan dan dapat merugikan perusahaan.

CV. Indoscots Baby Utama adalah perusahaan milik swasta yang bergerak pada bidang industri produksi aksesoris dan perlengkapan bayi dengan kualitas produk yang dihadirkan bervariasi, menarik, dan elegan.

Setelah penulis menemukan permasalahan di CV. Indoscots Baby Utama masih terdapat beberapa kendala yang berkaitan dengan sistem presensi, diantaranya adalah kegiatan pengisian formulir (kertas) presensi ketika jam masuk dan jam pulang dilakukan sendiri oleh karyawan, sehingga bisa saja karyawan mengisi jam masuk dan jam pulang yang tidak sesuai dengan kenyataan semisal keterlambatan ketika tiba di kantor namun pada formulir presensi diisi tepat waktu. Selain itu, terhitung sejak bulan maret 2020 CV. Indoscots Baby Utama menjalankan kebijakan bekerja dari rumah atau dikenal dengan istilah *work from home*, kebijakan ini memiliki kekurangan dimana perusahaan tidak mengetahui secara pasti dimana karyawannya melakukan pekerjaan karena tidak adanya informasi keberadaan atau lokasi dimana karyawan tersebut melakukan kebijakan *work from home* yang diberikan oleh perusahaan.

Mengikuti perkembangan zaman pada era digital ini, perangkat ponsel *android* sudah menjadi kebutuhan keseharian bagi penggunanya. Dengan kemudahan dalam pengoperasiannya, dan fitur-fitur semacam kamera, penjelajah *website*, fitur *Global Positioning System (GPS)*, dan lain sebagainya membuat ponsel android banyak digunakan dalam membantu aktifitas penggunanya. Fitur kamera pada ponsel android tidak hanya berfungsi sebagai pengambil gambar atau perekam video saja, namun dapat juga dipakai untuk memindai kode-kode unik pada kode batang ataupun kode-qr (*Quick Response Code*).

Berdasarkan pada beberapa permasalahan diatas, maka dari itu penulis mengambil sebuah judul laporan kerja praktek tentang “**Perancangan Sistem**

**Presensi *Quick Response Code* Berbasis *Web* dan *Android Client* CV. Indoscots Baby Utama”.**

## **1.2 Identifikasi Masalah**

Berdasarkan latar belakang yang telah disampaikan diatas, maka penulis menyimpulkan beberapa identifikasi masalah sebagai berikut :

1. Sistem presensi tulis tangan yang dilakukan oleh karyawan CV. Indoscots Baby Utama memiliki resiko kecurangan dalam pengisian data kehadiran pada formulir presensi.
2. Keberadaan karyawan yang melakukan kebijakan *work from home* tidak dapat diketahui oleh pihak perusahaan karena tidak adanya sistem yang dapat memberikan informasi keberadaan karyawan yang sedang menjalankan kebijakan tersebut.
3. Proses sistem pencatatan kehadiran karyawan masih dilakukan secara semi *Electronic Data Processing*.

## **1.3 Batasan Masalah**

Berdasarkan idetifikasi masalah diatas, maka penulis membatasi masalah tersebut sebagai berikut :

1. Kecurangan pengisian data kehadiran pada formulir presensi oleh karyawan.
2. Informasi keberadaan karyawan yang sedang menjalankan kebijakan *work from home* tidak diketahui oleh pihak perusahaan.
3. Sistem presensi yang sedang berjalan belum terkomputerisasi secara menyeluruh.

## **1.4 Tujuan dan Kegunaan Penelitian**

### **1.4.1 Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah :

1. Sebagai salah satu sarana untuk mempermudah kegiatan pencatatan kehadiran di lingkungan perusahaan CV. Indoscots Baby Utama.
2. Sebagai alat untuk melakukan monitor lokasi karyawan yang sedang melaksanakan kebijakan *Work From Home* (bekerja dari rumah).
3. Menjadikan ponsel *Android* sebagai alat presensi yang praktis.

### **1.4.2 Kegunaan Penelitian**

Adapun kegunaan dari penelitian ini adalah sebagai berikut :

1. Bagi Penulis

Untuk mengimpelementasikan ilmu yang didapat saat kuliah dalam menganalisa dan merancang sebuah sistem perangkat lunak.

2. Bagi Pengguna

Untuk menjadikan kegiatan presensi di CV. Indoscots Baby Utama berjalan dengan tepat, akurat, dan dapat digunakan dengan praktis.

3. Bagi Lembaga

Untuk memenuhi salah satu syarat kelulusan di STMIK Mardira Indonesia Mata Kuliah Kerja Praktek pada Program Studi Teknik Informatika Jenjang S1 (Sarjana).

## **1.5 Objek dan Waktu Penelitian**

Adapun yang menjadi objek penelitian oleh penulis adalah karyawan pengguna perangkat *smartphone* Android di CV. Indoscot Baby Utama dan waktu penelitian ini dilakukan pada bulan Oktober 2020 hingga Desember 2020.

## **1.6 Metodologi Penelitian**

Untuk menyelesaikan masalah dengan tepat, sistematis dan logis maka digunakan metodologi penelitian.

### **1.6.1 Metode Penelitian**

Metode penelitian yang digunakan adalah metode penelitian deskriptif dimana penelitian yang ditujukan untuk menggambarkan fenomena-fenomena yang ada, yang berlangsung saat ini atau saat yang lampau (Sujana dan Ibrahim dalam Rahima, 2016:130).

Berkaitan dengan penelitian untuk mendapatkan data dan informasi, maka metode yang digunakan dalam proses pengumpulan data adalah sebagai berikut :

#### **1. Observasi**

Melakukan penelitian secara langsung pada objek yang diteliti dengan mengadakan pengamatan dan pencatatan terhadap data dan informasi yang berkaitan dengan penelitian.

#### **2. Wawancara**

Salah satu metode atau cara yang digunakan untuk memperoleh data untuk mengajukan pertanyaan-pertanyaan dengan pihak yang terkait.

### 3. Studi pustaka

Teknik ini digunakan untuk mengkaji permasalahan melalui analisa literatur seperti buku, jurnal, dan media internet.

### 4. Dokumentasi

Merupakan teknik pengumpulan data dengan menghimpun dan menganalisa dokumen-dokumen, baik dokumen tertulis, gambar, maupun elektronik.

## 1.6.2 Metode Pengembangan Sistem

Adapun metode pengembangan sistem yang digunakan dalam penelitian ini adalah *Object Oriented Analysis and Design (OOAD)*. *OOAD* adalah metode pengembangan sistem yang lebih menekankan pada objek dibandingkan dengan data atau proses. Ada beberapa ciri khas dari pendekatan ini yaitu *object*, *Inheritance*, dan *object class*.

*Object* adalah struktur yang mengkapsulasi atribut dan metode yang beroperasi berdasarkan atribut-atribut tadi. Objek adalah abstraksi dari benda nyata dimana data dan proses diletakkan bersama untuk memodelkan struktur dan perilaku objek dunia nyata. *Object class* adalah sekumpulan objek yang berbagi struktur yang sama dan perilaku yang sama.

*Inheritance* properti yang muncul ketika tipe entitas atau *object class* disusun secara hirarki dan setiap tipe entitas atau *objek class* menerima atau mewarisi atribut dan metode dari pendahulunya.

Tahap-tahap pada metode OOAD ini adalah sebagai berikut:

*a. Inception*

Pada tahapan awal ini ialah membangun dan memprioritaskan kebutuhan-kebutuhan sistem, membuat sebuah kesepakatan dengan *customer* mengenai apa yang akan dibangun, secara pasti dapat mengetahui kendala yang terjadi dan memastikan lingkup lingkungan pembangunan sistem yang akan dibangun.

*b. Elaboration*

Pada tahapan ini akan dibangun rancangan sistem yang akan dibuat. Seluruh kebutuhan customer digambarkan melalui desain diagram.

*c. Construction*

Pada tahapan ini merupakan tahapan pembangunan atau pembuatan *coding* serta pengujian sistem yang akan dibuat.

*d. Transition*

Pada tahapan ini sistem telah selesai dibuat dan akan siap diberikan kepada *end user*.

Pemodelan pengembangan sistem akan digambarkan menggunakan metode pemodelan *UML (Unified Modeling Language)*. Beberapa diagram *UML* diantaranya adalah:

1. *Use case diagram*
2. *Activity diagram*
3. *Sequence Diagram*
4. *Class Diagram*



## 1.7 Sistematika Penulisan

Untuk mempermudah dan memahami isi dari penulisan, maka penulis membuat sistematika penulisan sebagai berikut :

### **BAB I PENDAHULUAN**

Bab ini memaparkan tentang latar belakang masalah, identifikasi masalah, batasan masalah, tujuan dan kegunaan penelitian, metodologi penelitian dan sistematika penulisan.

### **BAB II LANDASAN TEORI**

Bab ini menjelaskan tentang teori – teori umum yang berkaitan dengan konsep dasar sistem, sistem presensi, teori *web*, kode qr, basis data, dan *android*.

### **BAB III ANALISIS DAN PERANCANGAN APLIKASI**

Bab ini menjelaskan mengenai gambaran instansi tempat penelitian yang didalamnya mencakup nama instansi, alamat, sejarah singkat instansi, *job desk*, dan sistem yang sedang berjalan pada instansi tersebut. Selain itu, bab ini juga berisi tentang desain sistem yang diusulkan berupa perancangan sistem yang digambarkan menggunakan diagram-diagram *Unfied Modelling Language (UML)*, struktur basis data, pengkodean, serta perancangan antarmuka.

### **BAB IV IMPLEMENTASI**

Bab ini berisikan uraian tentang pengimplementasian sistem, tampilan dan hasil dari sistem yang dirancang.

## **BAB V KESIMPULAN DAN SARAN**

Bab ini penulis memberikan beberapa kesimpulan dan saran sebagai akhir penulisan.

## BAB II

### LANDASAN TEORI

Untuk mendukung penulisan laporan kerja praktek ini, maka perlu dikemukakan hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam pembuatan laporan ini.

#### 2.1 Definisi Perancangan

Menurut Nur dan Suyuti (2018:5), perancangan merupakan suatu proses yang bertujuan untuk menganalisis, menilai, memperbaiki, dan menyusun suatu sistem, baik sistem fisik maupun non fisik yang optimum untuk waktu yang akan datang dengan memanfaatkan informasi yang ada. .

Sedangkan perancangan menurut Kusri (dalam Nur dan Suyuti, 2018:5), adalah proses pengembangan spesifikasi sistem baru berdasarkan hasil rekomendasi analisis sistem.

Berdasarkan beberapa pendapat diatas, maka dapat disimpulkan bahwa perancangan merupakan suatu proses yang dilakukan untuk mendesain atau membuat sistem baru berdasarkan hasil rekomendasi dari analisis sistem.

#### 2.2 Definisi Sistem

Sistem menurut Kadir (dalam Muslihudin dan Oktafianto, 2016:2), dapat didefinisikan sebagai sekumpulan elemen yang saling terkait atau terpadu yang dimaksud untuk mencapai suatu tujuan.

**Commented [S1]:** Referensi : Googlebooks  
Judul : " Perancangan mesin-mesin industri"  
Halaman : 5  
Penulis : Nur dan Suyuti

**Commented [S2]:** Referensi : Googlebooks  
Judul : " Perancangan mesin-mesin industri"  
Halaman : 5  
Penulis : Nur dan Suyuti

**Commented [S3]:**  
Referensi : Googlebooks  
Judul : "Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML "  
Halaman : 2  
Penulis : Oktafianto

Sistem menurut Indra (dalam Muslihudin dan Oktafianto, 2016:2), dapat didefinisikan sebagai sekumpulan elemen atau subsistem yang saling berhubungan satu dengan yang lain membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan .

Berdasarkan definisi diatas, maka dapat disimpulkan bahwa sistem merupakan kumpulan dari beberapa bagian yang saling berkaitan dan berfungsi bersama-sama untuk mencapai satu tujuan yang sama.

### 2.2.1 Karakteristik Sistem

Menurut Jogianto (dalam Muslihudin dan Oktafianto, 2016:4), sesuatu dapat dikatakan sebagai sistem jika memiliki beberapa sifat atau karakteristik tertentu. Adapun sifat atau karakteristik yang dimaksud adalah :

1. Komponen sistem, ialah sistem yang terdiri atas bagian-bagian yang saling berkaitan dan bervariasi yang bersama-sama mencapai beberapa sasaran. Sebuah sistem bukanlah seperangkat unsur yang tersusun secara teratur, tetapi terdiri atas unsur yang dapat dikenal dan saling melengkapi karena suatu maksud, tujuan dan sasaran.
2. Batasan sistem (*Boundary*), merupakan daerah yang membatasi antara sistem yang satu dengan sistem yang lainnya atau dengan lingkungan luarnya.
3. Lingkungan luar sistem (*Environment*), adalah apapun diluar dari batasan sistem yang mempengaruhi operasi sistem. Lingkungan luar dapat merugikan sistem tersebut. Lingkungan luar yang menguntungkan berupa energi dari sistem, sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, jika tidak akan mengganggu kelangsungan sistem tersebut.

**Commented [S4]:** Referensi : Googlebooks  
Judul : "Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML "  
Halaman : 2  
Penulis : Oktafianto

**Commented [S5]:** Referensi : Googlebooks  
Judul : "Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML "  
Halaman : 4  
Penulis : Oktafianto

4. Sistem penghubung (*interface*), merupakan media penghubung antara suatu subsistem dengan yang lainnya. Penghubung ini memungkinkan sumber daya mengalir dari suatu sistem ke sistem yang lainnya dengan melalui penghubung suatu subsistem dapat berinteraksi dengan subsistem lainnya membentuk suatu kesatuan.
5. Sistem masukan (*input*), adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal. Masukan perawatan adalah energi yang dimasukkan supaya sistem dapat beroperasi. Sedangkan masukan sinyal adalah energi yang diproses untuk mendapatkan keluaran.
6. Sistem keluaran, adalah energi yang diolah, diklasifikasikan menjadi keluaran yang berguna untuk subsistem lain.
7. Sistem sasaran, ialah suatu sistem yang mempunyai tujuan atau sasaran. Jika suatu sistem yang mempunyai batasan sasaran, maka operasi sistem tidak akan ada gunanya.

### 2.3 Definisi Presensi

Presensi menurut Mulyana (dalam Khoiriyah dkk, 2018:54), dapat didefinisikan sebagai pencatatan dan pengolahan data kehadiran yang dilakukan secara terus menerus, pencatatan dilakukan setiap hari jam kerja dan dilakukan pelaporan kepada pihak HRD / Manager Perusahaan.

### 2.4 Definisi Sistem Presensi

Menurut Hidayat dan Wagya (2019:279), sistem presensi dapat didefinisikan sebagai suatu kegiatan yang dilakukan oleh karyawan suatu

**Commented [S6]:** Referensi : Jurnal Scholar  
 Nama Jurnal : Jurnal Informatika  
 Judul : "Rancangan Bangun Presensi Online Berbasis Granteed Validitas Data "  
 Halaman : 54  
 Penulis : Khoiriyah dkk

**Commented [S7]:** Referensi : Jurnal Scholar  
 Nama Jurnal : Jurnal Ilmiah Setrum  
 Judul : "Rancangan Bangun Sistem Presensi Menggunakan Face Recognition dengan Metode Eigenface "  
 Halaman : 279  
 Penulis : Hidayat dan Wagya

perusahaan untuk membuktikan bahwa dirinya hadir dalam melaksanakan pekerjaan.

Adapun sumber yang menjelaskan secara spesifik adalah menurut Purnama (dalam Khoiriyah dkk, 2018:54) yang menyatakan bahwa :

“Sistem presensi adalah sistem manajemen kehadiran personal suatu lembaga/instansi yang secara otomatis mencatat data kehadiran dan dapat digunakan sebagai sumber laporan untuk kebutuhan manajemen personal “.

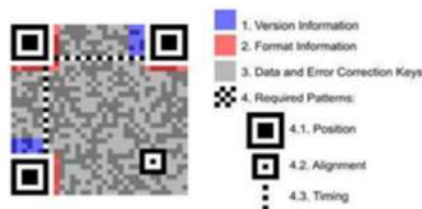
## 2.5 Pengertian *Quick Response Code (QR-Code)*

Yudhanto dan Aziz (2019:53), menyatakan bahwa :

“*qr-code* merupakan singkatan dari *quick response* atau respon cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respon yang cepat pula, berbeda dengan kode batang, yang hanya menyimpan informasi secara horizontal, *qr-code* mampu menyimpan informasi secara horizontal dan vertikal ”.

### 2.5.1 Struktur *Quick Response Code (QR-Code)*

Menurut Yudhanto dan Aziz (2019:55), struktur *Quick Response Code (QR-Code)* terdiri dari beberapa bagian, adapun beberapa bagian yang dimaksud adalah sebagai berikut :



Gambar 2.1 Struktur *qr-code*  
(Sumber : Yudhanto dan Azis, 2019:55)

**Commented [S8]:** Referensi : Jurnal Scholar  
Nama Jurnal : Jurnal Informatika  
Judul : “Rancangan Bangun Presensi Online Berbasis Granteed Validitas Data “  
Halaman : 54  
Penulis : Khoiriyah dkk

**Commented [S9]:** Referensi : Googlebooks  
Judul : “Pengantar Teknologi Internet of Things (IoT) “  
Halaman : 53  
Penulis : Yudhanto dan Aziz

**Commented [S10]:** Referensi : Googlebooks  
Judul : “Pengantar Teknologi Internet of Things (IoT) “  
Halaman : 55  
Penulis : Yudhanto dan Aziz

1. *Version Information* atau versi informasi mengenai *QR-Code* sendiri, versi dan ukuran.
2. *Format Information* atau format informasi kesalahan mengkoreksi pola tingkat dan masker (dijelaskan kemudian) di simpan.
3. *Data and Error Correction Keys* adalah dimana data yang di kodekan disimpan, pola koreksi kesalahan ini juga tertanam ke dalam data.
4. *Position* atau posisi digunakan untuk membantu posisi kode pada saat *decoding*.
5. *Alignment* atau penyalaras pola digunakan untuk mencari sudut yang tepat saat *decoding*.
6. *Timing* atau pola waktu digunakan untuk membantu menentukan simbol kordinat dalam aplikasi *decoder*.

### 2.5.2 Karakteristik *Quick Response Code (QR-Code)*

Menurut Yudhanto dan Azis (2019:56), karakteristik dari kode *qr* adalah dapat menampung jumlah data yang besar. Sebanyak 7089 karakter numerik maksimum data dapat tersimpan di dalamnya, kerapatan tinggi (100 kali lebih tinggi dari kode simbol linier) dan pembacaan kode dengan cepat. Kode *qr* juga memiliki kelebihan lebih baik dalam hal unjuk kerja dan fungsi. Adapun kelebihan yang dimaksud adalah :

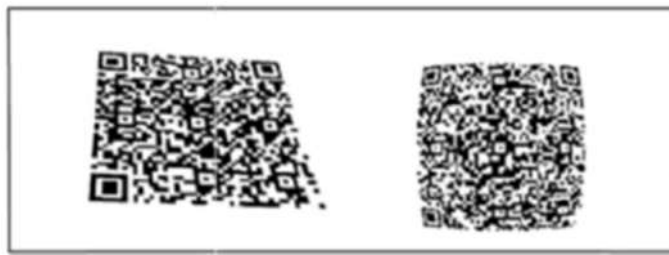
1. Pembacaan data dari segala arah (360 derajat)

Pembacaan kode matriks dengan menggunakan sensor kamera CCD (*Charged Coupled Device*) dimana data akan memindai baris perbaris dari citra yang ditangkap dan kemudian disimpan dalam memori.

**Commented [S11]:** Referensi : Googlebooks  
Judul : "Pengantar Teknologi Internet of Things (IoT) "  
Halaman : 56  
Penulis : Yudhanto dan Aziz

## 2. Ketahanan terhadap penyimpangan *symbol*

Simbol matriks dua dimensi akan rentan terhadap penyimpangan bentuk ketika di tempatkan pada permukaan yang tidak rata (bergelombang) sehingga sensor pembaca menjadi miring karena sudut antara sensor CCD dan simbol matriks dua dimensi ini telah berubah. Untuk memperbaiki penyimpangan ini, *qr-code* memiliki perata pola (*alignment pattern*) yang menyusun dengan jarak teratur dalam suatu daerah. *Alignment pattern* akan memperhitungkan titik pusat dengan daerah terluar dari simbol matriks, sehingga dengan cara ini penyimpangan linear maupun non linear masih terbaca.



Gambar 2.2 Pembacaan *qr-code* dari segala arah  
(Sumber : Yudhanto dan Azis, 2019:57)

## 3. Fungsi pemulihan data (ketahanan terhadap kotoran maupun kerusakan)

Dalam mengendalikan kerusakan yang diakibatkan kotor maupun usak, *qr-code* memanfaatkan algoritma Reed Salomon yang tahan terhadap kerusakan tingkat tinggi. Jadi, ketika *qr-code* akan digunakan dalam lingkungan yang rawan kerusakan akibat lingkungan, disarankan menggunakan koreksi *error* 3%.





Gambar 2.3 Kerusakan pada *QR-Code*  
(Sumber : Yudhanto dan Azis, 2019:58)

4. Kemampuan *encode* karakter kanji dan kana Jepang

*Qr-code* berkembang pesat di negara Jepang. Hal ini menyebabkan perkembangan *qr-code* untuk dapat menerima input data berupa karakter yang non alfabetis. Ketika pembuatan *qr-code* dengan inputan berupa huruf Jepang, maka data tersebut akan diubah kedalam bentuk *binner* 16 bit (2 *byte*) untuk karakter tunggal sedangkan untuk gabungan karakter akan di *encode* ke dalam *binner* 13 bit.

5. Fungsi *linking* pada *symbol*

*Qr-code* memiliki kemampuan dapat dipecah menjadi beberapa bagian dengan maksimum pembagian adalah 16 bagian. Dengan fungsi *Linking* ini maka *qr-code* dicetak pada daerah yang tidak terlalu luas untuk sebuah *qr-code* tunggal.

6. Proses *Masking*

Proses *Masking* pada *qr-code* berperan sangat penting dalam hal penyusunan modul hitam dan modul putih agar memiliki jumlah yang seimbang. Untuk memungkinkan hal ini digunakan operasi *XOR* yang diaplikasikan diantara area data dan daerah *mask pattern*. Ada sebanyak delapan *mask pattern* dalam *qr-code* yang kesemuanya itu dalam bentuk *binner* tiga bit.

### 2.5.3 Alat pemindai *Quick Response Code (QR-Code)*

Sugiantoro dan Hasan (2025:134), agar dapat membaca *qr-code* atau kode *qr* maka diperlukan sebuah alat pembaca atau pemindai berupa *software* yaitu *QR-Code Reader* atau *QR-Code Scanner* yang harus di *install* pada perangkat telepon *mobile*.

**Commented [S12]:** Referensi : Jurnal Scholar  
Nama Jurnal : Jurnal Telematika  
Judul : "PENGEMBANGAN QR CODE SCANNER BERBASIS ANDROID UNTUK SISTEM INFORMASI MUSEUM SONOBUDOYO"  
Halaman : 134  
Penulis : Sugiantoro dan Hasan

### 2.6 Pengertian *Android*

*Android* pertama kali dikembangkan oleh perusahaan kecil di *Silicon Valley* yang bernama *Android Inc.* Pada tahun 2005, sistem operasi tersebut diambil alih oleh *Google* dan menjadikan sistem operasi tersebut bersifat "*Open Source*" sehingga siapapun dapat menggunakannya dengan gratis, termasuk penggunaan kode sumber yang digunakan dalam sistem operasi tersebut.

Menurut *Google Developer Training Team* (2016:5), "*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 berbeda".

Menurut Kadir (2017:2), "*Android* adalah nama sistem operasi yang juga bersifat '*Open Source*' yang umumnya ditujukan untuk *smartphone* dan tablet".

Menurut Herlinah dan Musliadi (2019:1), "*Android* merupakan sistem operasi yang banyak digunakan pada perangkat bergerak yang dewasa ini sangat terkenal dan populer digunakan pada ponsel cerdas".

Dari beberapa pernyataan diatas maka dapat disimpulkan bahwa *Android* adalah sistem operasi bersifat *open source* untuk perangkat *mobile* seperti *smartphone* atau tablet yang dikembangkan oleh *Google*.

### 2.6.1 Versi Sistem *Android*

Menurut Takdirillah (2020), hingga tahun 2020 ini setidaknya sudah ada 17 versi pembaruan sistem *Android*, diantaranya:

1. *Android 1.0 Alpha*

Versi *Android 1.0 Alpha* ini pertama kali dirilis pada September 2008 dan tidak ditujukan untuk kebutuhan komersial. Walaupun belum dirilis secara komersial, versi *Android* ini telah dilengkapi fitur dukungan akses *web browser*, *streaming youtube*, pemutar media, *google map*, dan sinkronisasi dengan aplikasi *google* lainnya.

2. *Android 1.1 Beta*

Selang beberapa bulan versi *Android 1.1 Beta* diluncurkan pada tanggal 9 Februari 2009. Sama seperti pendahulunya yakni *Alpha* versi *Android* belum dirilis secara komersial, hanya diperuntukan untuk satu perangkat. Pembaruan ini dilakukan untuk memperbaiki *bugs* dan meningkatkan beberapa fitur seperti rincian lokasi pada aplikasi *maps* serta fitur menyembunyikan dan menampilkan tombol panggilan.

3. *Android 1.5 Cupcake*

Barulah pada tanggal 30 April 2009 *Android* versi *Android 1.5 Cupcake* yang diperkenalkan secara komersial. Versi *Cupcake* merupakan generasi pertama yang dirilis secara komersial dan berawal dari sinilah *Android* mulai menggunakan nama makanan manis kepada setiap versi *Android* yang dirilisnya. Beberapa fitur yang dimiliki *Android Cupcake* seperti dukungan akan rotasi layar otomatis, *widget*, dan *keyboard virtual*.

#### 4. *Android 1.6 Donut*

Versi pembaruan *Android* berikutnya adalah versi 1.6 *Donut* yang dirilis pada tanggal 15 September 2009. Pada versi *Donut* ini *Android* menambahkan beberapa fitur seperti persentase daya baterai, dukungan gestur, fasilitas pencarian di *Android market* atau yang sekarang kita kenal dengan *play store*.

#### 5. *Android 2.0 Eclair*

Selang sebulan *Android* melakukan pembaruan kembali untuk memperbaiki *bug* dan memperkaya fiturnya tepatnya pada tanggal 26 Oktober 2009. Pembaruan *Android versi 2.0* diberi nama *Eclair* dan pada versi ini terdapat beberapa fitur seperti *multi touch*, *live wallpaper*, perubahan tampilan antarmuka dan dukungan *browser* untuk *HTML5*.

#### 6. *Android 2.2 Froyo*

Pada tanggal 20 Mei 2010 *Android* kembali meluncurkan versi terbarunya yaitu versi 2.2 *Froyo*. Pada versi *Froyo* ini *Android* sudah mulai dikenal luas oleh *vendor* atau pabrikan ponsel. Pembaruan *Android 2.2* membawa beberapa fitur unggulan seperti memperbesar gambar pada galeri dengan gestur, peningkatan fitur *USB tethering* dan *hotspot WIFI* serta dukungan animasi *GIF* pada *web browser*.

#### 7. *Android 2.3 Gingerbread*

Pada versi ini tidak bisa diragukan lagi *Android* telah menjadi sistem operasi *mobile* yang populer. Kerjasama dengan pabrikan Samsung dalam membuat produk Samsung Galaxy Series semakin menambah kepopuleran *Android*. Versi 2.3 *Gingerbread* dirilis pada tanggal 6 Desember 2010 dengan

menambah beberapa fitur seperti dukungan *NFC*, fitur *copy* atau *paste* dengan memilih kata melalui layar yang ditekan serta dukungan beberapa sensor lainnya.

8. *Android 3.0 Honeycomb*

Versi *Android* berikutnya adalah versi *3.0 Honeycomb* yang diperkenalkan pada tanggal 22 Februari 2011. Versi *Android* ini dikhususkan untuk perangkat *tablet PC*, dengan membawa beberapa fitur yang ditujukan untuk komputasi bisnis pada tablet. Fitur yang dimiliki *Android Honeycomb* seperti dukungan *prosesor multi core*, dukungan obrolan video dengan *Google Talk* dan percepatan saat berpindah aplikasi yang sedang berjalan dengan fitur *multitasking recent apps*.

9. *Android 4.0 Ice Cream Sandwich*

Tepat pada tanggal 19 Oktober 2011 *Android* kembali meluncurkan versi terbarunya *4.0 Ice Cream Sandwich*. Versi ini membawa fitur yang dimiliki oleh *Honeycomb* untuk bisa berjalan pada *smartphone* yang sebelumnya hanya ditujukan untuk *tablet PC*. Selain fitur yang sebelumnya ada di *Honeycomb*, versi 4.0 juga menambahkan beberapa fitur lain seperti perbaikan antarmuka dan kostumisasi *widget*.

10. *Android 4.1 Jelly Bean*

*Android 4.1 Jelly Bean* dirilis pada tanggal 27 Juni 2012. Pengembangan versi *Jelly Bean* lebih berfokus terhadap peningkatan performa tampilan antarmuka. Fitur terbaru yang disematkan pada versi *Jelly bean* adalah *keyboard* yang bisa dikostumisasi oleh pengguna dan dukungan gestur pada *keyboard*, *UI* yang

lebih *smooth*, dukungan tampilan *nirkabel*, *widget* yang bisa diatur dan disesuaikan ukurannya.

#### 11. *Android 4.4 Kitkat*

Nama *KitKat* dipilih *Android* untuk digunakan pada versi terbarunya yaitu versi 4.4 yang diresmikan pada tanggal 31 Oktober 2013. Pada versi ini *Android* meningkatkan optimalisasi dengan memberikan fitur yang lebih baik. Beberapa fitur yang dibawa *android KitKat* seperti *WebViews* yang berbasis *Chromium*, pengoptimalan kinerja terhadap perangkat dengan spesifikasi rendah, dukungan sensor *batching* dan *step detector*.

#### 12. *Android 5.0 lollipop*

Versi *Android* selanjutnya adalah versi *5.0 Lollipop* yang dirilis pada tanggal 25 Juni 2014. Pada versi ini *Android* tidak hanya menjadi sistem operasi pada perangkat *smartphone*, namun juga telah berjalan pada perangkat *mobile* lainnya seperti *Android TV* dan juga *Google Fit*. Beberapa fitur yang ditambahkan pada versi ini adalah *user interface* yang mengikuti desain *Google* yaitu *material design* dan fitur *factory reset protection* untuk menjaga *smartphone* agar tidak di *reset* apabila hilang.

#### 13. *Android 6.0 Marshmallow*

*Android 6.0 Marshmallow* di perkenalkan pada tanggal 5 Mei 2015. Fitur yang dibawa oleh versi *Android Marshmallow* adalah dukungan sensor sidik jari untuk mengakses *smartphone*, fasilitas menjalankan beberapa aplikasi pada tata letak layar dengan dukungan *multi window*, dukungan *platform virtual*

*reality*, dan kemampuan dalam mengurangi pemakaian *bandwidth* pada *mode data saver*.

14. *Android 7.0 Nougat*

Diperkenalkan pada tanggal 19 Oktober 2016 *Android 7.0 Nougat* berfokus pada peningkatan performa *user interface* sehingga lebih intuitif dan penggunaan aplikasi secara bersamaan lebih banyak pada fitur *multi window*. Selain peningkatan fitur tadi, *Android Nougat* juga menambahkan beberapa fitur lain seperti dukungan cahaya malam atau mode malam, *keyboard default* yang dapat mengirim animasi *GIF* langsung dan dukungan panggilan *multi-endpoint*.

15. *Android 8.0 Oreo*

Nama *Oreo* dipilih *Android* untuk digunakan pada versi *Android 8.0* yang diluncurkan pada bulan Agustus 2017. *User interface* pada *Android Oreo* lebih simpel agar memudahkan dalam mengakses aplikasi. Pembaruan pada versi *Oreo* membawa beberapa fitur seperti fitur *Autofill* yang memberikan kemudahan dalam mengisi formulir misal, dukungan gambar dalam gambar dan pengoptimalan *booting* agar lebih cepat.

16. *Android 9.0 Pie*

*Android 9.0 Pie* merupakan versi *Android* terbaru yang dirilis pada bulan Agustus 2018. Fitur unggulan yang dimiliki oleh versi *9.0 Pie* ini adalah kemampuan *Artificial Intelligent* atau kecerdasan buatan.

### 17. *Android 10*

Nama *Android* berikutnya yang paling terbaru tidak diberi nama seperti pendahulunya yang berkaitan dengan makanan manis. Versi *Android* terbaru diberi nama *Android 10* untuk memperingati bahwa *Android* telah mencapai 1 dekade secara komersial. Versi *Android 10* lebih berfokus pada penyempurnaan mode malam atau gelap serta peningkatan fitur *sound amplifier* untuk mengatur kualitas audio.

### 2.7 Pengertian *Web*

Abdulloh (2018:1) menyatakan bahwa *web* dapat diartikan sebagai “kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya”.

Hariyanto (dalam Destiningrum & Adrian, 2017 : 32) menyatakan bahwa *web* merupakan kumpulan halaman yang menampilkan informasi teks, data gambar, data animasi, suara, video, dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

Berdasarkan pada pernyataan diatas maka dapat disimpulkan bahwa *website* adalah kumpulan halaman berisi informasi teks, data gambar, suara, data animasi, video dan atau gabungan dari semuanya yang membentuk sebuah rangkaian halaman yang saling terkait satu sama lain.



### 2.7.1 Jenis-Jenis Web

Abdulloh (2018: 1) membagi *web* menjadi 3 jenis, adapun 3 jenis *web* yang dimaksud adalah sebagai berikut :

1. *Website Statis*

*Website* statis yaitu jenis *website* yang isinya tidak diperbaharui secara berkala, sehingga isinya dari waktu akan selalu tetap. *Website* jenis ini biasanya hanya digunakan untuk penampilan profil dari pemilik *website* seperti profil perusahaan atau organisasi.

2. *Website Dinamis*

*Website* dinamis yaitu jenis *website* yang isinya terus diperbaharui secara berkala oleh pengelola *web* atau pemilik *website*. *Website* jenis ini banyak dimiliki oleh perusahaan atau perorangan yang aktifitas bisnisnya memang berkaitan dengan internet. Contoh paling mudah dari *website* jenis ini yaitu *web blog* dan *website* berita.

3. *Website Interaktif*

*Website* interaktif pada dasarnya termasuk dalam kategori *website* dinamis, dimana isi informasinya selalu diperbaharui dari waktu ke waktu. Hanya saja, isi informasi tidak hanya diubah oleh pengelola *website* tetapi lebih banyak dilakukan oleh pengguna *website* itu sendiri. Contoh *website* jenis ini yaitu *website* jejaring sosial seperti *facebook*, dan *twitter* atau *website marketplace* seperti *bukalapak*, *tokopedia*, dan sebagainya.

### 2.7.2 Pemrograman Web

Menurut Abdulloh (2018 : 2), pemrograman *web* dapat diartikan sebagai kegiatan pembuatan program atau aplikasi berbasis *web* menggunakan bahasa pemrograman tertentu sehingga dapat memproses data dan menghasilkan informasi yang dikehendaki oleh pemilik *website*. Adapun bahasa pemrograman yang dimaksud adalah :

1. *Hypertext Markup Language (HTML)*

Berperan sebagai pembentuk struktur halaman *website* yang menempatkan setiap elemen *website* sesuai *layout* yang diinginkan.

2. *Chascading Style (CSS)*

Berperan sebagai pembentuk desain *website* dengan mengatur setiap elemen *HTML* agar tampil menarik pada *browser*.

3. *Hypertext Preprocessor (PHP)*

Berperan sebagai pemroses data pada sisi *server* sesuai yang diminta oleh *client* menjadi informasi yang siap ditampilkan, juga sebagai penghubung aplikasi *web* dengan *database*.

4. *Structure Query Language (SQL)*

Berperan sebagai bahasa yang mengatur transaksi data antara aplikasi dengan *database* sebagai tempat penyimpanan data. *Database* yang biasanya digunakan adalah *MySQL*, *Oracle*, *SQL server* dan sebagainya.

5. *JavaScript*

Berperan sebagai bahasa pemrosesan data pada sisi *client* serta dapat memanipulasi *HTML* dan *CSS* secara dinamis.

## 2.8 Definisi Basis Data (*Database*)

Menurut Rachmadi (2020:2), basis data (*database*) merupakan kumpulan data dalam bentuk *file/table*/arsip yang saling berhubungan dan tersimpan dalam media penyimpanan elektronis, untuk kemudahan dalam pengaturan, pemilahan, pengelompokan dan pengorganisasian.

Menurut Lubis (2016:3), basis data (*database*) adalah tempat berkumpulnya data yang berhubungan dalam suatu wadah (organisasi/perusahaan) bertujuan agar dapat mempermudah dan mempercepat untuk pemanggilan atau pemanfaatan kembali data tersebut.

Menurut Yanto (2016:10), basis data (*database*) dapat di definisikan kedalam tiga sudut pandang yaitu :

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*), untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file* yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Dari beberapa pernyataan tersebut maka dapat disimpulkan bahwa basis data (*database*) adalah sekumpulan data yang saling berelasi yang disimpan dalam media elektronis dan diorganisasi sedemikian rupa agar dapat dimanfaatkan kembali.

### 2.8.1 Keuntungan dan Kerugian Penggunaan Basis Data (*Database*)

Adapun keuntungan dan kerugian penggunaan basis data (*database*) menurut Lubis (2016:8) adalah sebagai berikut :

1. Terkontrolnya kerangkapan data dan inkonsistensi
2. Terpeliharanya keselarasan data
3. Data dapat dipakai secara bersama-sama
4. Memudahkan penerapan standarisasi
5. Memudahkan penerapan batasan-batasan pengamanan
6. Terpeliharanya integritas data
7. Terpeliharanya keseimbangan atas perbedaan kebutuhan data dari setiap aplikasi.
8. Program / *data independent*

Adapun kerugian dari penggunaan basis data (*database*) adalah sebagai berikut :

1. Mahal dalam implementasinya
2. Rumit / kompleks
3. Penanganan proses *recovery & backup* sulit
4. Kerusakan pada sistem basis data (*database*) dapat mempengaruhi departemen yang terkait.

## 2.9 Pendukung Perancangan

### 2.9.1 *Object Oriented Analysis Design (OOAD)*

Menurut Mathiassen (dalam Purwaningtias 2018:2), OOAD adalah metode untuk menganalisa dan merancang sistem dengan pendekatan berorientasi *object*.

*Object* diartikan sebagai entitas yang memiliki identitas, *state* dan *behavior*. Pada analisa identitas *object* menjelaskan bagaimana *user* membedakannya dari *object* lain dan *behavior object* digambarkan melalui *event* yang dilakukan. Pada perancangan, identitas *object* dengan bagaimana *object* lain mengenalinya sehingga bisa diakses dan *behavior* dengan operation yang dilakukan. Sehingga *object* satu bisa mempengaruhi *object* lain dalam sistem .

Menurut Hasanuddin (2016:90) OOAD adalah metode analisis yang memeriksa *requirements* dari sudut pandang kelas-kelas dan objek yang ditemui dalam ruang lingkup permasalahan yang mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

Adapun teknik untuk perancangan OOAD (*Object Oriented Analysis Design*) adalah menggunakan *Unified Modelling Language* (UML).

### **2.9.2 Unified Modelling Language (UML)**

Menurut Herlinah dan Musliadi (2019:50), UML merupakan salah satu alat bantu pengembangan sistem berorientasi obyek .

Sedangkan menurut Huda (2019:110), UML adalah bahasa pemodelan standar yang terdiri dari seperangkat diagram ter-integrasi untuk membantu pengembangan sistem dan perangkat lunak menentukan, membuat gambaran, mengembangkan, dan mendokumentasikan artefak sistem perangkat lunak, serta untuk pemodelan bisnis dan sistem non-perangkat lunak lainnya .

Berdasarkan beberapa pernyataan diatas, dapat disimpulkan bahwa *Unified Modelling Language* adalah bahasa pemodelan standar yang menjadi alat bantu

dalam pengembangan sistem berorientasi obyek baik itu sistem dengan perangkat lunak maupun non-perangkat lunak.

Adapun beberapa Tipe-tipe struktur Diagram UML, yaitu:

#### 1. *Use Case Diagram*

Menurut Munawar (2018:89), *Use case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) dengan sistemnya sendiri melalui cerita bagaimana sebuah sistem dipakai.

Diagram use case menunjukkan 3 aspek dari sistem yaitu :

##### a. *Actor*

*Actor* adalah *abstraction* dari orang dalam sistem yang lain yang mengaktifkan fungsi dari target. Orang atau sistem bisa muncul dalam beberapa peran.

##### b. *Use Case*

*Use Case* adalah abstraksi dari interaksi antara sistem dan *actor*.

##### c. *System / Sub System Boundary* (batasan Sistem / Sub Sistem)

#### 2. *Use Case Specification*

Menurut Munawar (2018:97), *usecase specification* adalah representasi tekstual yang menggambarkan urutan kejadian bersama dengan *use case* terkait lainnya dalam format tertentu.

Setiap *usecase* perlu disertasi teks yang menjelaskan tujuan dari *usecase* dan juga fungsi apa yang dicapai ketika *usecase* dijalankan. Pada awalnya, hanya deskripsi singkat tentang langkah-langkah yang diperlukan untuk

melaksanakan aliran usecase (yaitu, fungsionalitas apa yang disediakan oleh *usecase*) yang ditulis. Seiring berjalannya analisis, langkah-langkah apa yang harus dilakukan perlu dijelaskan lebih lanjut. Akhirnya, kondisi perkecualian (jika ada) bisa ditambahkan.

### 3. *Class Diagram*

Menurut Munawar Munawar (2018: 101), *Class diagram* adalah diagram statis. Ini mewakili pandangan statis dari suatu aplikasi. *Class diagram* menggambarkan atribut, operation dan juga *constraint* yang terjadi pada sistem.

Ada beberapa hal yang harus diingat saat membuat *class diagram* yaitu :

- a. Nama *class diagram* harus memiliki makna untuk menggambarkan aspek sistem
- b. Setiap elemen dan hubungan mereka harus diidentifikasi sebelumnya.
- c. Atribut dan *operation* dari masing-masing *class* harus diidentifikasi secara jelas.
- d. Untuk setiap *class*, jumlah minimum properti harus ditentukan, karena properti yang tidak perlu akan membuat diagram menjadi rumit
- e. Gunakan catatan apapun yang diperlukan untuk menjelaskan beberapa aspek diagram, namun tetap harus dimengerti oleh pengembang/*programmer*.
- f. Yang terakhir, pikirkan berulang kali sebelum membuat versi final. Pastikan bahwa *class diagram* yang tergambar benar-benar mempresentasikan keadaan riil dari sistem.

#### 4. *Activity Diagram*

Menurut Munawar (2018:127), *Activity diagram* adalah bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa.

Secara umum tujuan dari *activity diagram* bisa digambarkan sebagai berikut:

- a. Menggambarkan aliran aktifitas dari sistem
- b. Menggambarkan urutan aktifitas dari suatu aktifitas ke aktifitas lainnya.
- c. Menggambarkan paralelisme, percabangan aliran konkuren dari sistem.

#### 5. *Sequence Diagram*

Menurut Munawar (2018:137), *Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh obyek yang dituliskan dengan kotak segiempat bernama. *Message* diawali oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress *vertical*.

### 2.10 Perangkat Lunak Pendukung

Adapun perangkat lunak yang digunakan untuk membangun Sistem Presensi *Quick Response Code* Berbasis Web dan *Android Client* adalah *framework Php CodeIgniter* untuk *interface web*, IDE MIT App Inventor untuk *interface Android*, dan menggunakan MySQL sebagai basis data.



### 2.10.1 Framework Php CodeIgniter

Menurut Sulistiono (2018:5) “ PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang digunakan untuk membuat *website* atau situs dinamis dan menangani rangkaian bahasa pemrograman antara *client side* dan *server side* “.

Menurut Abdulloh (2018:127), “PHP merupakan kependekan dari *Hypertext Preprocessor* yaitu bahasa pemrograman web yang dapat disisipkan dalam skrip HTML dan bekerja disisi server. Tujuan dari bahasa ini adalah membantu para pengembang web untuk membangun web dinamis dengan cepat ”.

Berdasarkan beberapa pernyataan diatas maka dapat disimpulkan bahwa PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman web yang bekerja di sisi *server* dan dapat menangani rangkaian bahasa pemrograman *client* dan *server side*.

Menurut Widodo (2015:6), *Framework* adalah suatu kerangka kerja dalam aplikasi yang didalamnya memiliki fungsi-fungsi (bagian-bagian) program yang telah disusun, sehingga *programmer* tidak perlu membuat kode dari nol, karena *framework* telah menyediakan *library* yang diperlukan.

Menurut Sulistiono (2018:7), *Codeigniter* adalah sebuah aplikasi *open source* yang berupa kerangka kerja atau *framework* untuk membangun *website* menggunakan bahasa pemrograman PHP. Tujuannya memungkinkan pengembangan proyek yang lebih cepat daripada penulisan kode dasar atau kode terstruktur, dengan menyediakan banyak *library* yang biasanya digunakan dalam pengerjaan.

### 2.10.2 Konsep MVC (*Model View Controller*)

Menurut Sulistiono (2018:8), konsep MVC merupakan rangkaian logika yang dipresentasikan kedalam sebuah perangkat yang terpisah, sehingga dalam implementasi ini meminimalisir penulisan kode yang berulang. Adapun bagian dari konsep MVC tersebut adalah sebagai berikut :

#### 1. *Model*

*Model* merupakan rangkaian logika yang mengimplementasikan pengelolaan struktur basis data dan biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*), serta menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

#### 2. *View*

*View* merupakan struktur bagian yang menangani presentasi logika. Pada suatu aplikasi *web*, bagian ini biasanya berupa *file template HTML* yang diatur oleh *controller*. *View* berfungsi untuk menerima dan mempresentasikan data kepada pengguna. Bagian ini tidak mewakili akses langsung terhadap bagian *model*. Dalam *folder view* berisikan berkas-berkas atau *file* berekstensi *.php* dan biasanya *form, table, paragraph*, dan lain-lain.

#### 3. *Controller*

*Controller* adalah penghubung antara *Model* dan *View* yang memproses permintaan dari sisi klien ke sisi *server* dan memprosesnya ke *HTTP request* untuk mengeksplorasi ke sebuah *website*.

### 2.10.3 IDE MIT App Inventor

Menurut Kadir (2018:2), App inventor merupakan salah satu perangkat pengembangan aplikasi Android. Perangkat ini menggunakan pendekatan blok untuk membentuk aplikasi sehingga sangat mudah digunakan oleh siapa saja.

Menurut Kadir (2017:5), App Inventor adalah alat pengembangan yang digunakan untuk membangun aplikasi di Android. Peranti ini diciptakan di MIT (*Massachusetts Institute of Technology*) dengan tujuan untuk memudahkan pembuatan aplikasi di Android. MIT menciptakan alat pengembangan yang mudah digunakan oleh siapa saja dengan menggunakan pendekatan blok. Adapun desain layar dilakukan dengan pendekatan “*click & drag*”.

### 2.10.4 MySQL Database

Menurut Yanto (2016:72) MySQL merupakan sistem manajemen *database* yang bersifat *open source* atau gratis. Keunggulan yang bisa kita ketahui dari MySQL antara lain :

1. Kecepatan
2. Kemudahan bagi user dalam penggunaannya
3. Bersifat *open source* atau gratis
4. Support dengan bahasa query
5. *User* dapat mengakses lebih dari satu dalam satu waktu
6. Akses data dapat dilakukan di setiap tempat dengan fasilitas internet.
7. Mysql mudah didapatkan karna *source code* yang dapat disebarluaskan.

Informasi yang tersimpan dalam bahasa *query* terbagi atas dua bagian terdiri dari baris atau dikenal dengan *Record* dan kolom atau dikenal dengan *field*. Adapun ketentuan dalam memberikan perintah pada MySql adalah sebagai berikut:

- a. Perintah MySQL mengenal *case insensitive* yaitu penulisan dalam huruf besar ataupun huruf kecil.
- b. Setiap mengakhiri perintah menggunakan tanda titik koma (;) .

## BAB III

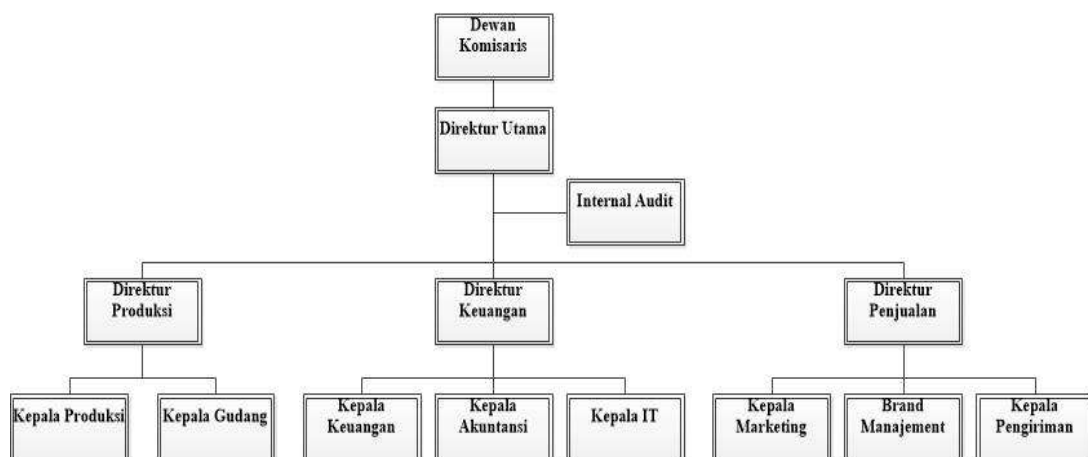
### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 CV. Indoscots Baby Utama

CV. Indoscots Baby Utama merupakan perusahaan yang bergerak dibidang garmen dan tekstil penghasil produk perlengkapan bayi dan pakaian anak yang telah meraih top *Brand For Kids* tahun 2012 – 2019. CV. Indoscots Baby Utama didirikan pada tanggal 5 Mei 2006, dan berlokasi di Jl. Bihbul Raya 2 No.102 Kopo . Distribusi dan pemasaran produk tersebar di seluruh Indonesia dengan beberapa kantor cabang yang tersebar di beberapa luar pulau jawa.

##### 3.1.1 Struktur Organisasi

CV. Indoscots Baby Utama dalam melaksanakan aktifitasnya sehari-hari terdiri dari beberapa bagian yang satu dengan yang lainnya saling berkaitan erat. Adapun Struktur organisasi CV. Indoscots Baby Utama adalah sebagai berikut :



Gambar 3.1 Struktur Organisasi CV. Indoscots Baby Utama

Adapun deskripsi pekerjaan dari struktur organisasi tersebut adalah sebagai berikut :

1. Dewan Komisaris

Adapun tugas dari Dewan Komisaris adalah sebagai berikut :

- a. Melakukan pengawasan terhadap kebijakan Direksi dalam melaksanakan pengurusan Perseroan serta memberi nasehat kepada Direksi termasuk pelaksanaan Rencana Jangka Panjang Perusahaan, rencana kerja dan anggaran perusahaan serta ketentuan Anggaran Dasar dan Rapat Umum Pemegang Saham dan peraturan perundang-undangan yang berlaku.
- b. Melakukan evaluasi terhadap laporan atas pencapaian target dari masing masing Indikator Kinerja Kunci (IKK) atau *Key Performance Indicator* (KPI) dan melakukan tindakan yang diperlukan dalam rangka pencapaian target yang telah ditetapkan.

2. Direktur Utama

Adapun Tugas dari Direktur Utama adalah sebagai berikut :

- a. Mengelola Perusahaan sesuai amanat RUPS untuk mewujudkan sasaran perusahaan.
- b. Memimpin, mengkoordinasikan dan mensinkronisasikan pelaksanaan program kegiatan Direktur Produksi, Direktur Keuangan dan Direktur Penjualan.
- c. Menjalankan arahan dari Dewan Komisaris dan RUPS.

- d. Mengatur pembagian tugas dan wewenang masing-masing anggota Direksi.
- e. Mengadakan dan memimpin rapat Direksi secara berkala, untuk mengevaluasi pelaksanaan program kegiatan masing-masing Direktorat.
- f. Memberi penjelasan kepada Dewan Komisaris atau Rapat Umum Pemegang Saham, mengenai Rencana Jangka Panjang Perusahaan, Rencana Kerja dan Anggaran Perusahaan serta Laporan Tahunan.
- g. Mengkoordinir penyelenggaraan akuntansi keuangan, akuntansi biaya, verifikasi dan administrasi aset.
- h. Mengkoordinir pembuatan Laporan Manajemen Triwulan, Semesteran dan Tahunan yang akan disampaikan kepada Dewan Komisaris dan Pemegang Saham

### 3. Audit Internal

Adapun tugas dari Audit Internal adalah sebagai berikut :

- a. Mencari informasi awal terkait bagian yang akan di audit
- b. Melakukan tinjauan dokumen dan persyaratan lain yang berkaitan dengan audit.
- c. Mempersiapkan program audit tahunan dan jadwal pelaksanaan audit secara terperinci.
- d. Membuat daftar pertanyaan audit
- e. Melaksanakan pemeriksaan sistem secara menyeluruh.
- f. Mengumpulkan dan menganalisis bukti audit yang relevan .

- g. Melaporkan temuan audit atau masalah-masalah yang ditemukan selama audit internal kepada Direktur Utama.
- h. Memantau tindak lanjut hasil audit internal sampai dinyatakan selesai.

#### 4. Direktur Produksi

Adapun tugas dari Direktur Produksi adalah sebagai berikut :

- a. Memimpin dan mengkoordinasikan tugas-tugas dibawah Direktorat Produksi.
- b. Menyusun Rencana Kerja dan Anggaran Perusahaan dibawah Direktorat Produksi.
- c. Melaksanakan dan mengendaikan program kegiatan produksi dan pengepakan.
- d. Menjalankan arahan dari Direktur Utama, Dewan Komisaris dan RUPS.
- e. Mengadakan rapat-rapat internal secara berkala di lingkungan Direktorat Produksi untuk mengevaluasi pelaksanaan program kegiatan.
- f. Menindaklanjuti temuan hasil audit internal yang berkaitan dengan tugas operasionalnya.
- g. Merumuskan kebijakan dan menyusun pedoman kerja yang digunakan dilingkungan Direktorat Produksi dan selanjutnya disampaikan kepada Direktur Utama untuk ditetapkan.



## 5. Direktur Keuangan

Adapun tugas dari Direktur Keuangan adalah sebagai berikut :

- a. Memimpin dan mengkoordinasikan tugas-tugas di bawah Direktorat Keuangan.
- b. Menyusun Rencana Kerja dan Anggaran Perusahaan di bawah Direktorat Keuangan.
- c. Melaksanakan dan mengendalikan program kegiatan yang telah dirumuskan, meliputi Bidang Keuangan dan Akuntansi.
- d. Menjalankan arahan-arahan dari Direktur Utama, Dewan Komisaris, dan RUPS.
- e. Mengadakan rapat internal secara berkala guna membahas masalah masalah dibidang Keuangan dan Akuntansi.
- f. Menindaklanjuti temuan hasil audit internal yang berkaitan dengan tugas operasionalnya.
- g. Menyelenggarakan dan memelihara akuntansi keuangan, akuntansi biaya, verifikasi dan akuntansi aset.
- h. Merumuskan kebijakan dan menyusun pedoman kerja yang digunakan dilingkungan Direktorat Keuangan dan selanjutnya disampaikan kepada Direktur Utama untuk ditetapkan.

## 6. Direktur Penjualan

Adapun tugas dari Direktur Penjualan adalah sebagai berikut :

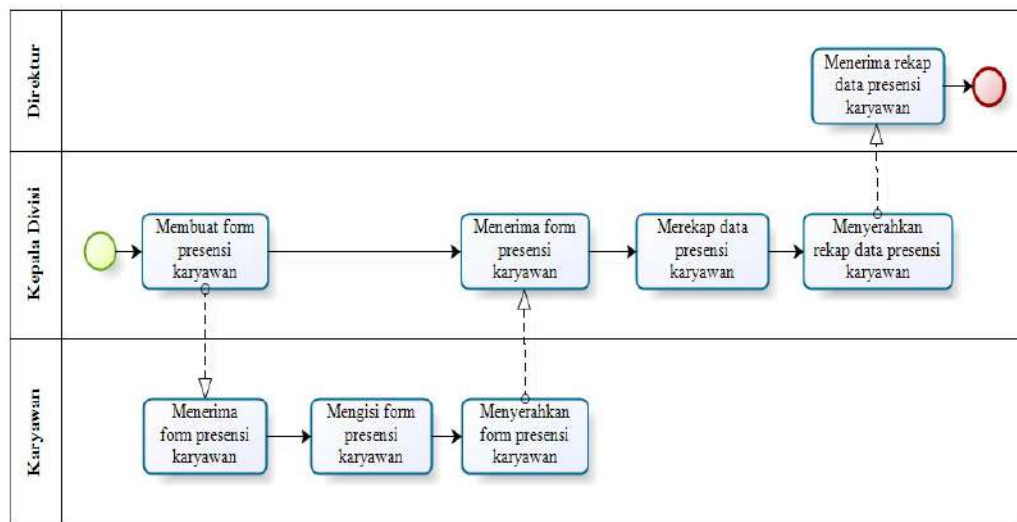
- a. Memimpin dan mengkoordinasikan tugas-tugas dibawah Direktorat Penjualan.

- b. Menyusun Rencana Kerja dan Anggaran Perusahaan dibawah Direktorat Penjualan.
- c. Melaksanakan dan mengendaikan program kegiatan pemasaran dan penjualan produk.
- d. Menjalankan arahan dari Direktur Utama, Dewan Komisaris dan RUPS.
- e. Mengadakan rapat-rapat internal secara berkala di lingkungan Direktorat Penjualan untuk mengevaluasi pelaksanaan program kegiatan.
- f. Menindaklanjuti temuan hasil audit internal yang berkaitan dengan tugas operasionalnya.
- g. Merumuskan kebijakan dan menyusun pedoman kerja yang digunakan dilingkungan Direktorat Penjualan dan selanjutnya disampaikan kepada Direktur Utama untuk ditetapkan.

### **3.2 Analisis Proses Bisnis Sistem Lama**

Analisis proses bisnis sistem yang sedang berjalan bertujuan untuk mengetahui secara jelas tentang sistem yang sedang berjalan. Tahap ini dilakukan untuk mempermudah dalam mengetahui hal apa saja yang perlu di usulkan untuk perancangan sistem baru yang akan diterapkan.

Adapun Analisis proses bisnis sistem yang sedang berjalan di CV. Indoscots Baby Utama penulis gambarkan sebagai berikut:



Gambar 3.2 Proses bisnis sistem yang sedang berjalan

### 3.3 Analisis SWOT

#### 1. *Strength* (Kekuatan)

Data presensi karyawan di kumpulkan dan diperiksa oleh Kepala Divisi sebelum diserahkan kepada masing-masing Direktur.

#### 2. *Weakness* (Kelemahan)

Proses perekapan data presensi yang dilakukan oleh Kepala Divisi masih dilakukan dengan cara semi *Elektronik Data Processing (EDP)* dengan melakukan entri data satu-persatu kedalam aplikasi pengolah angka (*Microsoft Excel*) sehingga menyebabkan proses perekapan data presensi memakan waktu cukup lama .

#### 3. *Opportunity* (Peluang)

Sebuah sistem yang sudah terkomputerisasi secara menyeluruh dalam kegiatan presensi akan mengatasi resiko kecurangan pengisian data presensi oleh karyawan dan mempunyai peluang untuk mempermudah Kepala Divisi dalam perekapan data presensi karyawan.

#### 4. *Threat (Ancaman)*

Proses kegiatan presensi dilakukan sendiri oleh karyawan pada sebuah kertas formulir presensi memiliki resiko kecurangan dalam pengisiannya.

### 3.4 **Gambaran Umum Sistem Yang Diusulkan**

Gambaran sistem yang diusulkan berupa sistem presensi *Quick Response Code* berbasis *web* dan *andorid client* . Dimana perancangan sistem yang diusulkan yaitu merubah sistem yang sedang berjalan secara semi *Elektronic Data Processing* menjadi lebih terkomputerisasi lagi, yang bertujuan untuk mempermudah kegiatan pendataan kehadiran karyawan secara cepat dan waktu yang akurat serta dapat memantau lokasi beberapa karyawan yang sedang menjalankan kebijakan *work from home* maupun yang sedang bekerja dilingkungan perkantoran CV. Indoscots Baby Utama.

### 3.5 ***Unified Modelling Language (UML)***

*Unified Modelling Language* adalah sebuah “ bahasa pemodelan “ yang telah menjadi standar dalam industri untuk visualisasi, perancangan, dan pendokumentasian sistem piranti lunak. Dalam UML dibawah digambarkan proses perancangan sistem presensi *quick response code* pada CV. Indoscots Baby Utama.

#### 3.5.1 ***Usecase Diagram***

*Usecase Diagram* memperlihatkan hubungan-hubungan yang terjadi antara aktor-aktor dengan *usecase* dalam sistem. Adapun pendefinisian aktor dan *usecase* diagram sistem yang akan diusulkan adalah sebagai berikut.

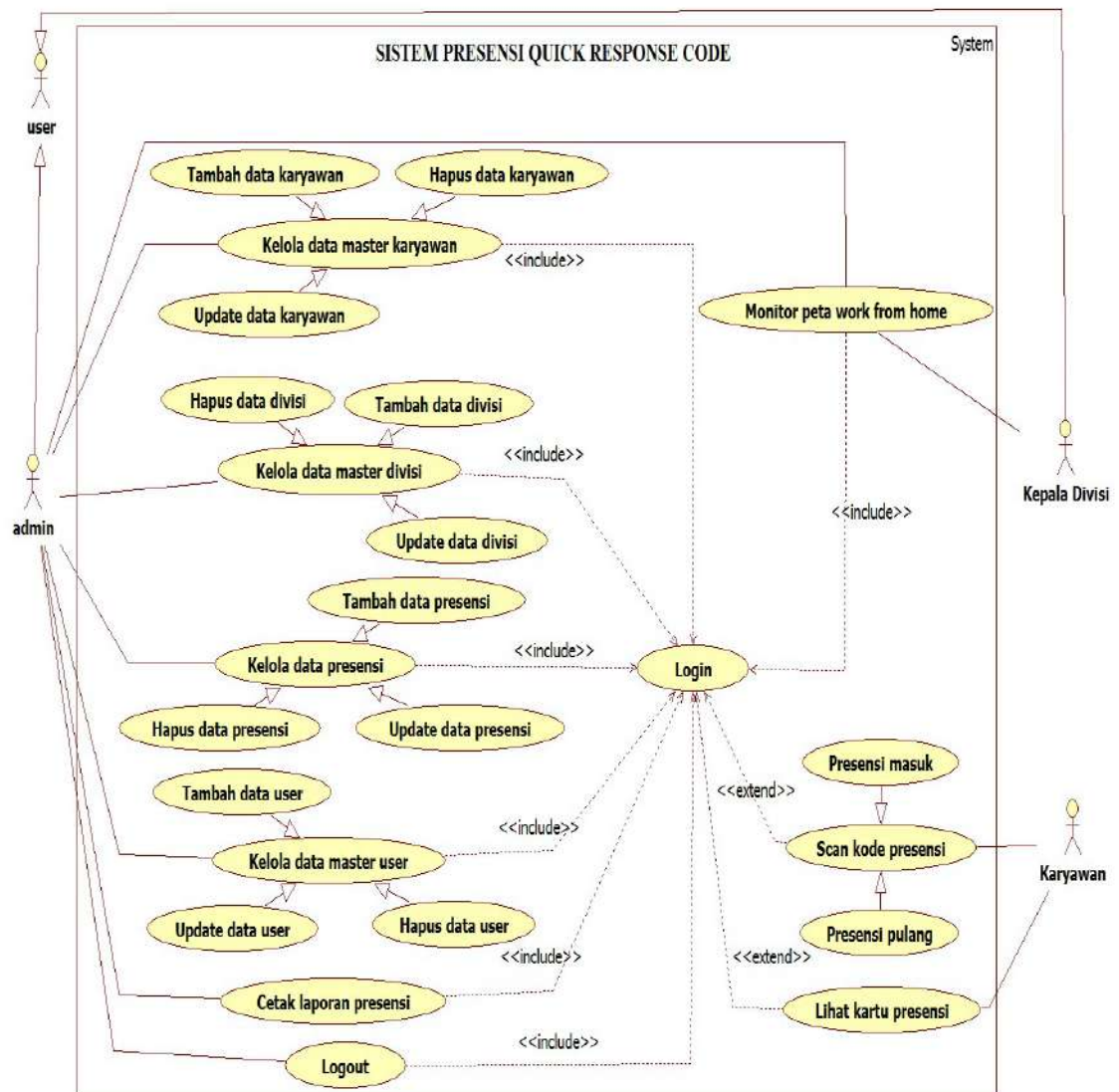
Tabel 3.1 Pendefinisian Aktor

No	Aktor	Deskripsi
1	Admin	Admin adalah orang yang memiliki akses ke dalam sistem dan bertugas untuk melakukan pengelolaan data karyawan, divisi, presensi, dan pengguna ( <i>user</i> ) sistem, serta memiliki akses untuk mencetak laporan presensi dan monitoring peta sebaran <i>work from home</i> .
2	Kepala Divisi	Kepala divisi adalah orang yang memiliki akses terbatas ke dalam sistem seperti mencetak laporan presensi dan melakukan monitoring peta sebaran <i>work from home</i>
3	<i>User</i>	<i>User</i> merupakan bentuk umum dari aktor Admin dan Kepala Divisi sebagai dua aktor yang memiliki akses masuk kedalam sistem.
4	Karyawan	Karyawan adalah orang yang telah terdaftar dalam sistem presensi, hal-hal yang dapat dilakukan oleh karyawan adalah melakukan <i>scan</i> pada kode QR pada waktu jam masuk dan jam pulang kerja sebagai pencatatan data kehadiran, serta dapat melihat kartu presensi dari karyawan tersebut

Tabel 3.2 Pendefinisian *Usecase*

No	<i>Usecase</i>	Deskripsi
1	Login	Memungkinkan aktor untuk masuk dan mengakses layanan sistem
2	Kelola Data Master Karyawan	Merupakan bentuk umum dari proses pengelolaan data karyawan yang meliputi menambah, <i>update</i> , dan menghapus data karyawan
3	Tambah Data Karyawan	Memungkinkan aktor untuk menambah data karyawan pada database
4	<i>Update</i> Data Karyawan	Memungkinkan aktor untuk <i>update</i> data karyawan pada database
5	Hapus Data Karyawan	Memungkinkan aktor untuk menghapus data karyawan pada database
6	Kelola Data Master Divisi	Merupakan bentuk umum dari proses pengelolaan data divisi yang meliputi menambah, <i>update</i> , dan menghapus data divisi
7	Tambah Data Divisi	Memungkinkan aktor untuk menambah data divisi pada database
8	<i>Update</i> Data Divisi	Memungkinkan aktor untuk <i>update</i> data divisi pada database
9	Hapus Data Divisi	Memungkinkan aktor untuk menghapus data

		divisi pada database
10	Kelola Data Master Presensi	Merupakan bentuk umum dari proses pengelolaan data presensi yang meliputi menambah, mengupdate, dan menghapus data presensi
11	Tambah Data Presensi	Memungkinkan aktor untuk menambah data presensi pada database
12	<i>Update</i> Data Presensi	Memungkinkan aktor untuk mengupdate data presensi pada database
13	Hapus Data Presensi	Memungkinkan aktor untuk menghapus data presensi pada database
14	Kelola Data Master <i>User</i>	Merupakan bentuk umum dari proses pengelolaan data <i>user</i> yang meliputi menambah, mengupdate, dan menghapus data <i>user</i>
15	Tambah Data <i>User</i>	Memungkinkan aktor untuk menambah data <i>user</i> pada database
16	<i>Update</i> Data <i>User</i>	Memungkinkan aktor untuk mengupdate data <i>user</i> pada database
17	Hapus Data <i>user</i>	Memungkinkan aktor untuk menghapus data <i>user</i> pada database
18	<i>Monitor</i> Peta <i>Work From Home</i>	Memungkinkan aktor untuk melihat peta sebaran karyawan yang sedang melakukan work from home (bekerja dari rumah)
19	Cetak Laporan Presensi	Memungkinkan aktor untuk mencetak laporan presensi
20	<i>Scan</i> Kode Presensi	Merupakan bentuk umum dari proses <i>scan</i> kode presensi saat jam masuk dan pulang dalam bentuk kode QR
21	Presensi Masuk	Memungkinkan aktor untuk melakukan presensi masuk pada sistem
22	Presensi Pulang	Memungkinkan aktor untuk melakukan presensi pulang pada sistem
23	Lihat Kartu Presensi	Memungkinkan aktor untuk melihat kartu presensi
24	Logout	Memungkinkan aktor untuk keluar dari layanan sistem



Gambar 3.3 Usecase Diagram

### 3.5.2 Usecase Spesification

Setiap *usecase* perlu disertai teks yang menjelaskan tujuan dari *usecase* dan juga fungsi apa yang dicapai ketika *usecase* dijalankan serta deskripsi singkat tentang langkah-langkah yang diperlukan untuk melaksanakan aliran *usecase*. *Usecase spesification* dibuat dalam bentuk tabel skenario (*flow of event*).

## 1. Spesifikasi *Login*

Tabel 3.3 *Usecase Specification Login*

<i>Usecase Specification</i>	
Nama <i>Use Case</i>	<i>Login</i>
Aktor	<i>User</i>
Deskripsi	Memungkinkan aktor untuk masuk dan mengakses layanan sistem
<b>Skenario Utama</b>	
<b><i>Pre-Condition</i></b>	Aktor tidak dapat mengakses layanan sistem
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Membuka halaman login	2. Menampilkan halaman login
3. Menginput <i>username</i> dan <i>password</i>	4. Validasi <i>username</i> dan <i>password</i> yang diinput aktor
	5. Halaman utama sistem ditampilkan
<b>Skenario Alternatif</b>	
1. Membuka halaman login	2. Menampilkan halaman login
3. Menginput <i>username</i> dan <i>password</i> yang valid	4. Validasi <i>username</i> dan <i>password</i> yang diinput aktor
	5. Halaman utama sistem ditampilkan
<b><i>Post-Condition</i></b>	Aktor dapat mengakses layanan sistem

## 2. Spesifikasi Tambah Data Karyawan

Tabel 3.4 *Usecase Specification Tambah Data Karyawan*

<b>Usecase Specification</b>	
Nama <i>Use Case</i>	Tambah data karyawan
Aktor	Admin
Deskripsi	Memungkinkan aktor untuk menambah data karyawan pada <i>database</i>
<b>Skenario Utama</b>	
<b><i>Pre-Condition</i></b>	Sistem menampilkan halaman utama
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>



1. Memilih menu <i>master</i> karyawan	2. Menampilkan data <i>master</i> karyawan
3. Memilih tambah data master karyawan	4. Menampilkan form tambah data
5. Mengisi form tambah data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data karyawan yang diinput aktor pada <i>database</i>
	8. Data karyawan baru ditambahkan pada <i>database</i>
<b>Skenario Alternatif</b>	
1. Memilih menu <i>master</i> karyawan	2. Menampilkan data <i>master</i> karyawan
3. Memilih tambah data master karyawan	4. Menampilkan form tambah data
5. Mengisi form tambah data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data karyawan yang diinput aktor pada <i>database</i>
	8. Menampilkan pesan data sudah ada
9. Mengisi form tambah data	
10. Memilih tombol <i>save</i>	11. Data karyawan di tambahkan pada <i>database</i>
<b>Post-Condition</b>	Data karyawan baru ditambahkan pada <i>database</i>

### 3. Spesifikasi *Update* Data Karyawan

Tabel 3.5 *Usecase Spesification Update* Data Karyawan

Use Case Specification	
Nama <i>Use Case</i>	<i>Update</i> data karyawan
Aktor	Admin
Deskripsi	Memungkinkan aktor untuk mengupdate data karyawan pada <i>database</i>
<b>Skenario Utama</b>	
<b>Pre-Condition</b>	Sistem menampilkan halaman utama
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih menu <i>master</i> karyawan	2. Menampilkan data <i>master</i> karyawan
3. Memilih <i>edit</i> data master karyawan	4. Menampilkan form <i>update</i> data
5. Mengisi form <i>update</i> data	

6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data karyawan yang diinput aktor pada <i>database</i>
	8. Data karyawan di- <i>update</i> pada <i>database</i>
<b>Skenario Alternatif</b>	
1. Memilih menu <i>master</i> karyawan	2. Menampilkan data <i>master</i> karyawan
3. Memilih edit data master karyawan	4. Menampilkan form <i>update</i> data
5. Mengisi form <i>update</i> data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data karyawan yang diinput aktor pada <i>database</i>
	8. Menampilkan pesan data sudah ada
9. Mengisi form <i>update</i> data	
10. Memilih tombol <i>save</i>	11. Data karyawan di- <i>update</i> pada <i>database</i>
<b>Post-Condition</b>	Data karyawan di- <i>update</i> pada <i>database</i>

#### 4. Spesifikasi Hapus Data Karyawan

Tabel 3.6 *Usecase Specification* Hapus Data Karyawan

<b>Use Case Specification</b>	
Nama <i>Use Case</i>	Hapus data karyawan
Aktor	Admin
Deskripsi	Memungkinkan aktor untuk menghapus data karyawan pada <i>database</i>
<b>Skenario Utama</b>	
<b>Pre-Condition</b>	Sistem menampilkan halaman utama
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih menu <i>master</i> karyawan	2. Menampilkan data <i>master</i> karyawan
3. Memilih tombol <i>delete</i>	4. Menampilkan pesan konfirmasi hapus
5. Konfirmasi hapus	6. Data karyawan di hapus pada <i>database</i>
<b>Post-Condition</b>	Data karyawan dihapus pada <i>database</i>

## 5. Spesifikasi Tambah Data Divisi

Tabel 3.7 *Usecase Spesification* Tambah Data Divisi

Use Case Specification	
Nama <i>Use Case</i>	Tambah data divisi
Aktor	Admin
Deskripsi	Memungkinkan aktor untuk menambah data divisi pada <i>database</i>
Skenario Utama	
<b><i>Pre-Condition</i></b>	Sistem menampilkan halaman utama
Aksi Aktor	Reaksi Sistem
1. Memilih menu <i>master</i> divisi	2. Menampilkan data <i>master</i> divisi
3. Memilih tambah data master divisi	4. Menampilkan form tambah data
5. Mengisi form tambah data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data divisi yang diinput aktor pada <i>database</i>
	8. Data divisi baru ditambahkan pada <i>database</i>
Skenario Alternatif	
1. Memilih menu <i>master</i> divisi	2. Menampilkan data <i>master</i> divisi
3. Memilih tambah data master divisi	4. Menampilkan form tambah data
5. Mengisi form tambah data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data divisi yang diinput aktor pada <i>database</i>
	8. Menampilkan pesan data sudah ada
9. Mengisi form tambah data	
10. Memilih tombol <i>save</i>	11. Data divisi di tambahkan pada <i>database</i>
<b><i>Post-Condition</i></b>	Data divisi baru ditambahkan pada <i>database</i>

## 6. Spesifikasi *Update Data Divisi*

Tabel 3.8 *Usecase Spesification Update Data Divisi*

Use Case Specification	
Nama <i>Use Case</i>	<i>Update data divisi</i>
Aktor	Admin
Deskripsi	Memungkinkan aktor untuk mengupdate data divisi pada <i>database</i>
Skenario Utama	
<b><i>Pre-Condition</i></b>	Sistem menampilkan halaman utama
Aksi Aktor	Reaksi Sistem
1. Memilih menu <i>master divisi</i>	2. Menampilkan data <i>master divisi</i>
3. Memilih <i>edit</i> data master divisi	4. Menampilkan form <i>update data</i>
5. Mengisi form <i>update data</i>	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data divisi yang diinput aktor pada <i>database</i>
	8. Data divisi di- <i>update</i> pada database
Skenario Alternatif	
1. Memilih menu <i>master divisi</i>	2. Menampilkan data <i>master divisi</i>
3. Memilih edit data master divisi	4. Menampilkan form <i>update data</i>
5. Mengisi form <i>update data</i>	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data divisi yang diinput aktor pada <i>database</i>
	8. Menampilkan pesan data sudah ada
9. Mengisi form <i>update data</i>	
10. Memilih tombol <i>save</i>	11. Data divisi di- <i>update</i> pada <i>database</i>
<b><i>Post-Condition</i></b>	Data divisi di- <i>update</i> pada <i>database</i>

## 7. Spesifikasi Hapus Data Divisi

Tabel 3.9 *Usecase Spesification Hapus Data Divisi*

Use Case Specification	
Nama <i>Use Case</i>	Hapus data divisi
Aktor	Admin

Deskripsi	Memungkinkan aktor untuk menghapus data divisi pada <i>database</i>
<b>Skenario Utama</b>	
<b><i>Pre-Condition</i></b>	Sistem menampilkan halaman utama
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih menu <i>master</i> divisi	2. Menampilkan data <i>master</i> divisi
3. Memilih tombol <i>delete</i>	4. Menampilkan pesan konfirmasi hapus
5. Konfirmasi hapus	6. Data divisi di hapus pada <i>database</i>
<b><i>Post-Condition</i></b>	Data divisi dihapus pada <i>database</i>

## 8. Spesifikasi Tambah Data Presensi

Tabel 3.10 *Usecase Spesification* Tambah Data Presensi

<b>Use Case Specification</b>	
Nama <i>Use Case</i>	Tambah data presensi
Aktor	Admin
Deskripsi	Memungkinkan aktor untuk menambah data presensi pada <i>database</i>
<b>Skenario Utama</b>	
<b><i>Pre-Condition</i></b>	Sistem menampilkan halaman utama
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih menu <i>master</i> presensi	2. Menampilkan data <i>master</i> presensi
3. Memilih tambah data master presensi	4. Menampilkan form tambah data
5. Mengisi form tambah data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data presensi yang diinput aktor pada <i>database</i>
	8. Data presensi baru ditambahkan pada <i>database</i>
<b>Skenario Alternatif</b>	
1. Memilih menu <i>master</i> presensi	2. Menampilkan data <i>master</i> presensi
3. Memilih tambah data master presensi	4. Menampilkan form tambah data
5. Mengisi form tambah data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data presensi yang diinput aktor pada

	<i>database</i>
	8. Menampilkan pesan data sudah ada
9. Mengisi form tambah data	
10. Memilih tombol <i>save</i>	11. Data presensi di tambahkan pada <i>database</i>
<b>Post-Condition</b>	Data presensi baru ditambahkan pada <i>database</i>

## 9. Spesifikasi Update Data Presensi

Tabel 3.11 *Usecase Spesification Update Data Presensi*

Use Case Specification	
Nama <i>Use Case</i>	<i>Update data presensi</i>
Aktor	Admin
Deskripsi	Memungkinkan aktor untuk mengupdate data presensi pada <i>database</i>
Skenario Utama	
<b>Pre-Condition</b>	Sistem menampilkan halaman utama
Aksi Aktor	Reaksi Sistem
1. Memilih menu <i>master</i> presensi	2. Menampilkan data <i>master</i> presensi
3. Memilih <i>edit</i> data master presensi	4. Menampilkan form <i>update</i> data
5. Mengisi form <i>update</i> data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data presensi yang diinput aktor pada <i>database</i>
	8. Data presensi di- <i>update</i> pada database
Skenario Alternatif	
1. Memilih menu <i>master</i> presensi	2. Menampilkan data <i>master</i> presensi
3. Memilih edit data master presensi	4. Menampilkan form <i>update</i> data
5. Mengisi form <i>update</i> data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data presensi yang diinput aktor pada <i>database</i>
	8. Menampilkan pesan data sudah ada
9. Mengisi form <i>update</i> data	
10. Memilih tombol <i>save</i>	11. Data presensi di- <i>update</i> pada <i>database</i>

<b>Post-Condition</b>	Data presensi di- <i>update</i> pada <i>database</i>
-----------------------	--

## 10. Spesifikasi Hapus Data Presensi

Tabel 3.12 *Usecase Spesification* Hapus Data Presensi

Use Case Specification	
Nama <i>Use Case</i>	Hapus data presensi
Aktor	Admin
Deskripsi	Memungkinkan aktor untuk menghapus data presensi pada <i>database</i>
Skenario Utama	
<b>Pre-Condition</b>	Sistem menampilkan halaman utama
Aksi Aktor	Reaksi Sistem
1. Memilih menu <i>master</i> presensi	2. Menampilkan data <i>master</i> presensi
3. Memilih tombol <i>delete</i>	4. Menampilkan pesan konfirmasi hapus
5. Konfirmasi hapus	6. Data presensi di hapus pada <i>database</i>
<b>Post-Condition</b>	Data presensi dihapus pada <i>database</i>

## 11. Spesifikasi Tambah Data User

Tabel 3.13 *Usecase Spesification* Tambah Data User

Use Case Specification	
Nama <i>Use Case</i>	Tambah data user
Aktor	Admin
Deskripsi	Memungkinkan aktor untuk menambah data user pada <i>database</i>
Skenario Utama	
<b>Pre-Condition</b>	Sistem menampilkan halaman utama
Aksi Aktor	Reaksi Sistem
1. Memilih menu <i>master</i> user	2. Menampilkan data <i>master</i> user
3. Memilih tambah data master user	4. Menampilkan form tambah data
5. Mengisi form tambah data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data user

	yang diinput aktor pada <i>database</i>
	8. Data user baru ditambahkan pada <i>database</i>
<b>Skenario Alternatif</b>	
1. Memilih menu <i>master user</i>	2. Menampilkan data <i>master user</i>
3. Memilih tambah data master user	4. Menampilkan form tambah data
5. Mengisi form tambah data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data user yang diinput aktor pada <i>database</i>
	8. Menampilkan pesan data sudah ada
9. Mengisi form tambah data	
10. Memilih tombol <i>save</i>	11. Data user di tambahkan pada <i>database</i>
<b>Post-Condition</b>	Data user baru ditambahkan pada <i>database</i>

## 12. Spesifikasi *Update Data User*

Tabel 3.14 *Usecase Spesification Update Data User*

<b>Use Case Specification</b>	
Nama <i>Use Case</i>	<i>Update data user</i>
Aktor	Admin
Deskripsi	Memungkinkan aktor untuk mengupdate data user pada <i>database</i>
<b>Skenario Utama</b>	
<b>Pre-Condition</b>	Sistem menampilkan halaman utama
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih menu <i>master user</i>	2. Menampilkan data <i>master user</i>
3. Memilih <i>edit</i> data master user	4. Menampilkan form <i>update</i> data
5. Mengisi form <i>update</i> data	
6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data user yang diinput aktor pada <i>database</i>
	8. Data user di- <i>update</i> pada <i>database</i>
<b>Skenario Alternatif</b>	
1. Memilih menu <i>master user</i>	2. Menampilkan data <i>master user</i>
3. Memilih edit data master user	4. Menampilkan form <i>update</i> data
5. Mengisi form <i>update</i> data	



6. Memilih tombol <i>save</i>	7. Cek (apakah data sudah ada) data user yang diinput aktor pada <i>database</i>
	8. Menampilkan pesan data sudah ada
9. Mengisi form <i>update</i> data	
10. Memilih tombol <i>save</i>	11. Data user di- <i>update</i> pada <i>database</i>
<b>Post-Condition</b>	Data user di- <i>update</i> pada <i>database</i>

### 13. Spesifikasi Hapus Data User

Tabel 3.15 *Usecase Spesification Hapus Data User*

Use Case Specification	
Nama <i>Use Case</i>	Hapus data user
Aktor	Admin
Deskripsi	Memungkinkan aktor untuk menghapus data user pada <i>database</i>
Skenario Utama	
<b>Pre-Condition</b>	Sistem menampilkan halaman utama
Aksi Aktor	Reaksi Sistem
1. Memilih menu <i>master</i> user	2. Menampilkan data <i>master</i> user
3. Memilih tombol <i>delete</i>	4. Menampilkan pesan konfirmasi hapus
5. Konfirmasi hapus	6. Data user di hapus pada <i>database</i>
<b>Post-Condition</b>	Data user dihapus pada <i>database</i>

### 14. Spesifikasi MonitorPeta *Work From Home*

Tabel 3.16 *Usecase Spesification Monitor Peta Work From Home*

Use Case Specification	
Nama <i>Use Case</i>	Monitor peta <i>work from home</i>
Aktor	<i>User</i>
Deskripsi	Memungkinkan aktor untuk melihat peta sebaran karyawan yang sedang melakukan <i>work from home</i> (bekerja dari rumah)
Skenario Utama	

<b>Pre-Condition</b>	Sistem menampilkan halaman utama
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih menu <i>maps</i>	2. Menampilkan peta sebaran karyawan
<b>Skenario Alternatif</b>	
1. Memilih menu <i>maps</i>	2. Menampilkan peta sebaran karyawan
3. Mengarahkan kursor pada markah peta	4. Menampilkan nama karyawan
<b>Post-Condition</b>	Aktor dapat melihat peta sebaran karyawan yang sedang melakukan <i>work from home</i>

## 15. Spesifikasi Cetak Laporan Presensi

Tabel 3.17 *Usecase Spesification* Cetak Laporan Presensi

Use Case Specification	
Nama <i>Use Case</i>	Cetak laporan presensi
Aktor	<i>User</i>
Deskripsi	Memungkinkan aktor untuk mencetak laporan presensi
<b>Skenario Utama</b>	
<b>Pre-Condition</b>	Sistem menampilkan halaman utama
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih menu <i>master presensi</i>	2. Menampilkan data <i>master presensi</i>
3. Memilih tombol <i>print</i>	4. Mencetak laporan presensi
<b>Skenario Alternatif</b>	
1. Memilih menu <i>master presensi</i>	2. Menampilkan data <i>master presensi</i>
3. Mengisi field tanggal periode presensi yang akan di cetak	
4. Memilih tombol <i>print</i>	5. Mencetak laporan presensi
<b>Post-Condition</b>	Aktor mendapat <i>output</i> berupa laporan presensi

## 16. Spesifikasi Presensi Masuk

Tabel 3.18 *Usecase Spesification* Presensi Masuk

Use Case Specification	
Nama <i>Use Case</i>	Presensi masuk
Aktor	Karyawan
Deskripsi	Memungkinkan aktor untuk melakukan <i>presensi</i> masuk pada sistem
Skenario Utama	
<b>Pre-Condition</b>	Sistem menampilkan kode <i>QR</i>
Aksi Aktor	Reaksi Sistem
1. Membuka aplikasi <i>presensi</i> pada ponsel <i>android</i>	
2. Memilih menu <i>scan presensi</i> masuk	
3. Melakukan <i>scan</i> pada kode <i>QR</i>	4. Mencatat waktu dan tanggal <i>presensi</i> masuk pada <i>database</i>
	5. Menampilkan pesan presensi telah di rekam
Skenario Alternatif	
1. Membuka aplikasi <i>presensi</i> pada ponsel <i>android</i>	
2. Memilih menu <i>scan presensi</i> masuk	
3. Melakukan <i>scan</i> pada kode <i>QR</i>	4. Menampilkan pesan bahwa aktor sudah pernah melakukan presensi masuk
<b>Post-Condition</b>	Sistem mencatat waktu dan tanggal presensi masuk pada <i>database</i>

## 17. Spesifikasi Presensi Pulang

Tabel 3.19 *Usecase Spesification* Presensi Pulang

Use Case Specification	
Nama <i>Use Case</i>	Presensi pulang
Aktor	Karyawan

Deskripsi	Memungkinkan aktor untuk melakukan <i>presensi</i> pulang pada sistem
<b>Skenario Utama</b>	
<b>Pre-Condition</b>	Sistem menampilkan kode <i>QR</i>
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Membuka aplikasi <i>presensi</i> pada ponsel <i>android</i>	
2. Memilih menu <i>scan presensi</i> pulang	
3. Melakukan <i>scan</i> pada kode <i>QR</i>	4. Mencatat waktu dan tanggal <i>presensi</i> pulang pada <i>database</i>
	5. Menampilkan pesan presensi telah di rekam
<b>Skenario Alternatif</b>	
5. Membuka aplikasi <i>presensi</i> pada ponsel <i>android</i>	
6. Memilih menu <i>scan presensi</i> pulang	
7. Melakukan <i>scan</i> pada kode <i>QR</i>	8. Menampilkan pesan bahwa aktor sudah pernah melakukan presensi pulang
<b>Post-Condition</b>	Sistem mencatat waktu dan tanggal presensi pulang pada <i>database</i>

## 18. Spesifikasi Lihat Kartu Presensi

Tabel 3.20 *Usecase Spesification* Lihat Kartu Presensi

Use Case Specification	
Nama <i>Use Case</i>	Lihat kartu presensi
Aktor	Karyawan
Deskripsi	Memungkinkan aktor untuk melihat kartu presensi
<b>Skenario Utama</b>	
<b>Pre-Condition</b>	Aktor membuka aplikasi presensi pada ponsel <i>android</i>
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih menu kartu presensi	2. Menampilkan form isian periode tanggal presensi

3. Mengisi field tanggal periode presensi yang akan di lihat	
4. Memilih tombol <i>show</i>	5. Menampilkan kartu presensi
<b>Post-Condition</b>	Aktor dapat melihat kartu presensi

## 19. Spesifikasi *Logout*

Tabel 3.21 *Usecase Spesification Logout*

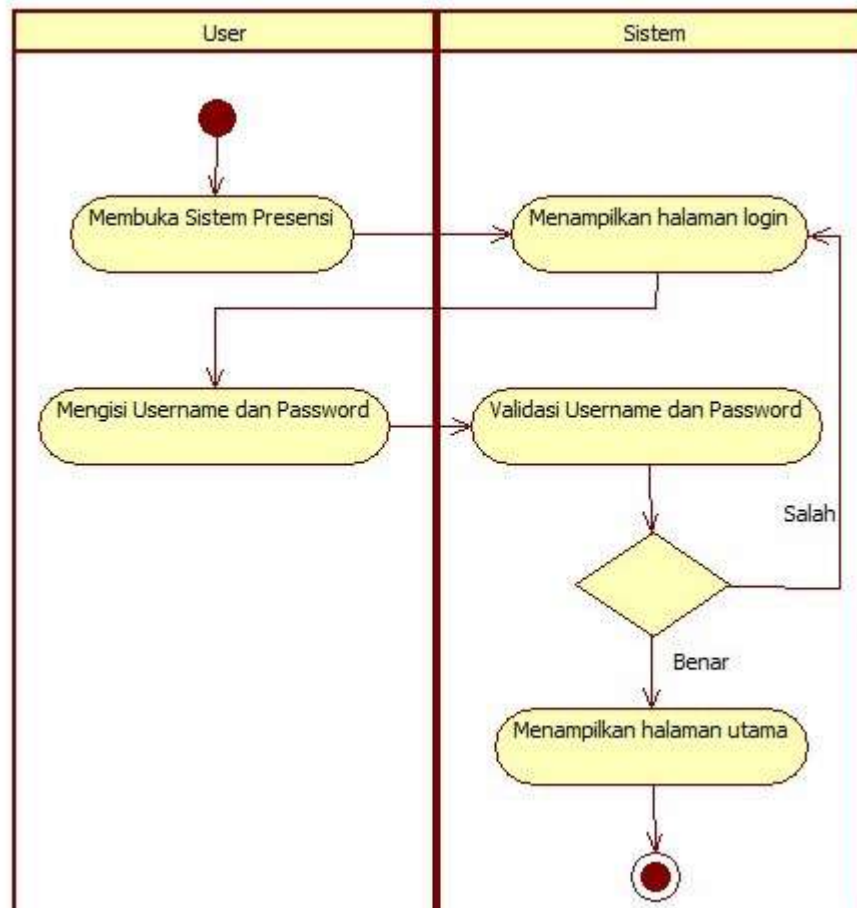
Use Case Specification	
Nama <i>Use Case</i>	Logout
Aktor	<i>User</i>
Deskripsi	Memungkinkan aktor untuk keluar dari layanan sistem
Skenario Utama	
<b>Pre-Condition</b>	Aktor dalam keadaan <i>login</i>
Aksi Aktor	Reaksi Sistem
1. Memilih menu <i>logout</i>	2. Menampilkan pesan konfirmasi untuk <i>logout</i>
3. Memilih “Yakin”	4. Menampilkan halaman login
Skenario Alternatif	
1. Memilih menu <i>logout</i>	2. Menampilkan pesan konfirmasi untuk <i>logout</i>
3. Memilih “Tidak”	4. Halaman utama sistem ditampilkan
<b>Post-Condition</b>	Aktor keluar dari layanan sistem

### 3.5.3 *Activity Diagram*

*Activity Diagram* adalah representasi grafis dari seluruh tahapan alur kerja.

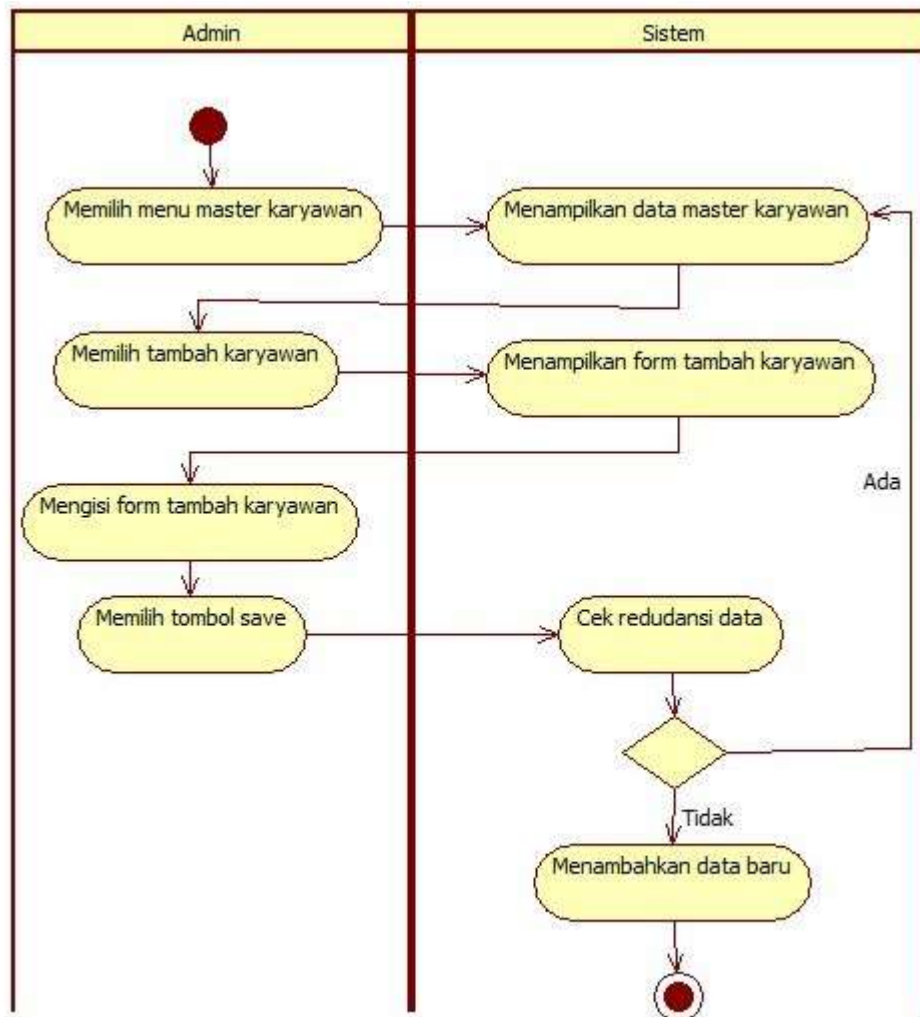
Diagram ini mengandung aktivitas, pilihan tindakan, perulangan, dan hasil dari aktifitas tersebut. Berikut adalah beberapa *Activity Diagram* dari sistem yang diusulkan.

### 1. Activity Diagram Login



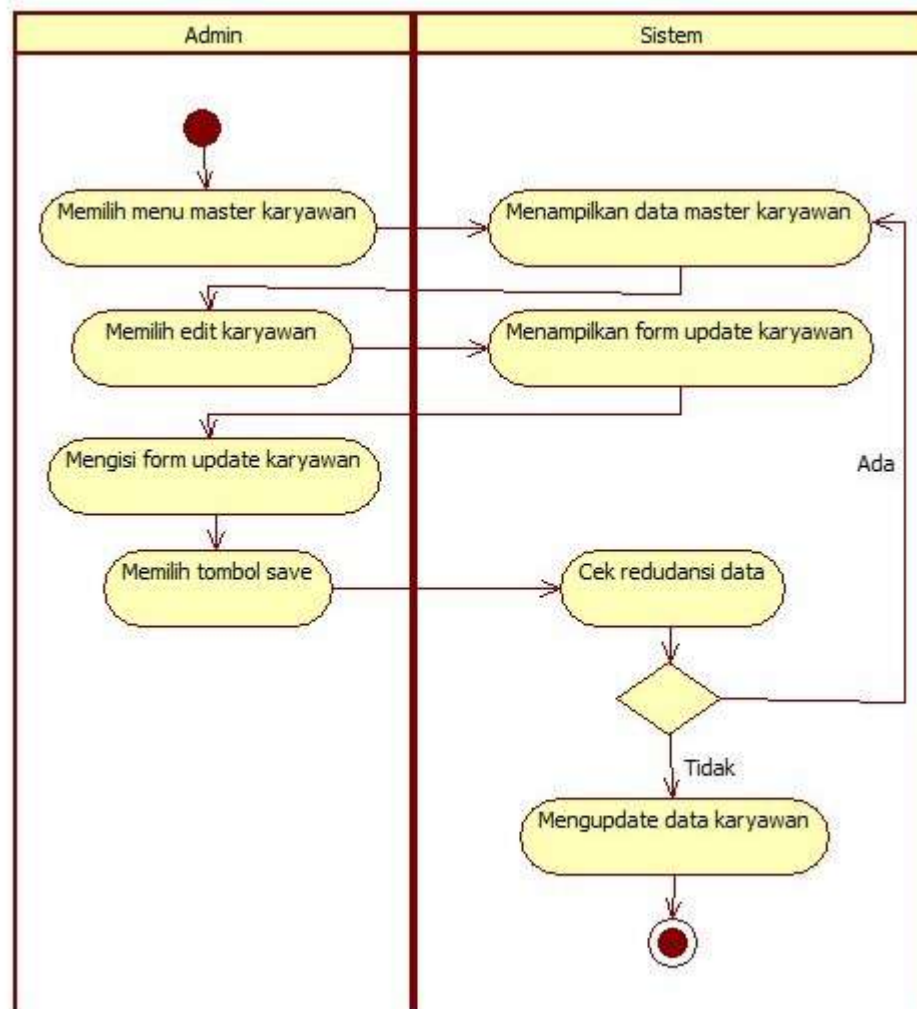
Gambar 3.4 Activity Diagram Login

## 2. Activity Diagram Tambah Data Karyawan



Gambar 3.5 Activity Diagram Tambah Data Karyawan

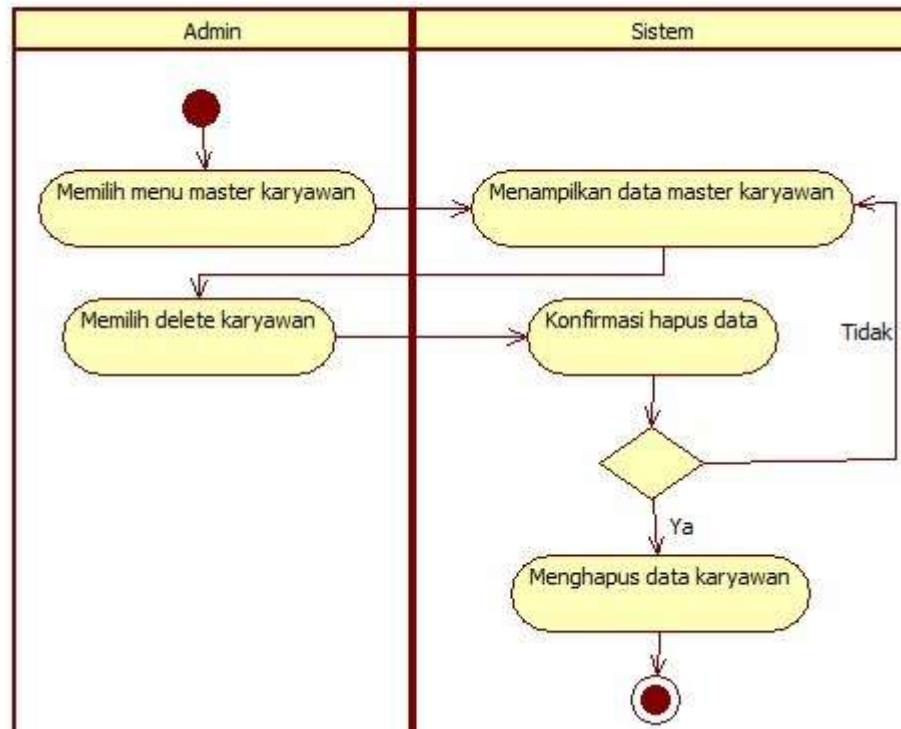
### 3. Activity Diagram Update Data Karyawan



Gambar 3.6 Activity Diagram Update Data Karyawan

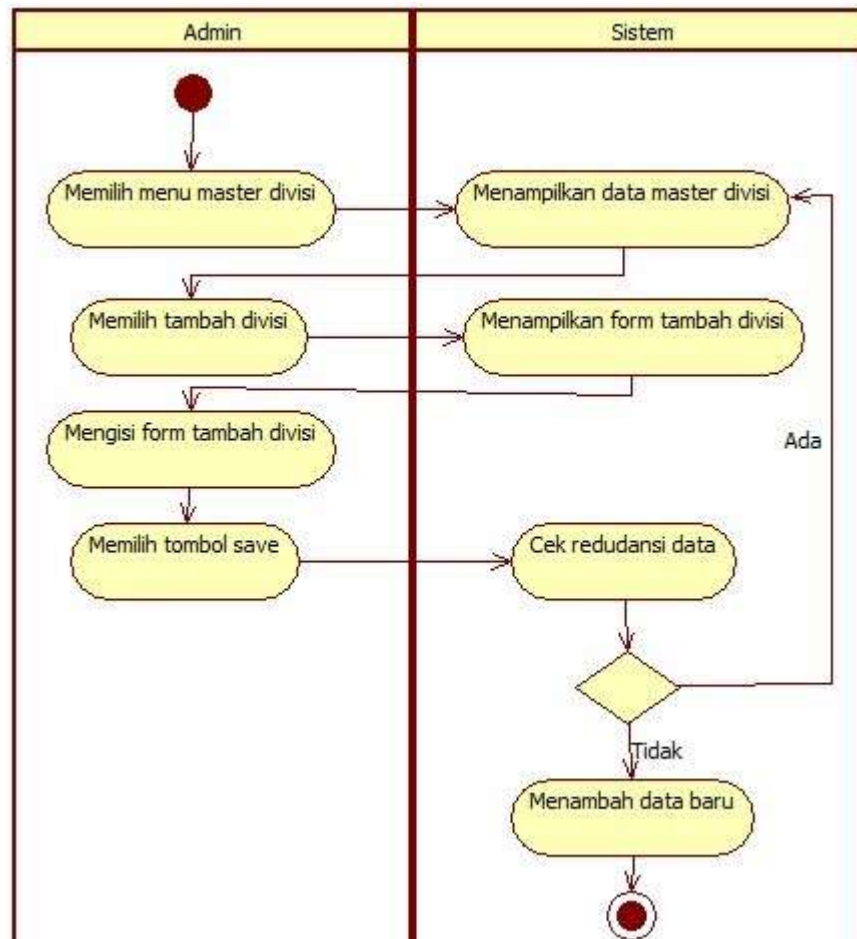


#### 4. Activity Diagram Hapus Data Karyawan



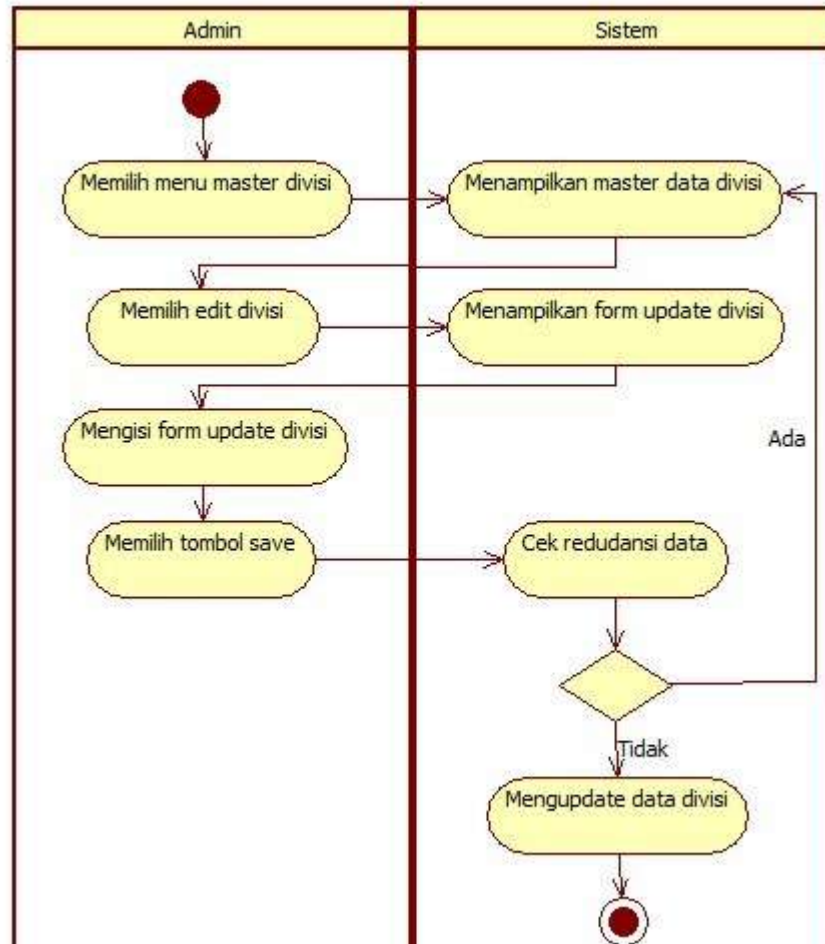
Gambar 3.7 Activity Diagram Hapus Data Karyawan

### 5. Activity Diagram Tambah Data Divisi



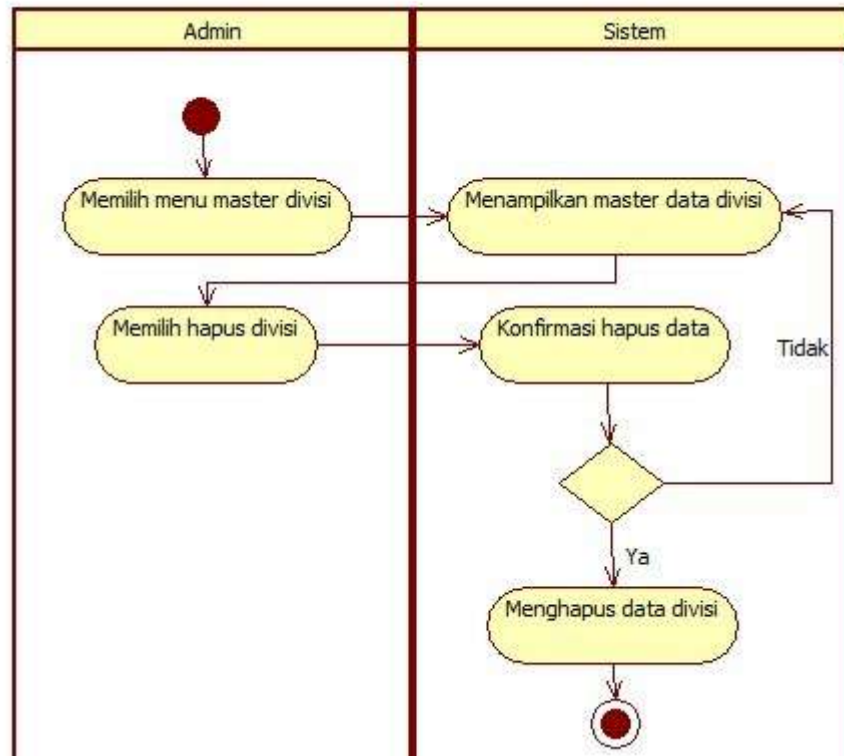
Gambar 3.8 Activity Diagram Tambah Data Divisi

## 6. Activity Diagram Update Data Divisi



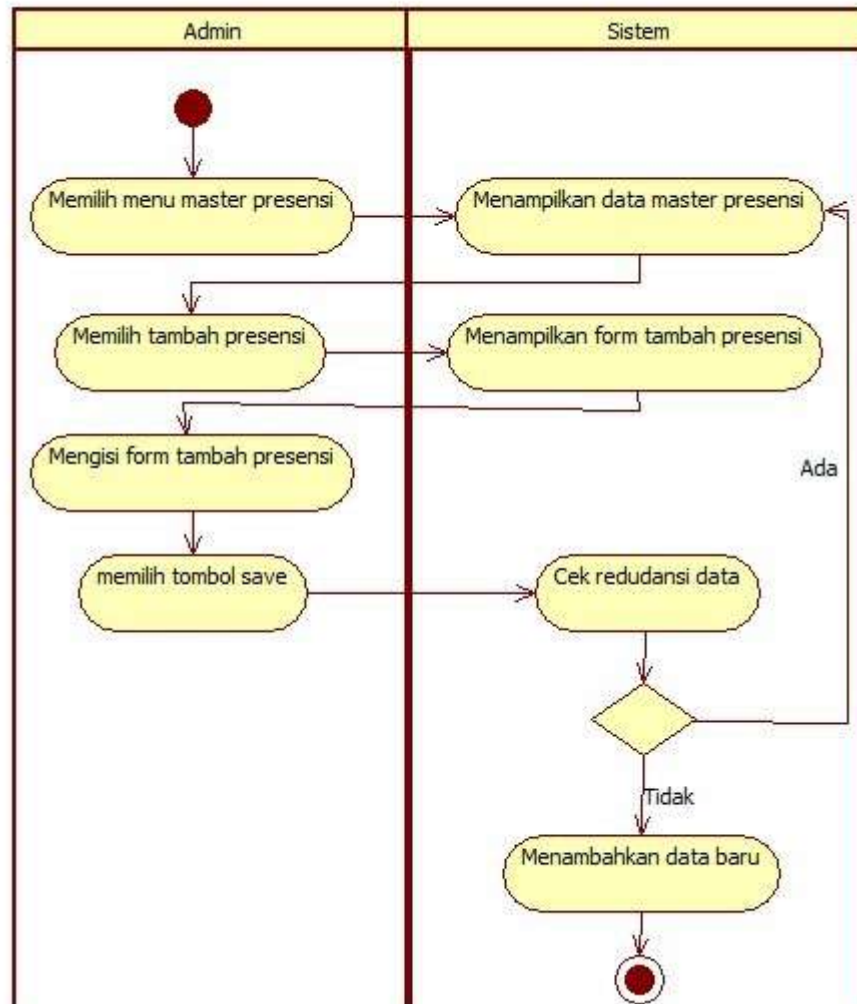
Gambar 3.9 Activity Diagram Update Data Divisi

### 7. Activity Diagram Hapus Data Divisi



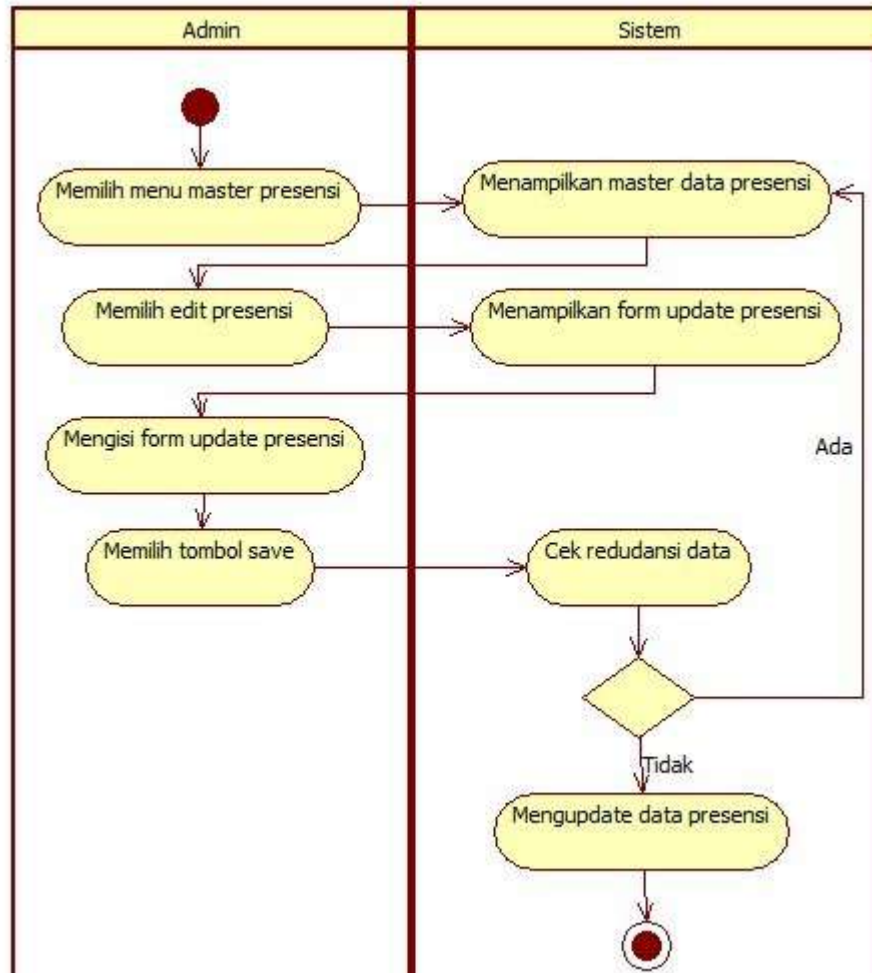
Gambar 3.10 Activity Diagram Hapus Data Divisi

### 8. Activity Diagram Tambah Data Presensi



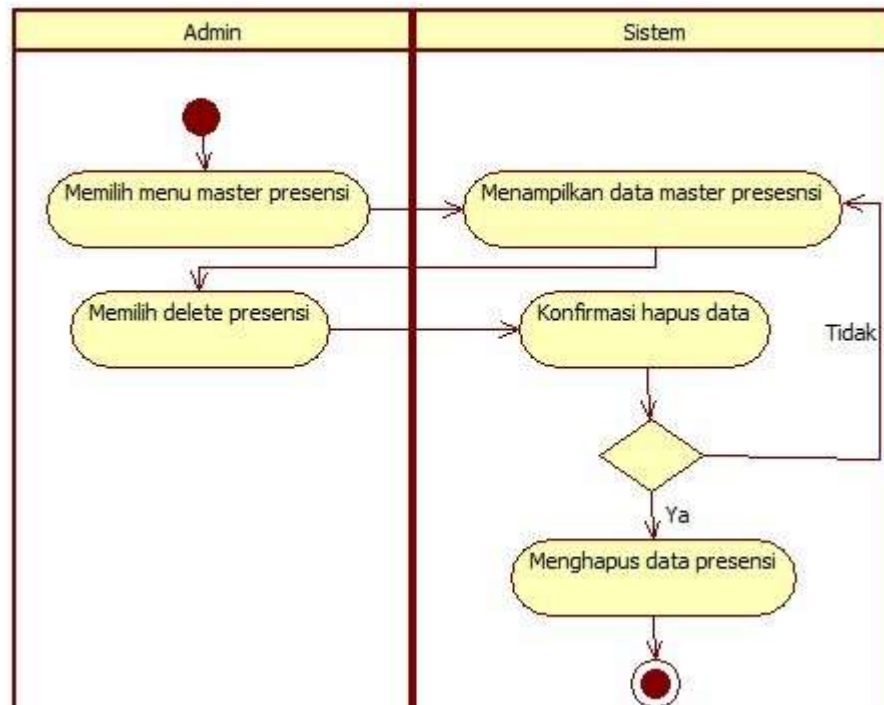
Gambar 3.11 Activity Diagram Tambah Data Presensi

### 9. Activity Diagram Update Data Presensi



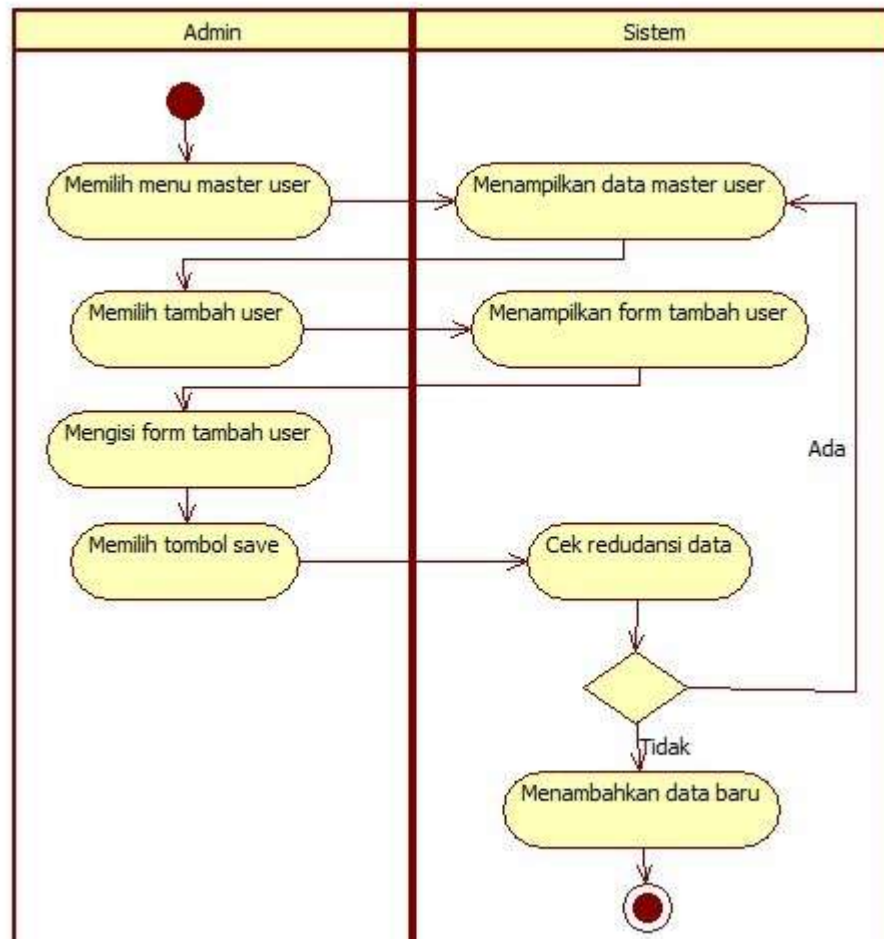
Gambar 3.12 Activity Diagram Update Data Presensi

### 10. Activity Diagram Hapus Data Presensi



Gambar 3.13 Activity Diagram Hapus Data Presensi

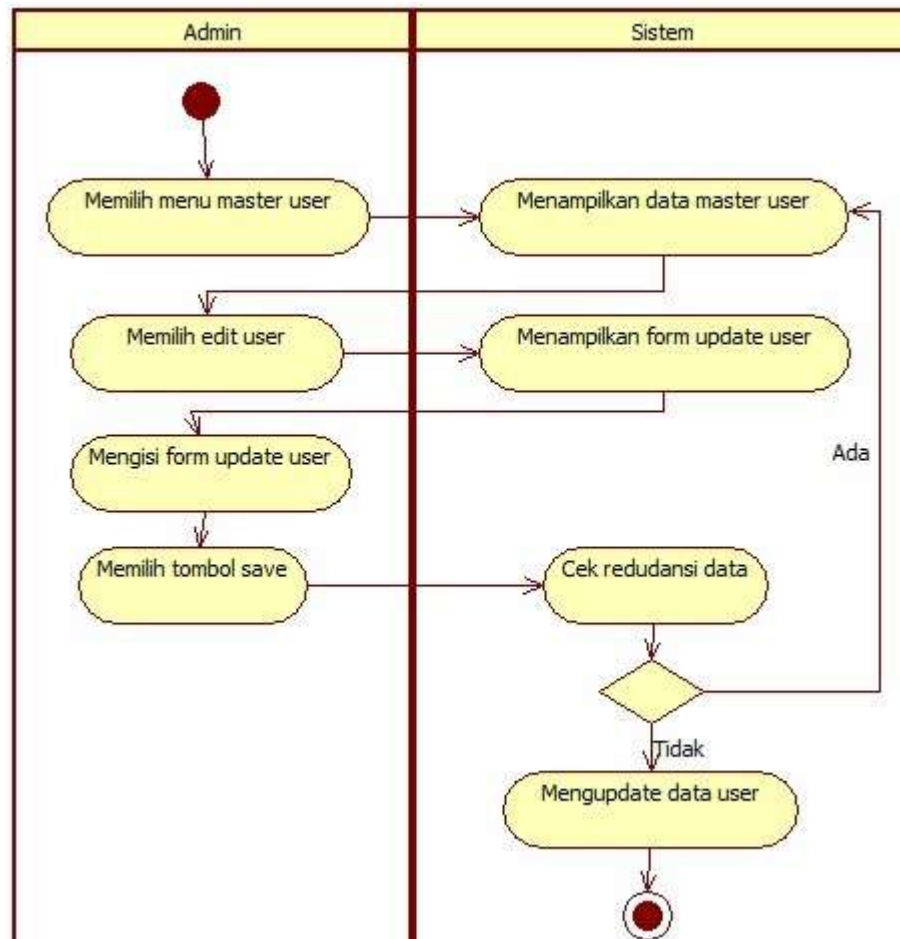
### 11. Activity Diagram Tambah Data User



Gambar 3.14 Activity Diagram Tambah Data User

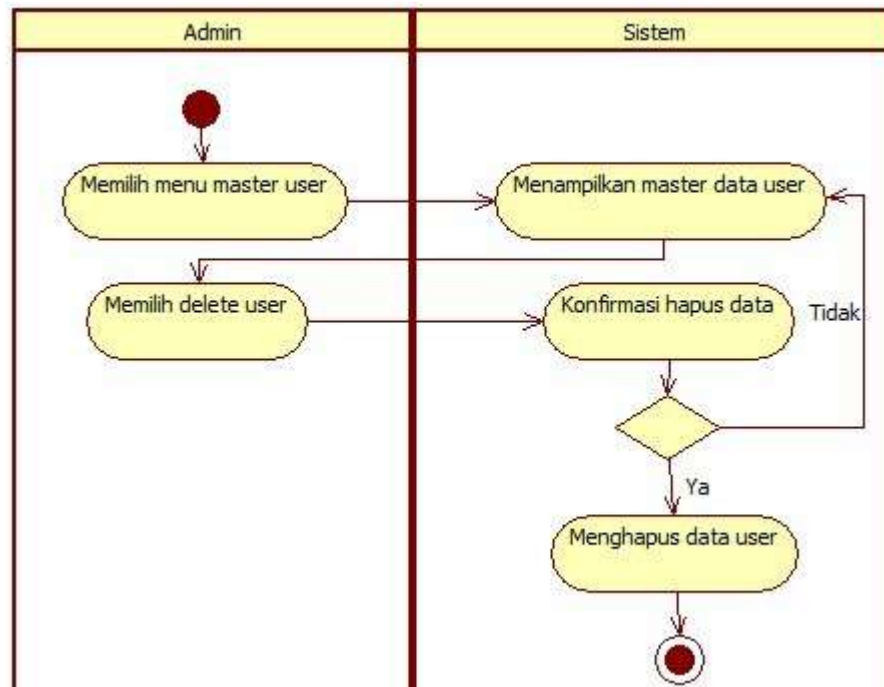


## 12. Activity Diagram Update Data User



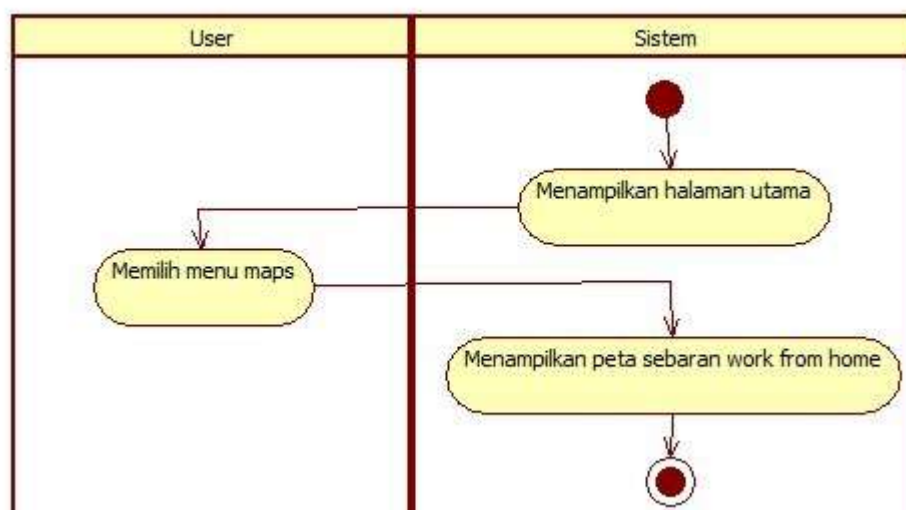
Gambar 3.15 Activity Diagram Update Data User

### 13. Activity Diagram Hapus Data User



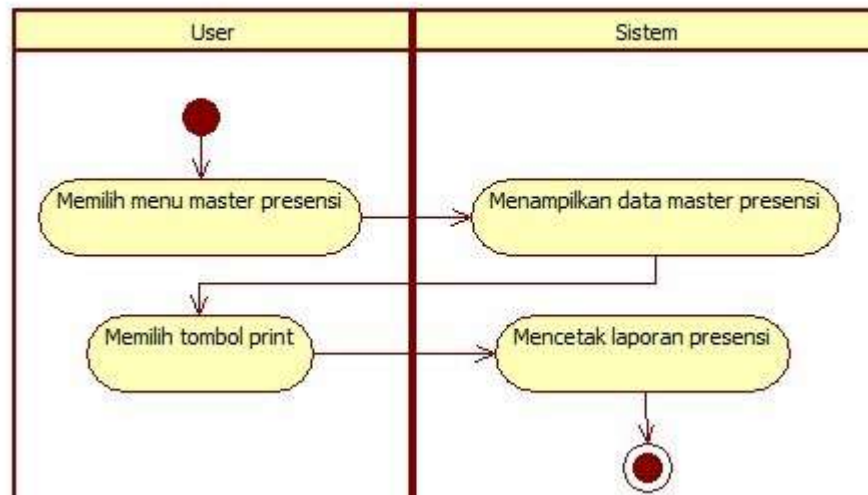
Gambar 3.16 Activity Diagram Hapus Data User

### 14. Activity Diagram Monitor Peta Work From Home



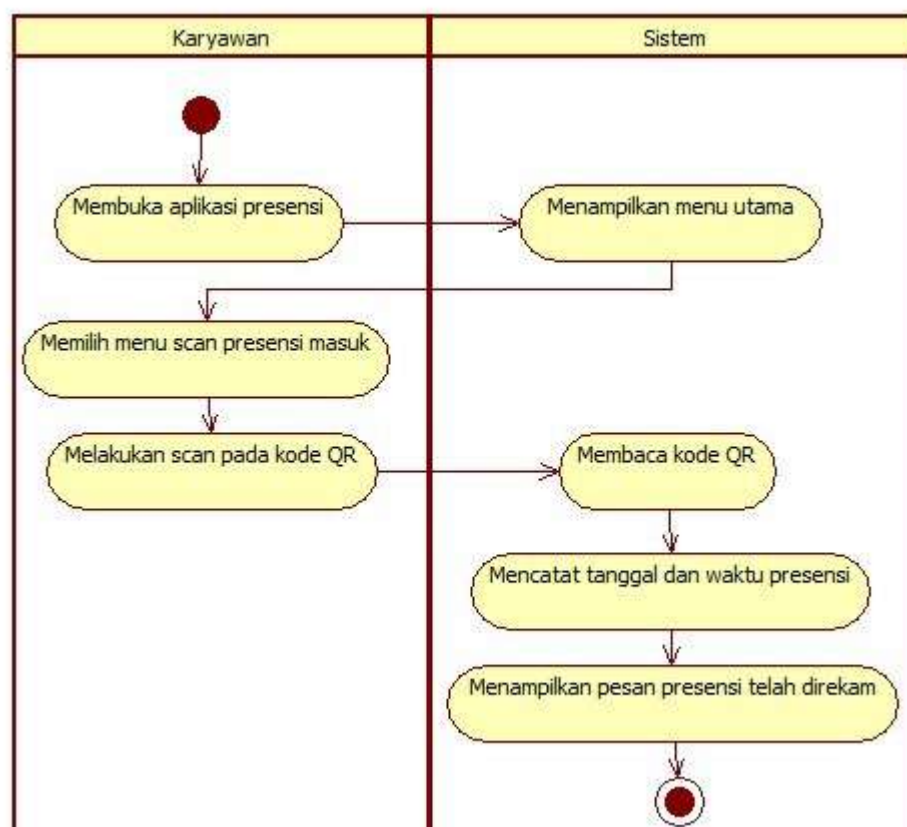
Gambar 3.17 Activity Diagram Monitor Peta Work From Home

### 15. Activity Diagram Cetak Laporan Presensi



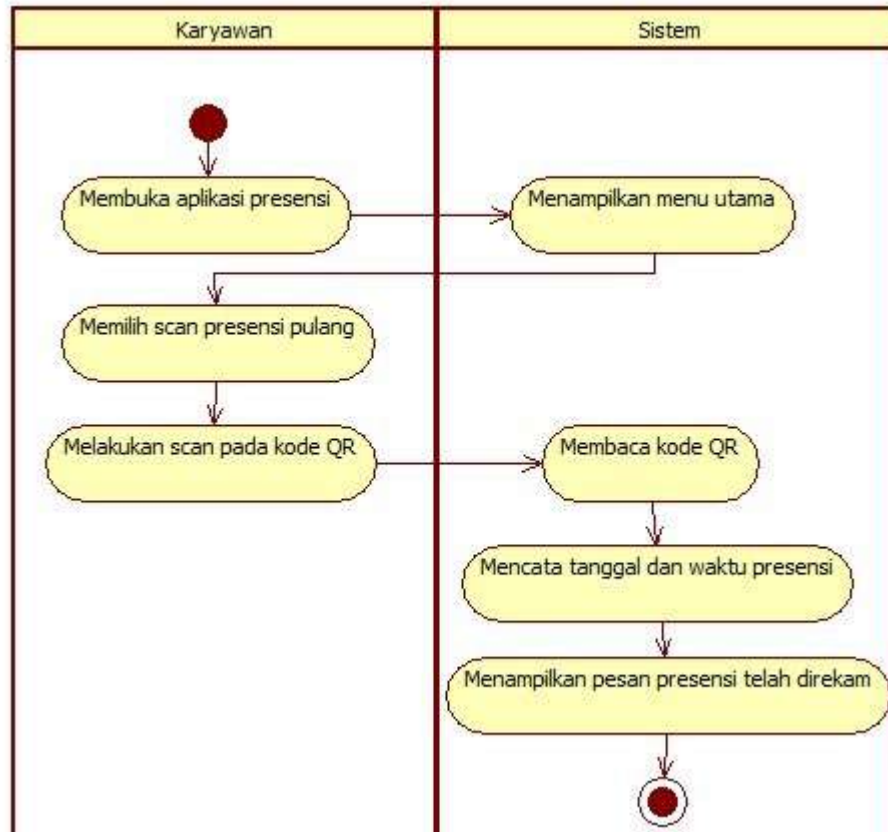
Gambar 3.18 Activity Diagram Cetak Laporan Presensi

### 16. Activity Diagram Presensi Masuk



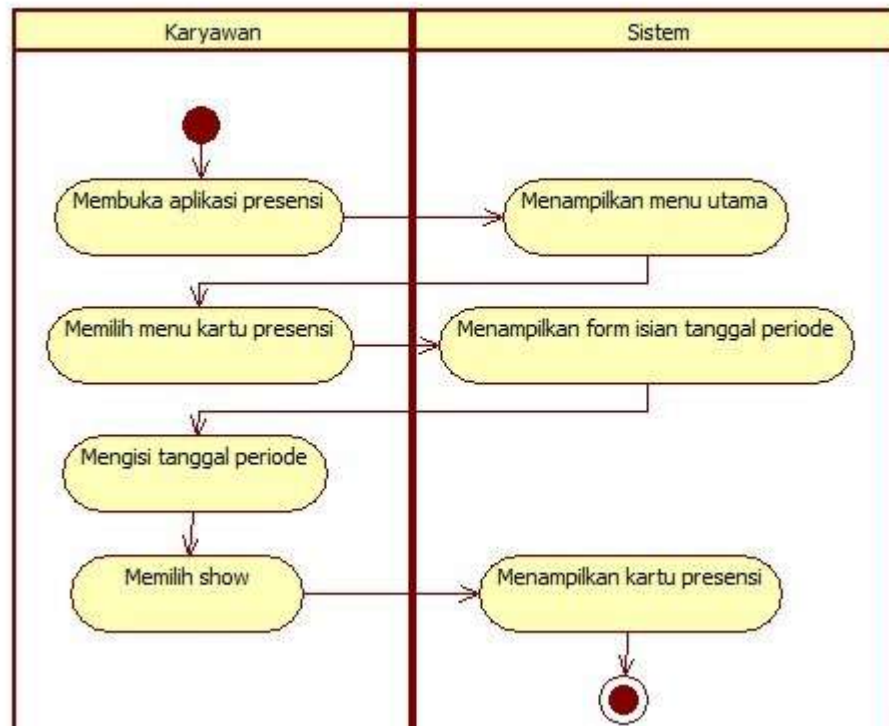
Gambar 3.19 Activity Diagram Presensi Masuk

### 17. Activity Diagram Presensi Pulang



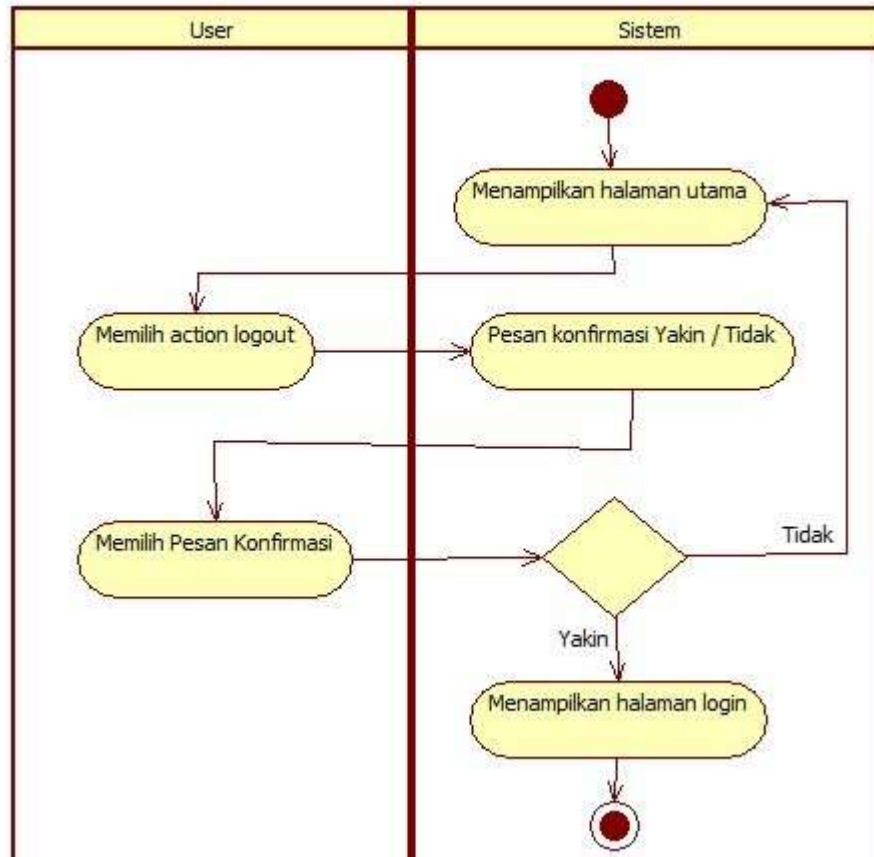
Gambar 3.20 Activity Diagram Presensi Pulang

### 18. Activity Diagram Lihat Kartu Presensi



Gambar 3.21 Activity Diagram Lihat Kartu Presensi

### 19. Activity Diagram Logout

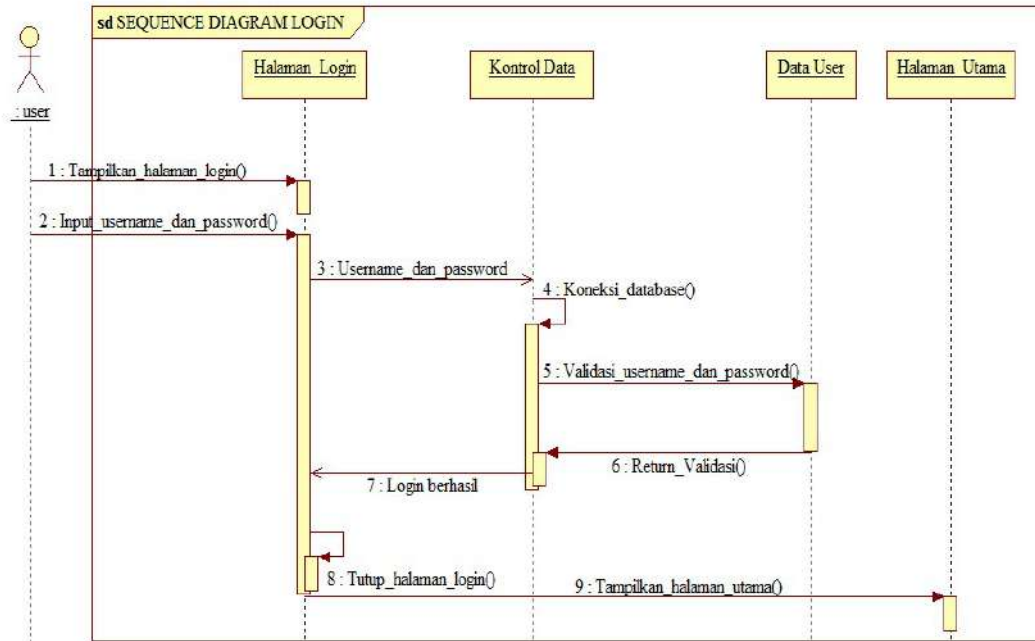


Gambar 3.22 Activity Diagram Logout

#### 3.5.4 Sequence Diagram

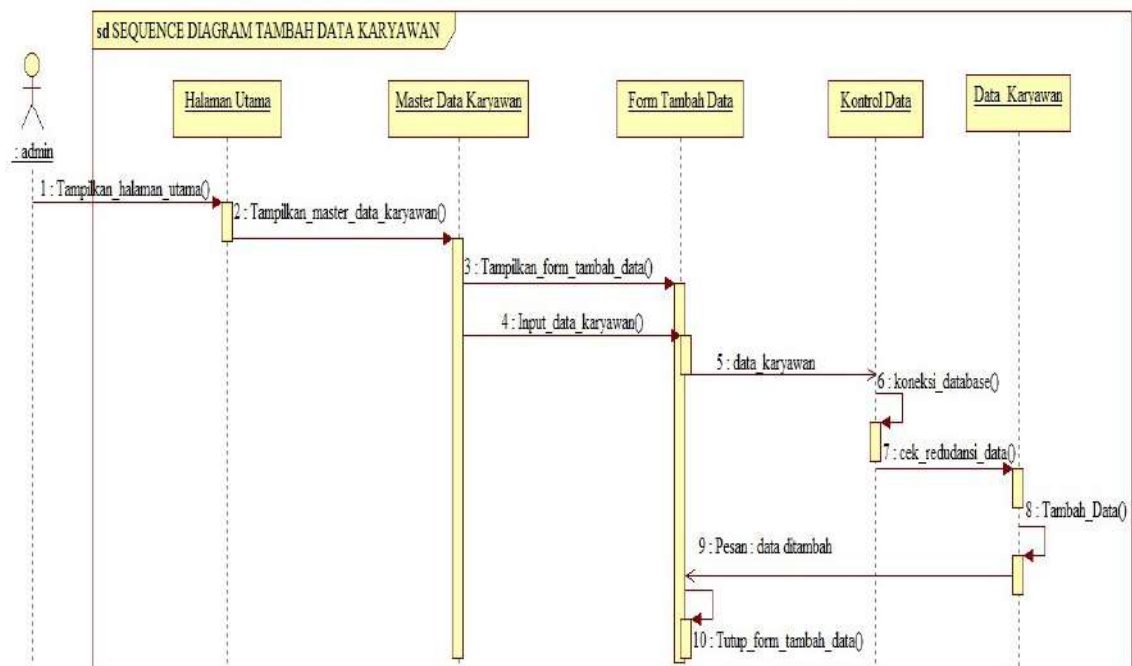
*Sequence diagram* menggambarkan perilaku pada sebuah skenario. *Sequence diagram* memperlihatkan interaksi antar objek di dalam sistem yang disusun pada sebuah urutan waktu. Interaksi antar objek termasuk aktor, display, dan sebagainya berupa pesan atau *Message*. Berikut adalah beberapa *Sequence Diagram* dari sistem yang diusulkan.

## 1. Sequence Diagram Login



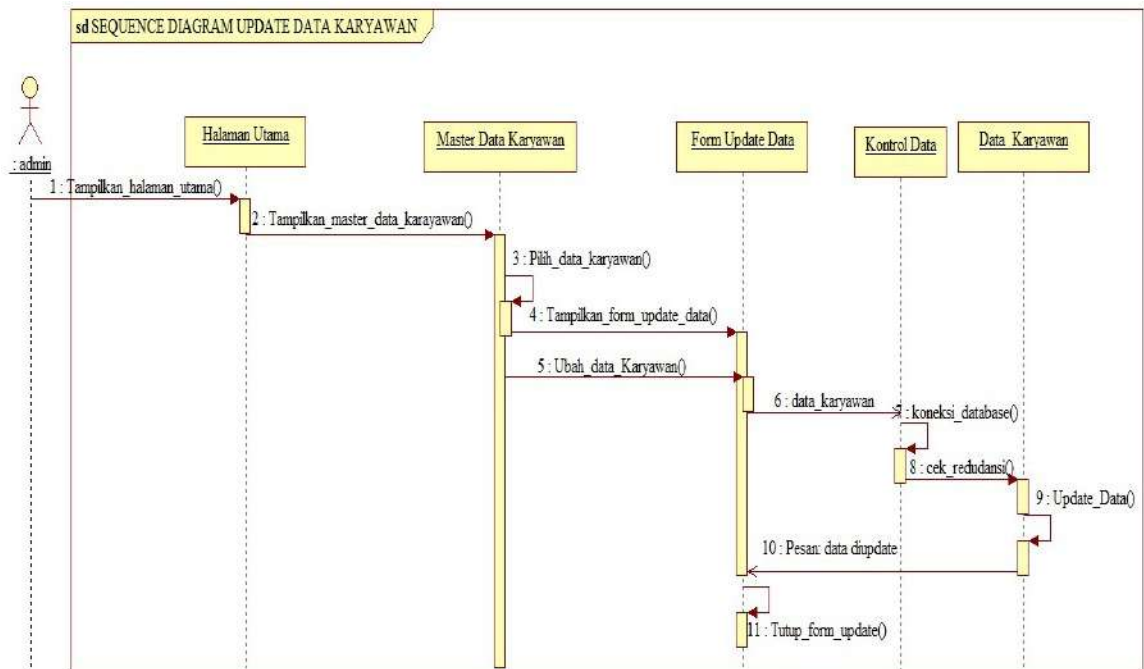
Gambar 3.23 Sequence Diagram Login

## 2. Sequence Diagram Tambah Data Karyawan



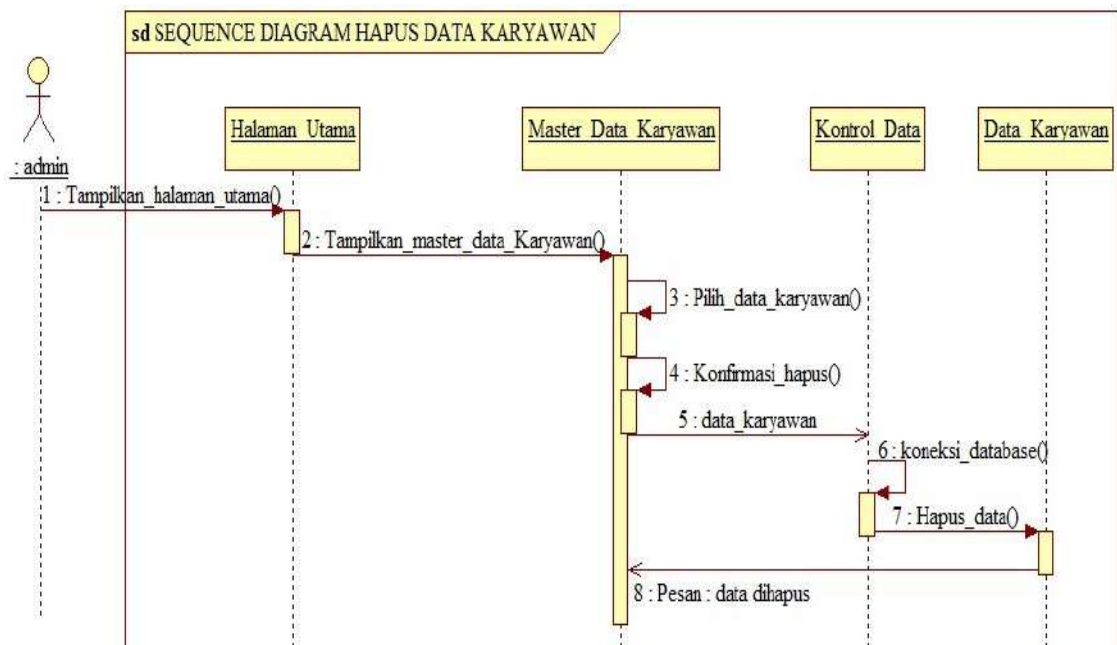
Gambar 3.24 Sequence Diagram Tambah Data Karyawan

### 3. Sequence Diagram Update Data Karyawan



Gambar 3.25 Sequence Diagram Update Data Karyawan

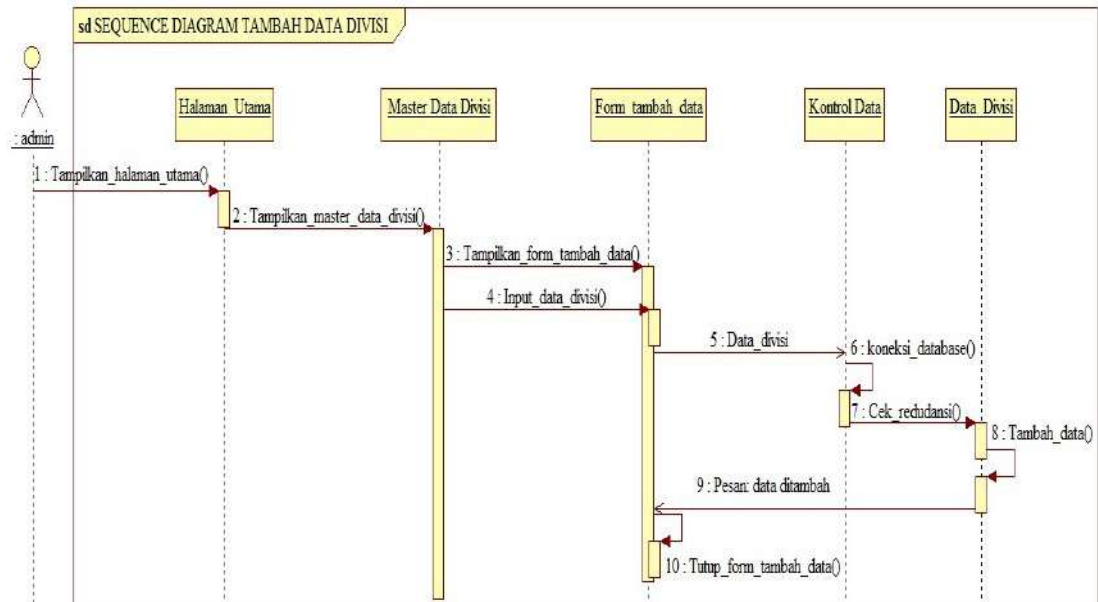
### 4. Sequence Diagram Hapus Data Karyawan



Gambar 3.26 Sequence Diagram Hapus Data Karyawan

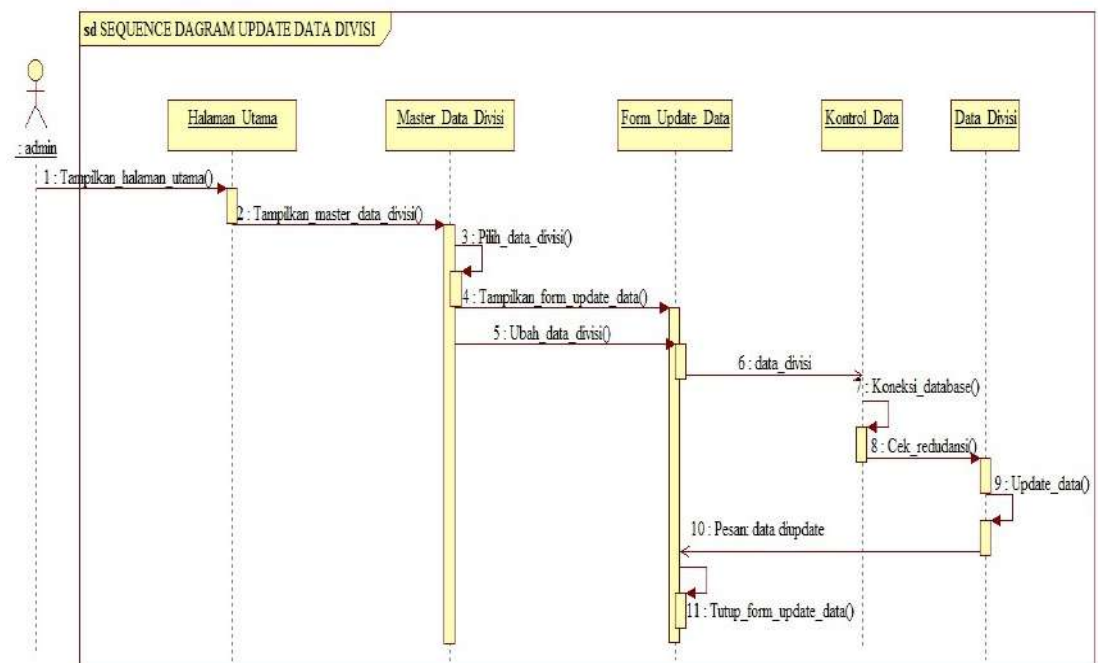


### 5. Sequence Diagram Tambah Data Divisi



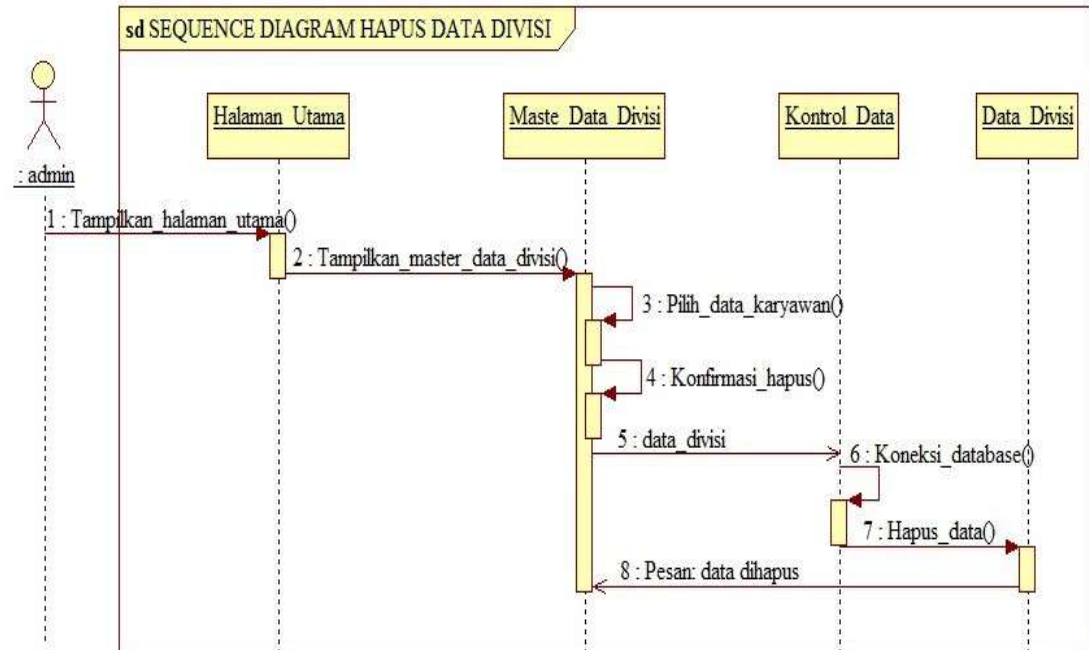
Gambar 3.27 Sequence Diagram Tambah Data Divisi

### 6. Sequence Diagram Update Data Divisi



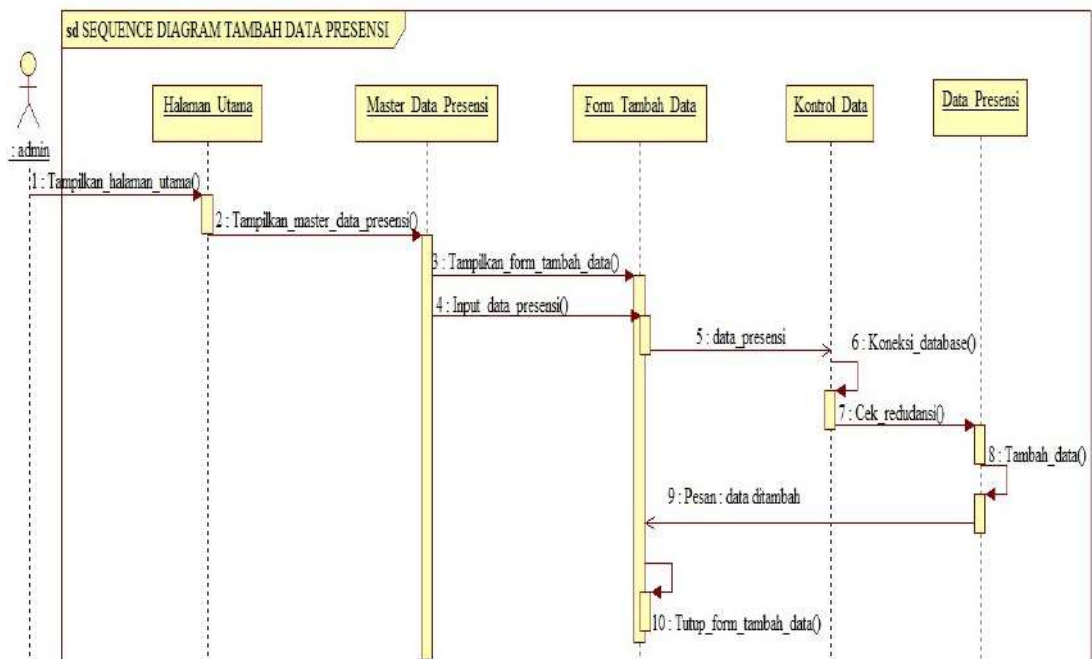
Gambar 3.28 Sequence Diagram Update Data Divisi

## 7. Sequence Diagram Hapus Data Divisi



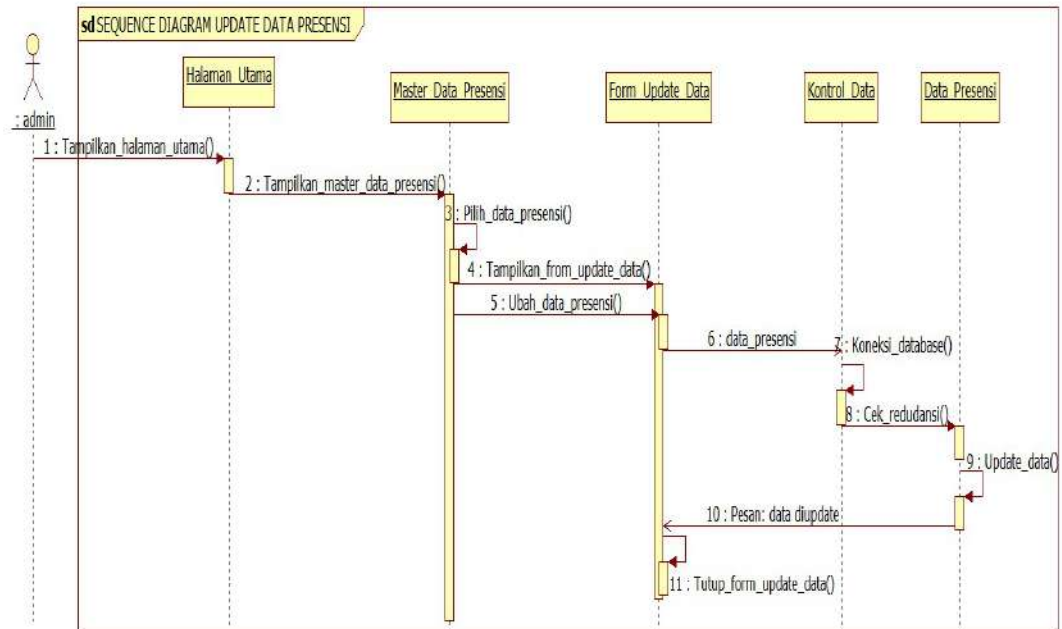
Gambar 3.29 Sequence Diagram Hapus Data Divisi

## 8. Sequence Diagram Tambah Data Presensi



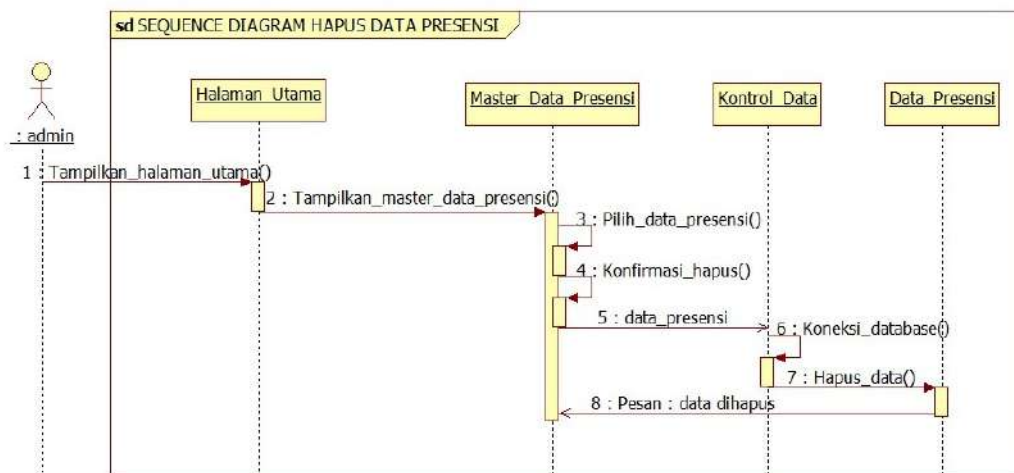
Gambar 3.30 Sequence Diagram Tambah Data Presensi

### 9. Sequence Diagram Update Data Presensi



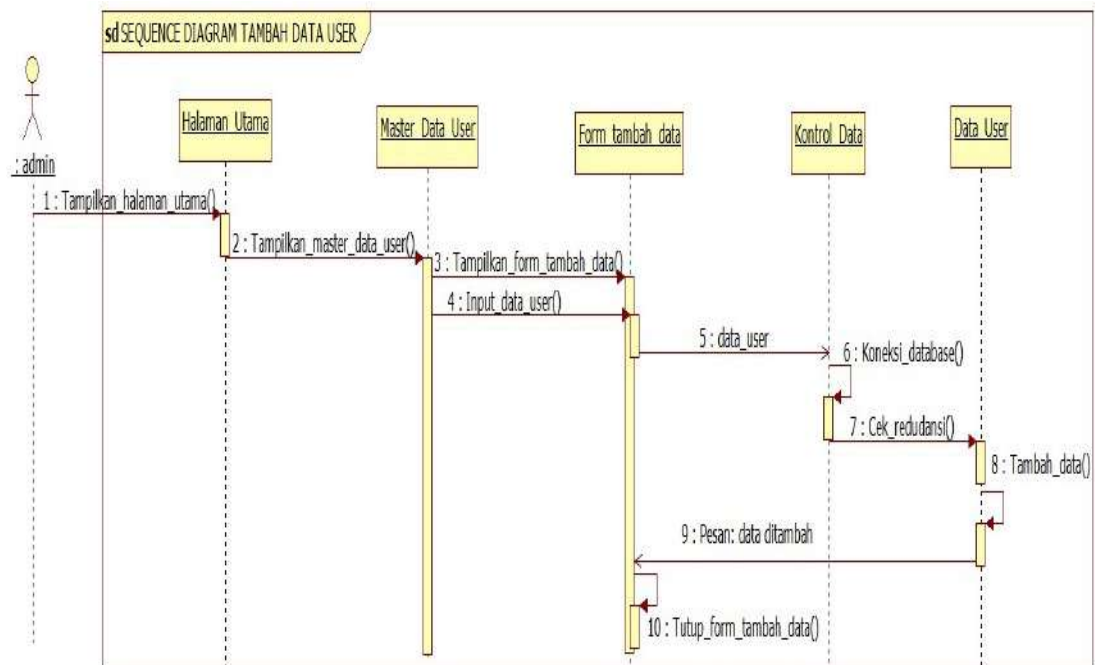
Gambar 3.31 Sequence Diagram Update Data Presensi

### 10. Sequence Diagram Hapus Data Presensi



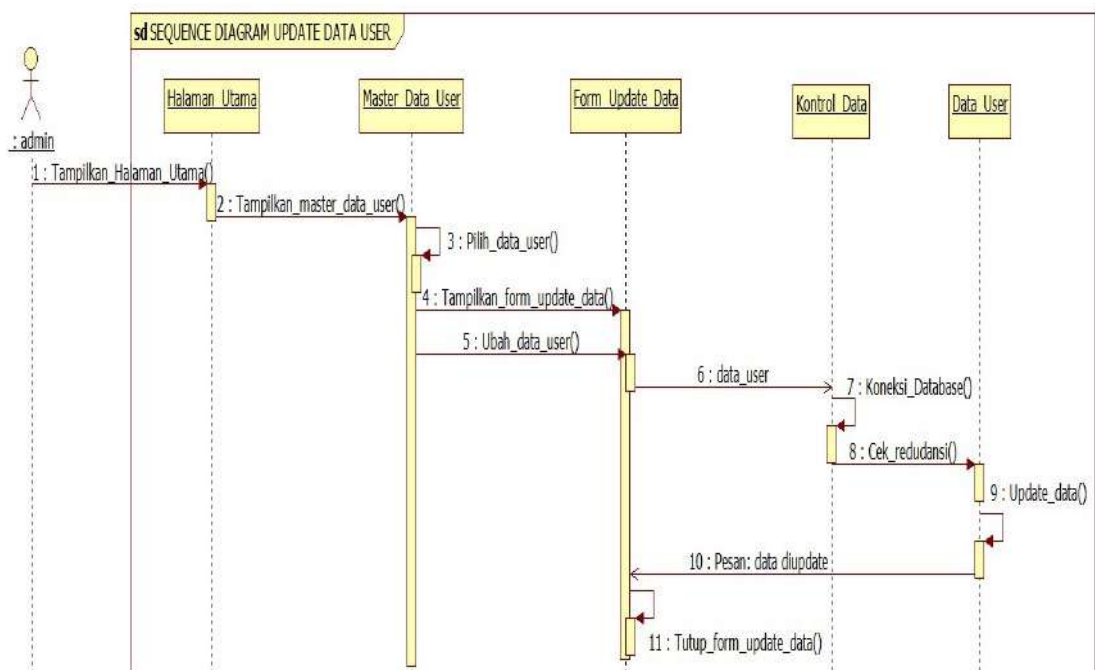
Gambar 3.32 Sequence Diagram Hapus Data Presensi

## 11. Sequence Diagram Tambah Data User



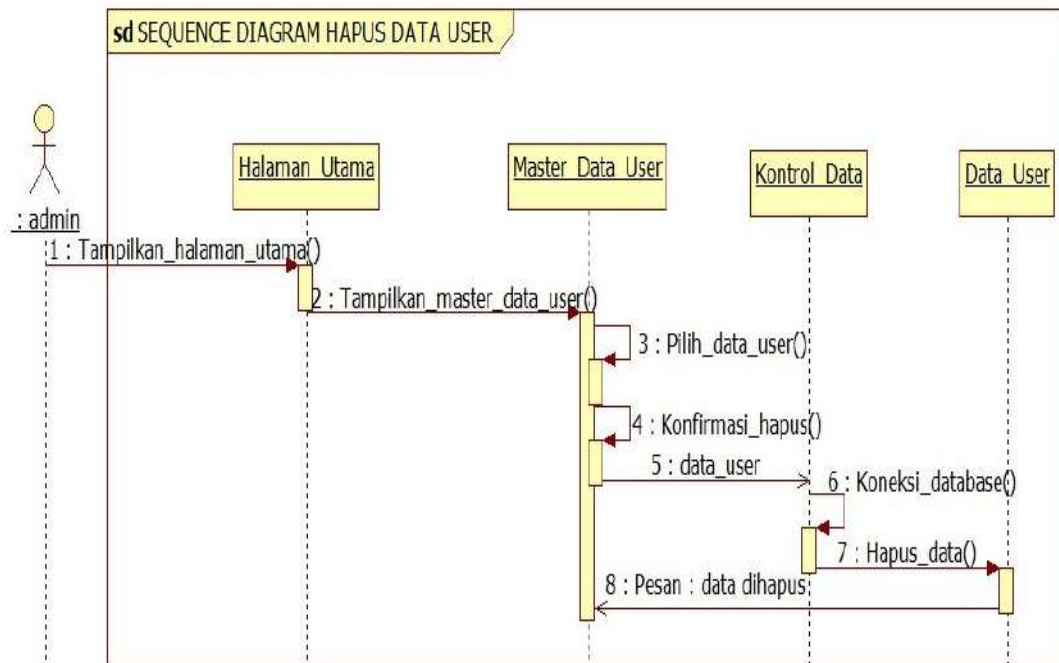
Gambar 3.33 Activity Diagram Tambah Data User

## 12. Sequence Diagram Update Data User



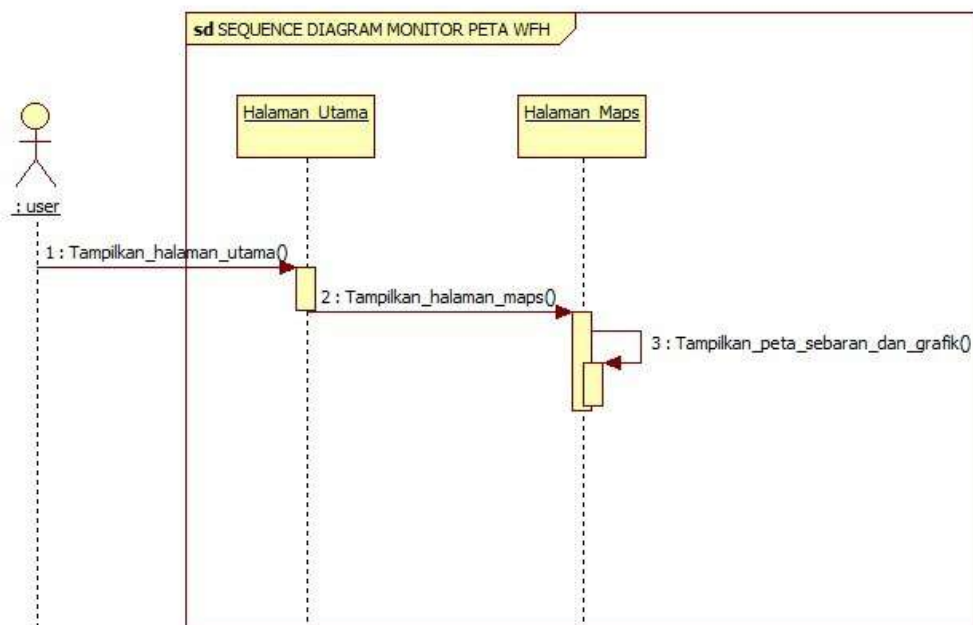
Gambar 3.34 Sequence Diagram Update Data User

### 13. Sequence Diagram Hapus Data User



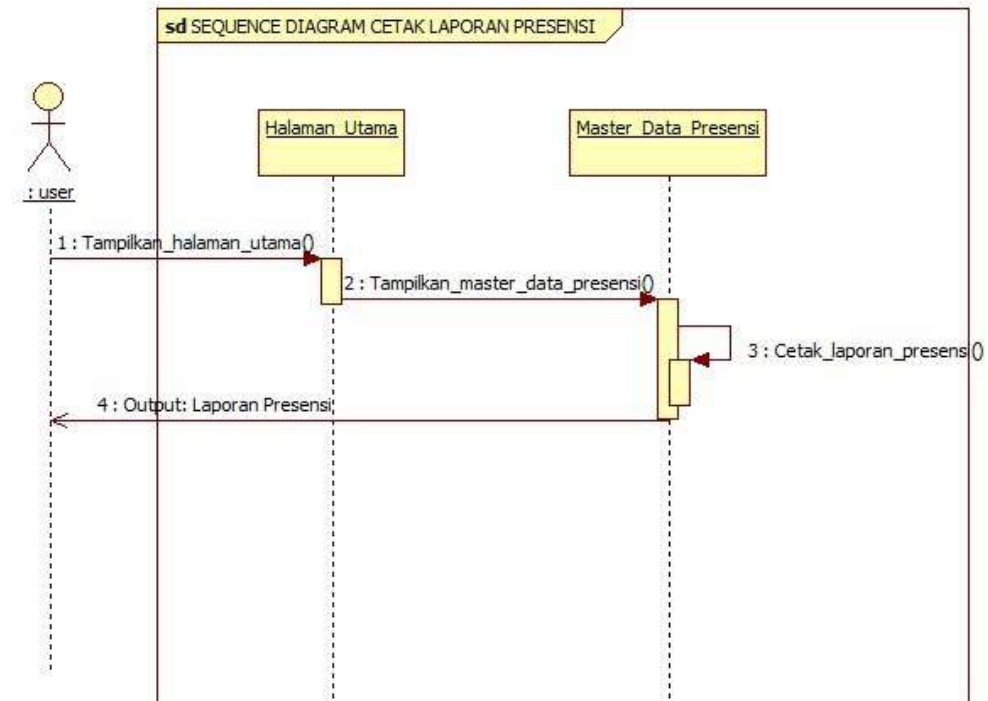
Gambar 3.35 Sequence Diagram Hapus Data User

### 14. Sequence Diagram Monitor Peta Work From Home



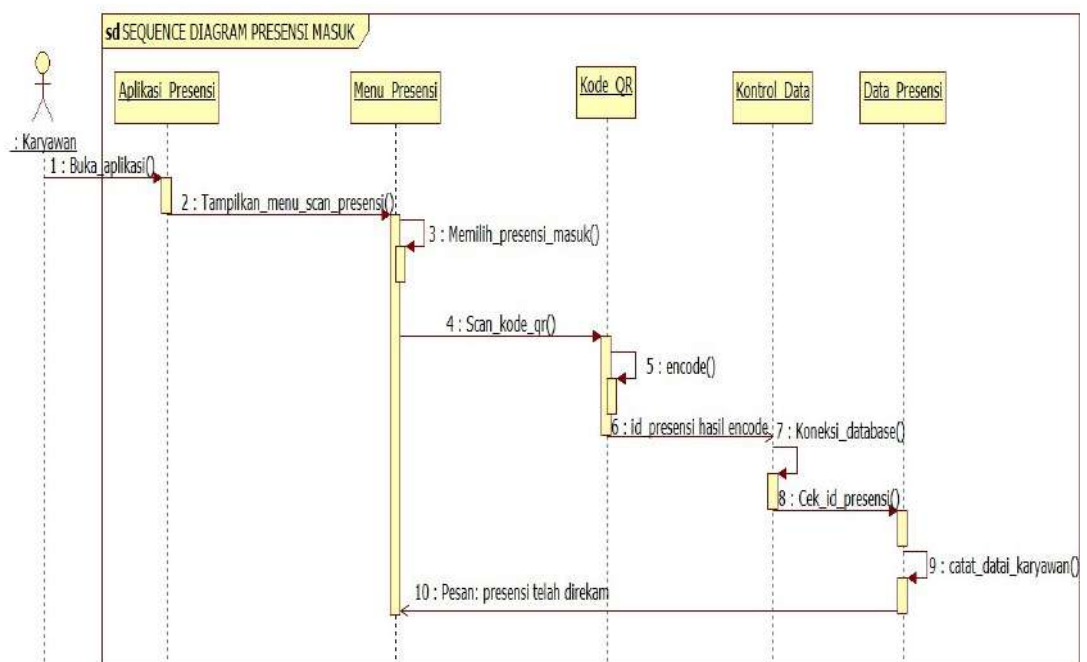
Gambar 3.36 Sequence Diagram Monitor Peta Work From Home

### 15. Sequence Diagram Cetak Laporan Presensi



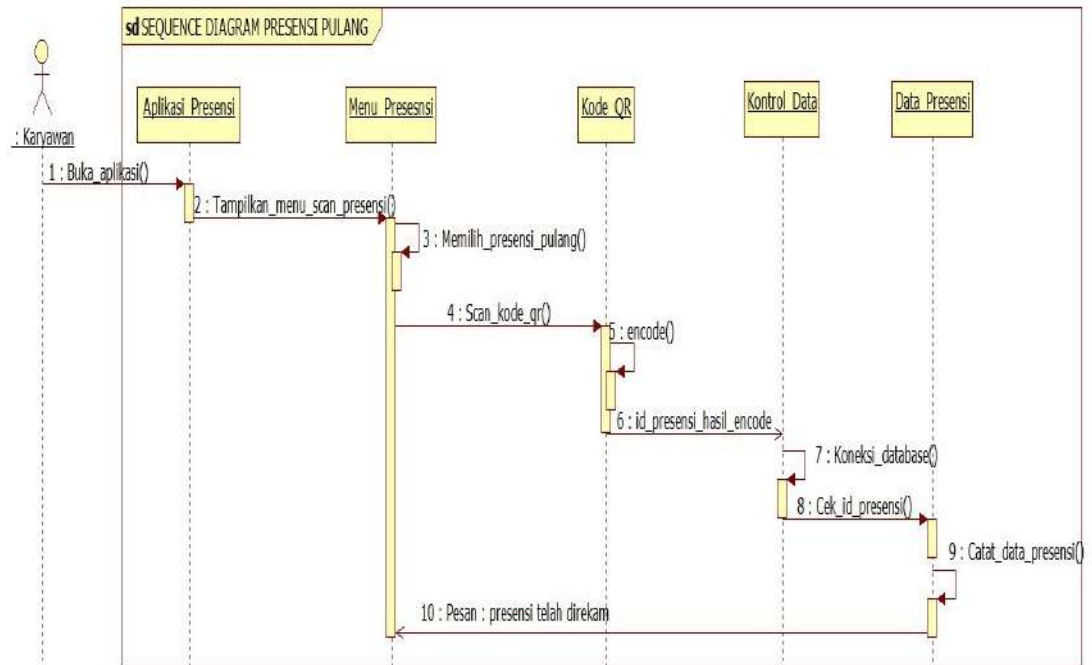
Gambar 3.37 Sequence Diagram Cetak Laporan Presensi

### 16. Sequence Diagram Presensi Masuk



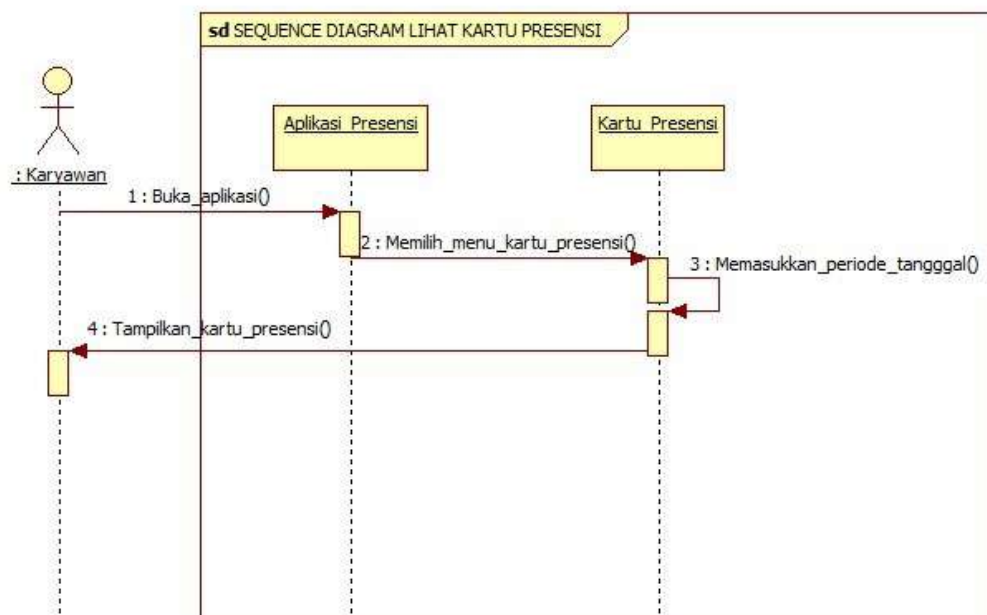
Gambar 3.38 Sequence Diagram Presensi Masuk

### 17. Sequence Diagram Presensi Pulang



Gambar 3.39 Sequence Diagram Presensi Pulang

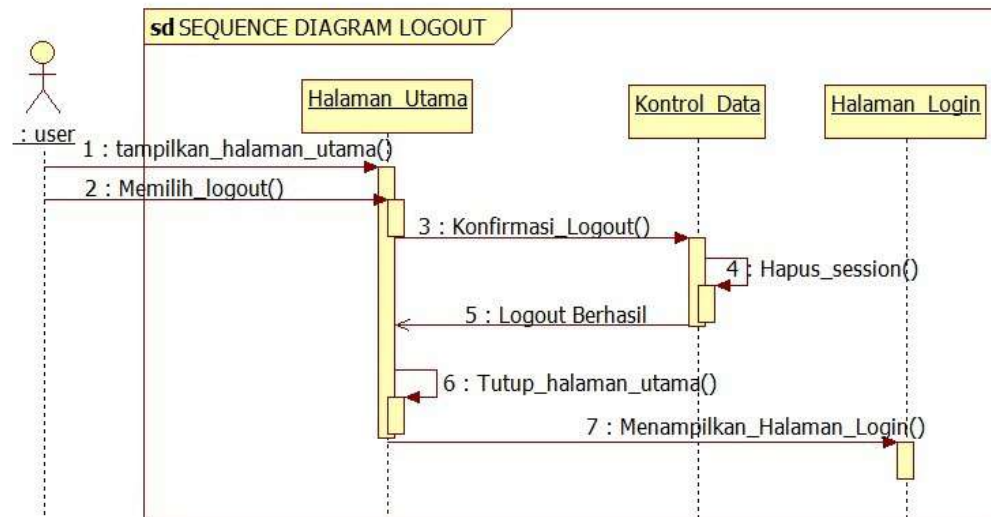
### 18. Sequence Diagram Lihat Kartu Presensi



Gambar 3.40 Sequence Diagram Lihat Kartu Presensi



### 19. Sequence Diagram Logout

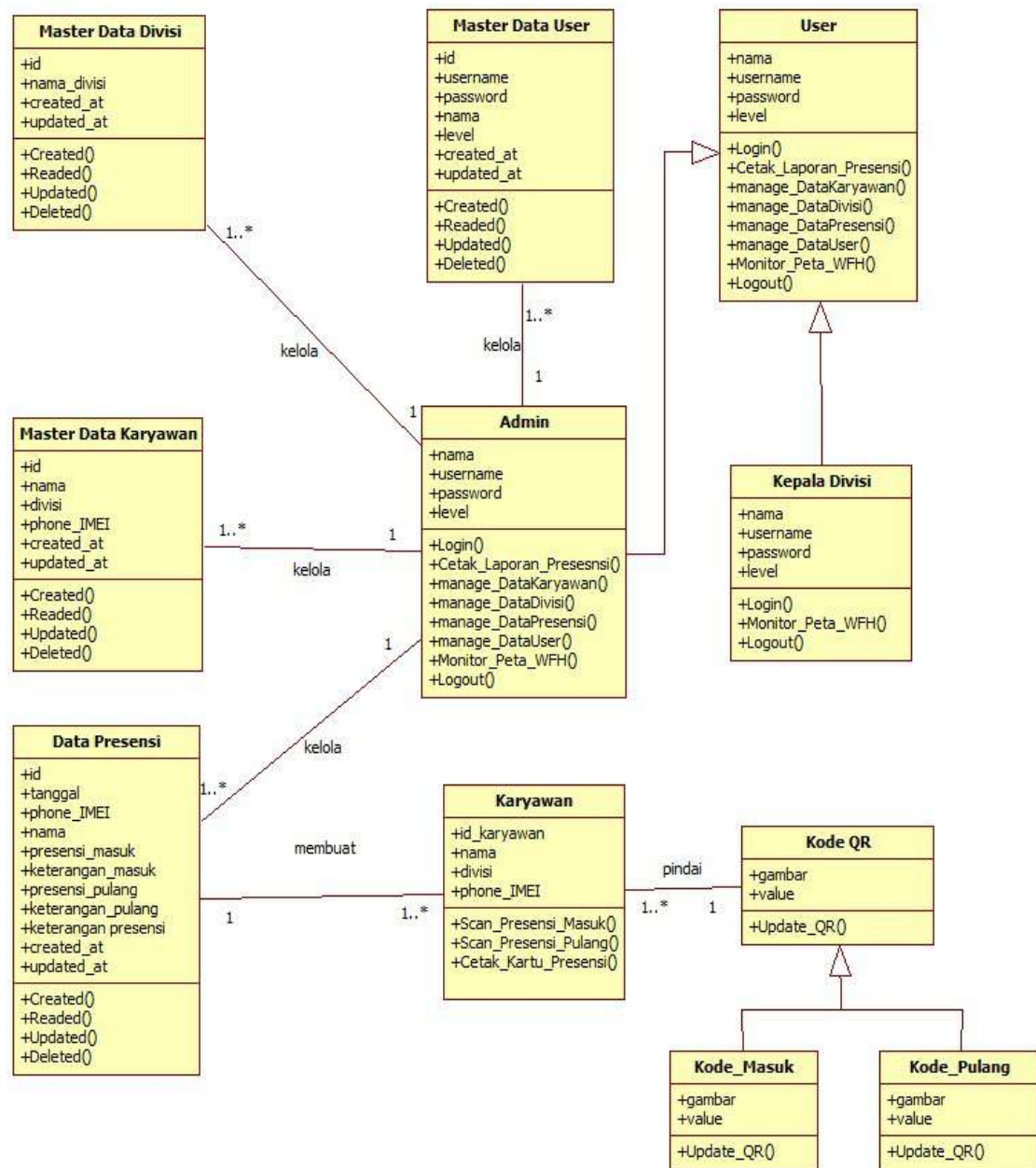


Gambar 3.41 Sequence Diagram Logout



### 3.5.5 Class Diagram

*Class diagram* menggambarkan atribut, operation dan juga *constraint* yang terjadi pada sistem.



Gambar 3.42 Class Diagram

### 3.6 Struktur Basis Data

1. Nama Tabel : tb\_karyawan  
Primary Key : ID\_KARYAWAN

Tabel 3.22 Struktur tb\_karyawan

No	Field	Type	Primary Key
1	ID_KARYAWAN	VARCHAR	*
2	NAMA	VARCHAR	
3	DIVISI	VARCHAR	
4	PHONE_IMEI	VARCHAR	
5	CREATED_AT	DATE	
6	UPDATED_AT	DATE	

2. Nama Tabel : tb\_divisi  
Primary Key : ID\_DIVISI

Tabel 3.23 Struktur tb\_divisi

No	Field	Type	Primary Key
1	ID_DIVISI	VARCHAR	*
2	NAMA_DIVISI	VARCHAR	
3	CREATED_AT	DATE	
4	UPDATED_AT	DATE	

3. Nama Tabel : tb\_presensi  
Primary Key : ID\_PRESENSI

Tabel 3.24 Struktur tb\_presensi

No	Field	Type	Primary Key
1	ID_PRESENSI	VARCHAR	*
2	KARYAWAN	VARCHAR	
3	PHONE_IMEI	VARCHAR	
4	TANGGAL	DATE	
5	MASUK	TIME	
6	KET_MASUK	VARCHAR	
7	PULANG	TIME	
8	KET_PULANG	VARCHAR	
9	KET_PRESENSI	VARCHAR	
10	LONGITUDE	VARCHAR	
11	LATITUDE	VARCHAR	
12	UPDATED_AT	DATE	

4. Nama Tabel : tb\_user  
 Primary Key : ID\_USER

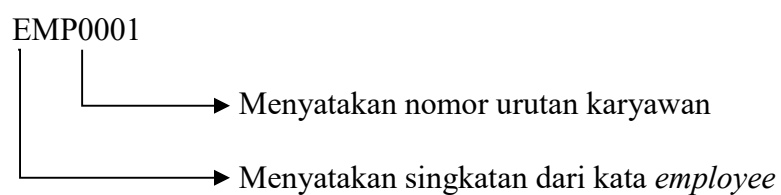
Tabel 3.25 Struktur tb\_user

No	Field	Type	Primary Key
1	ID_USER	VARCHAR	*
2	NAMA	VARCHAR	
3	USERNAME	VARCHAR	
4	PASSWORD	VARCHAR	
5	LEVEL	VARCHAR	
6	CREATED_AT	DATE	
7	UPDATED_AT	DATE	

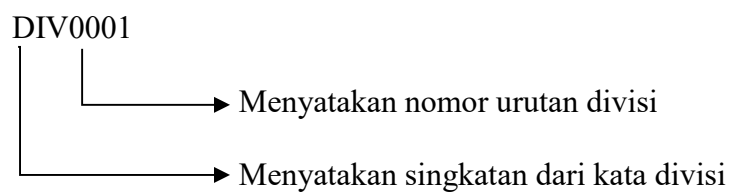
### 3.7 Pengkodean

Pengkodean digunakan sebagai alat bantu pengolahan data yang dapat mempermudah dalam pemasukan data dan pencarian data. Pembuaan kode juga dimaksudkan untuk memudahkan admin dalam mengingat arti dari sebuah struktur kode yang telah dibuat. Adapun pengkodean dalam sistem presensi *quick response code* berbasis web dan android *client* ini adalah sebagai berikut :

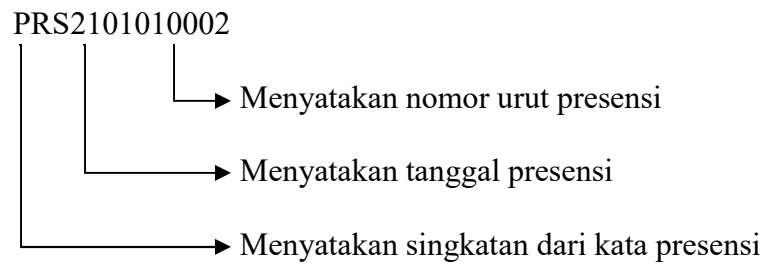
#### 1. ID\_KARYAWAN



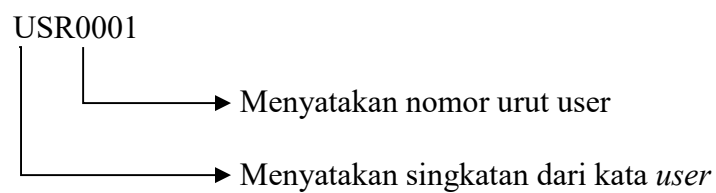
#### 2. ID\_DIVISI



### 3. ID\_PRESENSI



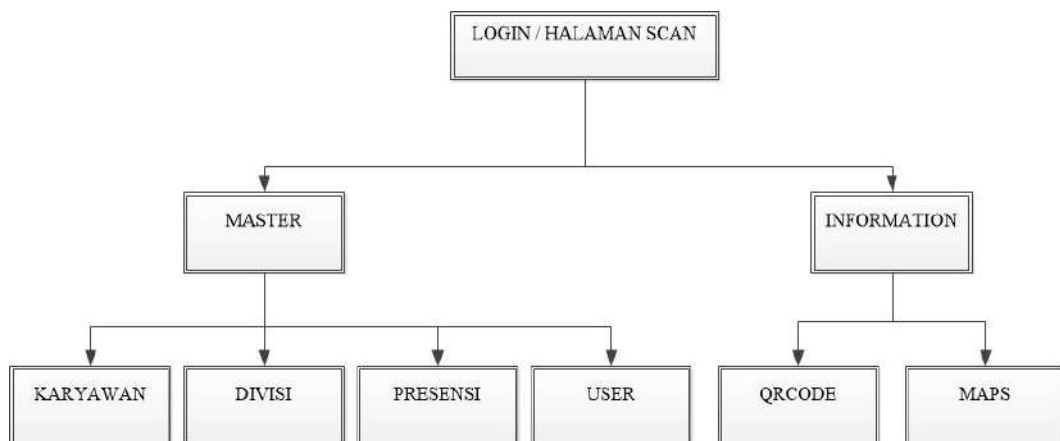
### 4. ID\_USER



## 3.8 Struktur Menu

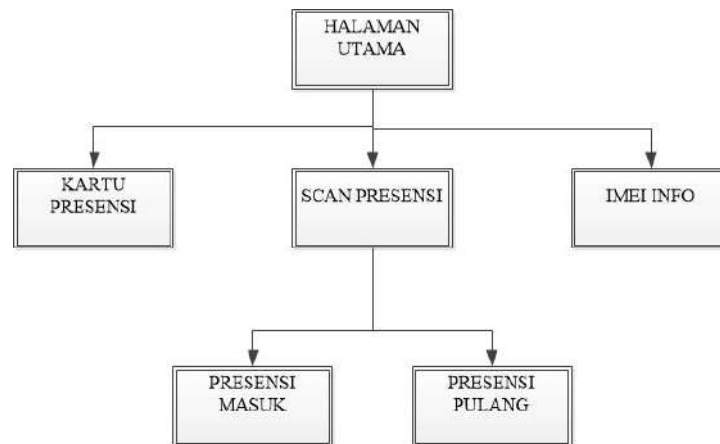
Adapun struktur menu dari sistem baru yang akan dibangun akan terbagi dalam dua antar muka (*interface*) yang terdiri dari beberapa menu yang satu sama lain saling berkaitan.

### 3.8.1 Struktur Menu Antarmuka *Web*



Gambar 3.43 Struktur menu antarmuka *web*

### 3.8.2 Struktur Menu Antarmuka Ponsel *Android*

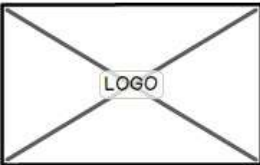


Gambar 3.44 Struktur menu antarmuka ponsel *android*

## 3.9 Perancangan Antarmuka

Perancangan antarmuka merupakan rancangan-rancangan tampilan antarmuka sistem yang akan dipakai oleh pengguna (*user*). Adapun perncangan antarmuka dari sistem presensi *quick response code* ini adalah sebagai berikut :

### 1. Perancangan Tampilan Halaman *Login*



LOGO

SCAN TO RECORD YOUR PRESENCE

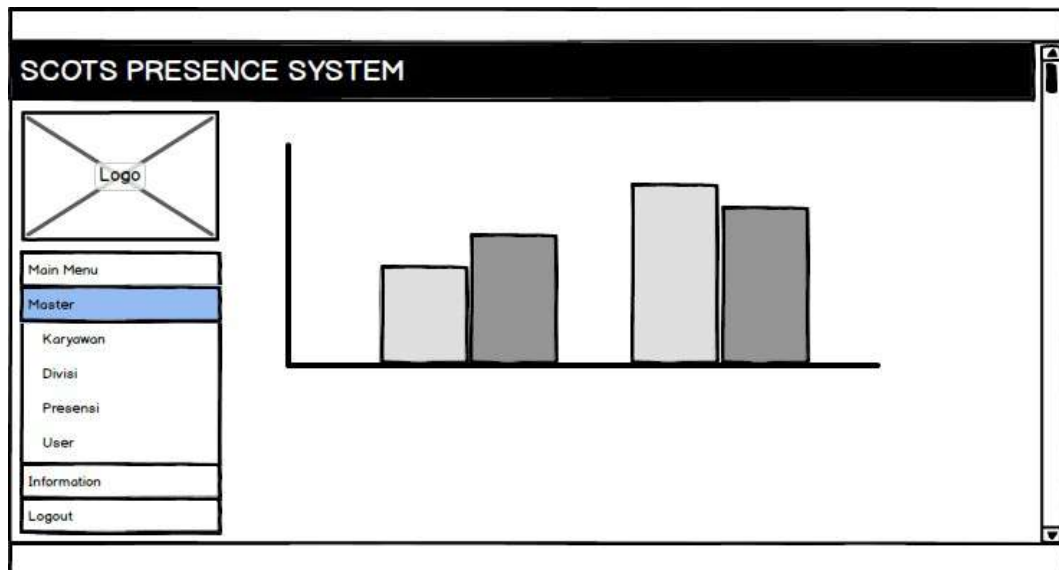
**SISTEM PRESENSI QUICK RESPONSE CODE**  
**CV. INDOSCOTS BABY UTAMA**

USERNAME

PASSWORD

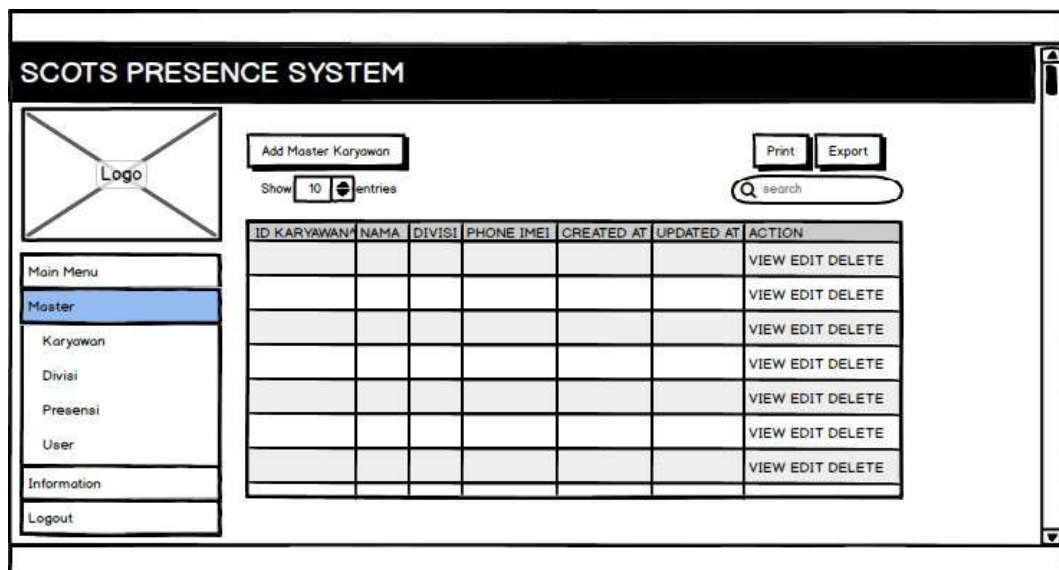
Gambar 3.45 Perancangan tampilan halaman *login*

## 2. Perancangan Tampilan Halaman Utama



Gambar 3.46 Perancangan tampilan halaman utama

## 3. Perancangan Tampilan Menu Master Karyawan



Gambar 3.47 Perancangan tampilan menu master karyawan

#### 4. Perancangan Tampilan *Form* Tambah Data Karyawan

The screenshot displays the 'SCOTS PRESENCE SYSTEM' interface. On the left is a 'Main Menu' sidebar with options: Logo, Master (highlighted), Karyawan, Divisi, Presensi, User, Information, and Logout. The main content area is titled 'Record Master Karyawan' and contains a form with the following fields: 'ID KARYAWAN', 'NAMA KARYAWAN', 'DIVISI' (a dropdown menu), 'PHONE IMEI', and 'CREATED AT'. At the bottom of the form are three buttons: 'SAVE', 'SAVE & GO BACK', and 'CANCEL'.

Gambar 3.48 Perancangan tampilan *form* tambah data karyawan

#### 5. Perancangan Tampilan *Form* Update Data Karyawan

The screenshot displays the 'SCOTS PRESENCE SYSTEM' interface, similar to the previous one. The 'Main Menu' sidebar is identical. The main content area is titled 'Record Master Karyawan' and contains a form for updating employee data. The fields are: 'ID KARYAWAN', 'NAMA KARYAWAN', 'DIVISI' (a dropdown menu), 'PHONE IMEI', and 'UPDATE AT'. At the bottom of the form are three buttons: 'UPDATE', 'UPDATE & GO BACK', and 'CANCEL'.

Gambar 3.49 Perancangan tampilan *form* update karyawan

## 6. Perancangan Tampilan Menu Master Divisi

**SCOTS PRESENCE SYSTEM**

Logo

Add Master Divisi

Show 10 entries

Print Export

Q search

ID DIVISI*	NAMA DIVISI	CREATED AT	UPDATED AT	ACTION
				VIEW EDIT DELETE
				VIEW EDIT DELETE
				VIEW EDIT DELETE
				VIEW EDIT DELETE
				VIEW EDIT DELETE
				VIEW EDIT DELETE
				VIEW EDIT DELETE

Main Menu

Master

Karyawan

Divisi

Presensi

User

Information

Logout

Gambar 3.50 Perancangan tampilan menu master divisi

## 7. Perancangan Tampilan *Form* Tambah Divisi

**SCOTS PRESENCE SYSTEM**

Logo

Record Master DIVISI

ID DIVISI

NAMA DIVISI

CREATED AT

SAVE SAVE & GO BACK CANCEL

Main Menu

Master

Karyawan

Divisi

Presensi

User

Information

Logout

Gambar 3.51 Perancangan tampilan *form* tambah divisi



## 8. Perancangan Tampilan *Form Update Divisi*

**SCOTS PRESENCE SYSTEM**

Record Master DIVISI

ID DIVISI

NAMA DIVISI

CREATED AT

Main Menu

Master

Karyawan

Divisi

Presensi

User

Information

Logout

Gambar 3.52 Perancangan tampilan *form update divisi*

## 9. Perancangan Tampilan Menu Master Presensi

**SCOTS PRESENCE SYSTEM**

Add Master Presensi

Print Export

Show 10 entries

Q search

ID PRESEN	KARYAWAN	PHONE IM	TANGGA	MASU	KET MASU	PULAN	KET PULAN	KET PRESEN	LONGITUD	LATITUD	ACTIONS
											VIEW EDIT DELE
											VIEW EDIT DELE
											VIEW EDIT DELE
											VIEW EDIT DELE
											VIEW EDIT DELE
											VIEW EDIT DELE
											VIEW EDIT DELE
											VIEW EDIT DELE
											VIEW EDIT DELE

Main Menu

Master

Karyawan

Divisi

Presensi

User

Information

Logout

Gambar 3.53 Perancangan tampilan menu master presensi

## 10. Perancangan Tampilan *Form* Tambah Data Presensi

The screenshot shows the 'Record Master PRESENSI' form in the SCOTS PRESENCE SYSTEM. The form is titled 'Record Master PRESENSI' and contains the following fields:

- ID PRESENSI: Text input field
- KARYAWAN: Dropdown menu with 'KARYAWAN' selected
- PHONE IMEI: Dropdown menu with 'PHONE IMEI' selected
- TANGGAL: Text input field
- MASUK: Text input field
- KET MASUK: Text input field
- PULANG: Text input field
- KET PULANG: Text input field
- KET PRESEN: Text input field
- LONGITUDE: Text input field
- LATITUDE: Text input field

At the bottom of the form are three buttons: 'SAVE', 'SAVE & GO BACK', and 'CANCEL'. On the left side, there is a sidebar menu with the following items: 'Main Menu', 'Master' (highlighted), 'Karyawan', 'Divisi', 'Presensi', 'User', 'Information', and 'Logout'.

Gambar 3.54 Perancangan tampilan *form* tambah presensi

## 11. Perancangan Tampilan *Form Update* Presensi

The screenshot shows the 'Record Master PRESENSI' form in the SCOTS PRESENCE SYSTEM, designed for updating existing attendance data. The form is titled 'Record Master PRESENSI' and contains the following fields:

- ID PRESENSI: Text input field
- KARYAWAN: Dropdown menu with 'KARYAWAN' selected
- PHONE IMEI: Dropdown menu with 'PHONE IMEI' selected
- TANGGAL: Text input field
- MASUK: Text input field
- KET MASUK: Text input field
- PULANG: Text input field
- KET PULANG: Text input field
- KET PRESEN: Text input field
- LONGITUDE: Text input field
- LATITUDE: Text input field

At the bottom of the form are three buttons: 'UPDATE', 'UPDATE & GO BACK', and 'CANCEL'. On the left side, there is a sidebar menu with the following items: 'Main Menu', 'Master' (highlighted), 'Karyawan', 'Divisi', 'Presensi', 'User', 'Information', and 'Logout'.

Gambar 3.55 Perancangan Tampilan *Form Update* Presensi

## 12. Perancangan Tampilan Menu Master User

The screenshot shows the 'SCOTS PRESENCE SYSTEM' interface. On the left is a 'Main Menu' sidebar with options: Main Menu, Master (highlighted), Karyawan, Divisi, Presensi, User, Information, and Logout. The main content area features a 'Logo' placeholder, an 'Add Master User' button, a 'Show 10 entries' dropdown, and 'Print' and 'Export' buttons. Below these is a search bar labeled 'Q search'. The central part of the interface is a table with the following columns: ID User, NAMA, USERNAME, PASSWORD, LEVEL, CREATED AT, UPDATED AT, and ACTION. The table contains 8 rows of data, each with a 'VIEW EDIT DELETE' link in the ACTION column.

ID User	NAMA	USERNAME	PASSWORD	LEVEL	CREATED AT	UPDATED AT	ACTION
							VIEW EDIT DELETE
							VIEW EDIT DELETE
							VIEW EDIT DELETE
							VIEW EDIT DELETE
							VIEW EDIT DELETE
							VIEW EDIT DELETE
							VIEW EDIT DELETE
							VIEW EDIT DELETE

Gambar 3.56 Perancangan tampilan menu master user

## 13. Perancangan Tampilan Form Tambah User

The screenshot shows the 'SCOTS PRESENCE SYSTEM' interface for adding a new user. The 'Main Menu' sidebar is the same as in the previous image, with 'Master' highlighted. The main content area has a 'Record Master USER' title. Below the title are input fields for: ID USER, NAMA, USERNAME, PASSWORD, LEVEL, and CREATED AT. At the bottom of the form are three buttons: 'SAVE', 'SAVE & GO BACK', and 'CANCEL'.

Gambar 3.57 Perancangan tampilan form tambah user

#### 14. Perancangan Tampilan Form *Update User*

The screenshot shows the 'SCOTS PRESENCE SYSTEM' interface. On the left is a sidebar menu with options: Main Menu, Master (highlighted), Karyawan, Divisi, Presensi, User, Information, and Logout. The main content area is titled 'Record Master USER' and contains the following fields: ID USER, NAMA, USERNAME, PASSWORD, LEVEL, and UPDATE AT. At the bottom of the form are three buttons: UPDATE, UPDATE & GO BACK, and CANCEL.

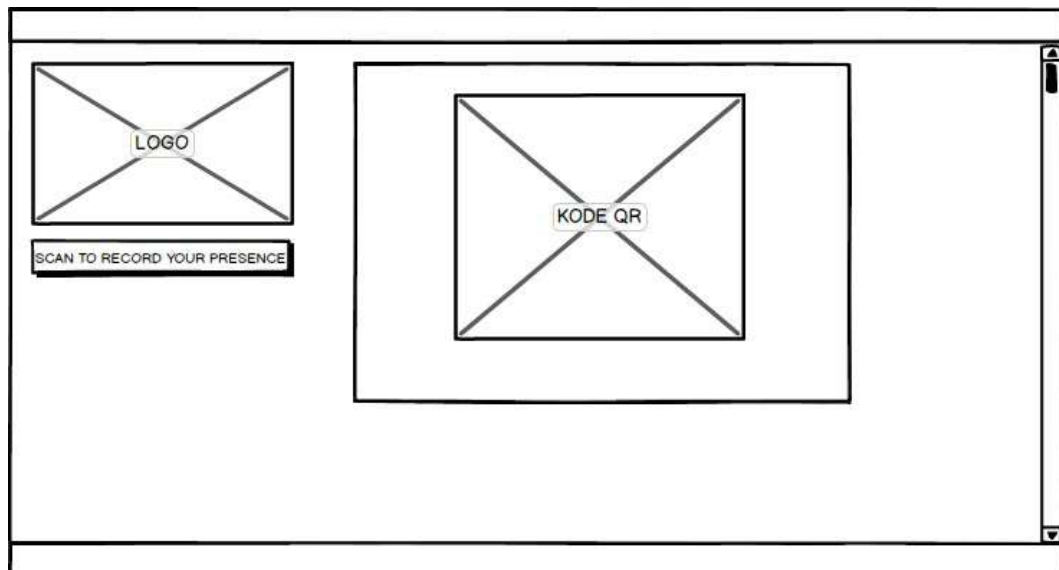
Gambar 3.58 Perancangan tampilan form *update user*

#### 15. Perancangan Tampilan Peta Sebaran *Work From Home*

The screenshot shows the 'SCOTS PRESENCE SYSTEM' interface. On the left is a sidebar menu with options: Main Menu, Master, Information (highlighted), QR Code, Maps, and Logout. The main content area is titled 'Peta Sebaran Work From Home' and contains a large placeholder box for a map.

Gambar 3.59 Perancangan peta sebaran *work from home*

## 16. Perancangan Tampilan *Scan* Presensi



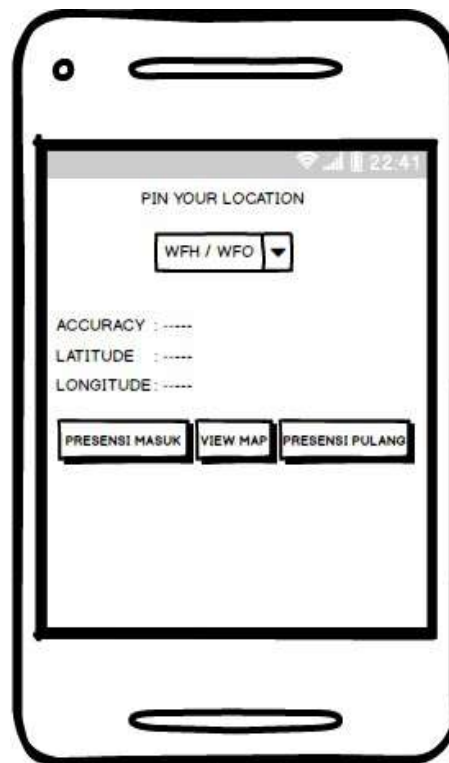
Gambar 3.60 Perancangan tampilan *scan* presensi

## 17. Perancangan Tampilan Menu Utama Aplikasi Presensi



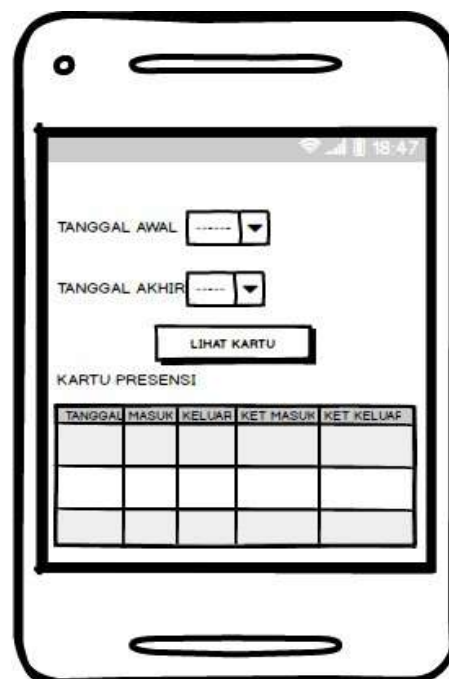
Gambar 3.61 Perancangan tampilan menu utama aplikasi presensi

### 18. Perancangan Tampilan Menu *Scan* Presensi



Gambar 3.62 Perancangan tampilan menu *scan* presensi

### 19. Perancangan Tampilan Menu Kartu Presensi



Gambar 3.63 Perancangan menu kartu presensi

## BAB IV

### IMPLEMENTASI SISTEM

#### 4.1 Implementasi *Hardware*



Perangkat keras yang digunakan untuk mendukung kegiatan presensi pada sistem presensi *quick response code* memiliki spesifikasi sebagai berikut :

1. Sebuah *web server* minimal bersistem operasi *linux ubuntu* atau *debian server*
2. Satu unit komputer atau satu unit laptop minimal memiliki processor *intel dual core*.
3. Satu unit printer
4. Ponsel android minimal versi 4.4 (*Kitkat*)

#### 4.2 Tampilan Struktur Tabel *Database*

Tampilan struktur *table database* merupakan bagian dari implementasi tabel-tabel pada *database* yang akan diakses oleh pengguna (*user*)

##### 4.2.1 Tampilan Struktur Tabel *tb\_karyawan*

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan
1	ID_KARYAWAN 	varchar(8)	latin1_swedish_ci		Tidak	Tidak ada
2	NAMA	varchar(30)	latin1_swedish_ci		Tidak	Tidak ada
3	DIVISI 	varchar(8)	latin1_swedish_ci		Tidak	Tidak ada
4	PHONE_IMEI	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada
5	CREATED_AT	date			Tidak	Tidak ada
6	UPDATED_AT	date			Tidak	Tidak ada


Gambar 4.1 Struktur tabel *tb\_karyawan*

#### 4.2.2 Tampilan Struktur Tabel tb\_divisi

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan
1	ID_DIVISI 	varchar(8)	latin1_swedish_ci		Tidak	Tidak ada
2	NAMA_DIVISI	varchar(20)	latin1_swedish_ci		Tidak	Tidak ada
3	CREATED_AT	date			Tidak	Tidak ada
4	UPDATED_AT	date			Tidak	Tidak ada

Gambar 4.2 Struktur tabel tb\_divisi

#### 4.2.3 Tampilan Struktur Tabel tb\_presensi

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan
1	ID_PRESENSI 	varchar(20)	latin1_swedish_ci		Tidak	Tidak ada
2	KARYAWAN	varchar(30)	latin1_swedish_ci		Ya	NULL
3	PHONE_IMEI	varchar(30)	latin1_swedish_ci		Tidak	Tidak ada
4	TANGGAL	date			Tidak	Tidak ada
5	MASUK	time			Ya	NULL
6	KET_MASUK	varchar(30)	latin1_swedish_ci		Ya	NULL
7	PULANG	time			Tidak	Tidak ada
8	KET_PULANG	varchar(30)	latin1_swedish_ci		Tidak	Tidak ada
9	KET_PRESENSI	varchar(30)	latin1_swedish_ci		Tidak	Tidak ada
10	LONGITUDE	varchar(255)	latin1_swedish_ci		Tidak	Tidak ada
11	LATITUDE	varchar(255)	latin1_swedish_ci		Tidak	Tidak ada
12	UPDATED_AT	date			Tidak	Tidak ada

Gambar 4.3 Struktur tabel tb\_presensi

#### 4.2.4 Tampilan Struktur Tabel tb\_user

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan
1	ID_USER 	varchar(10)	latin1_swedish_ci		Tidak	Tidak ada
2	NAMA	varchar(30)	latin1_swedish_ci		Tidak	Tidak ada
3	USERNAME	varchar(30)	latin1_swedish_ci		Tidak	Tidak ada
4	PASSWORD	varchar(30)	latin1_swedish_ci		Tidak	Tidak ada
5	LEVEL	varchar(1)	latin1_swedish_ci		Tidak	Tidak ada
6	CREATED_AT	date			Tidak	Tidak ada
7	UPDATED_AT	date			Tidak	Tidak ada

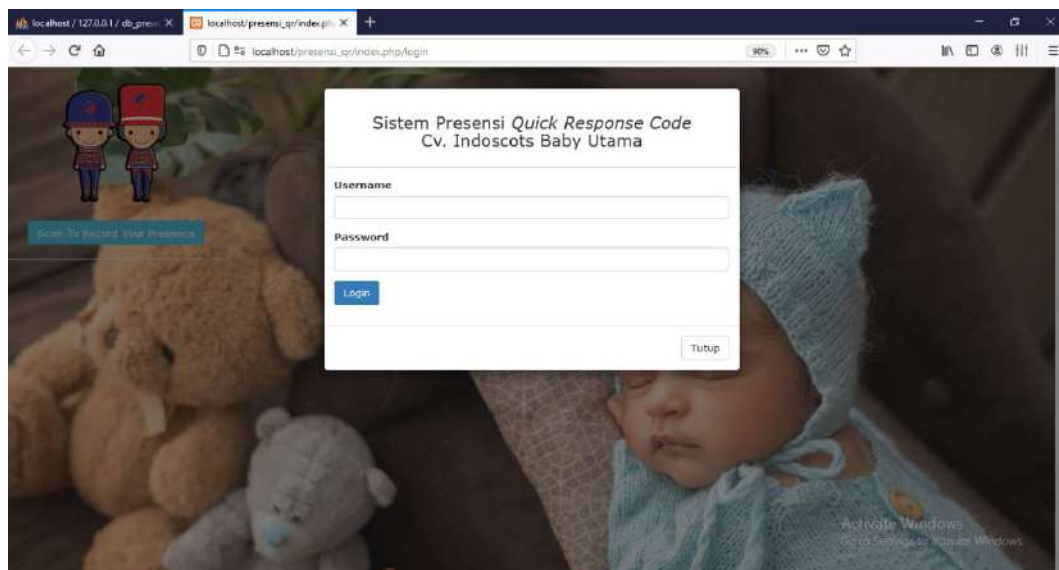
Gambar 4.4 Struktur tabel tb\_user



### 4.3 Tampilan *Program*

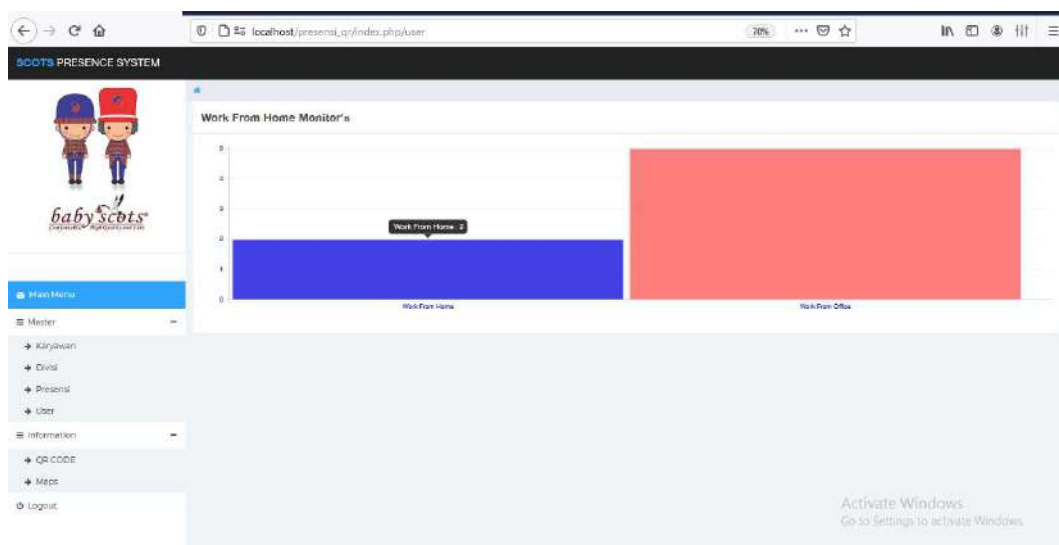
Tampilan program merupakan bagian dari implementasi perancangan antarmuka (*interface*) yang akan diakses oleh pengguna (*user*). Adapun tampilan dari sistem presensi *quick response code* ini adalah sebagai berikut :

#### 1. Tampilan *Login*



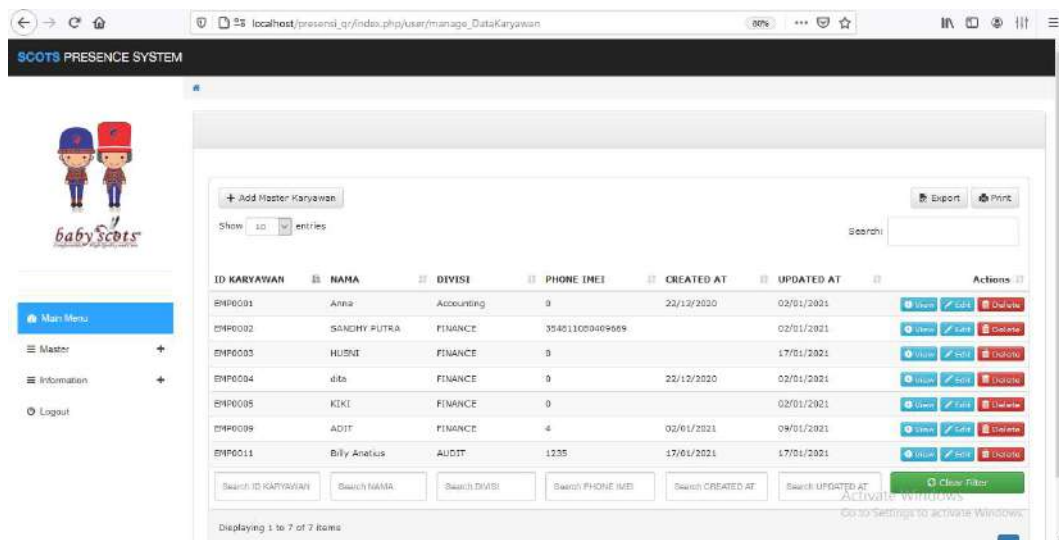
Gambar 4.5 Tampilan *login*

#### 2. Tampilan Halaman Utama



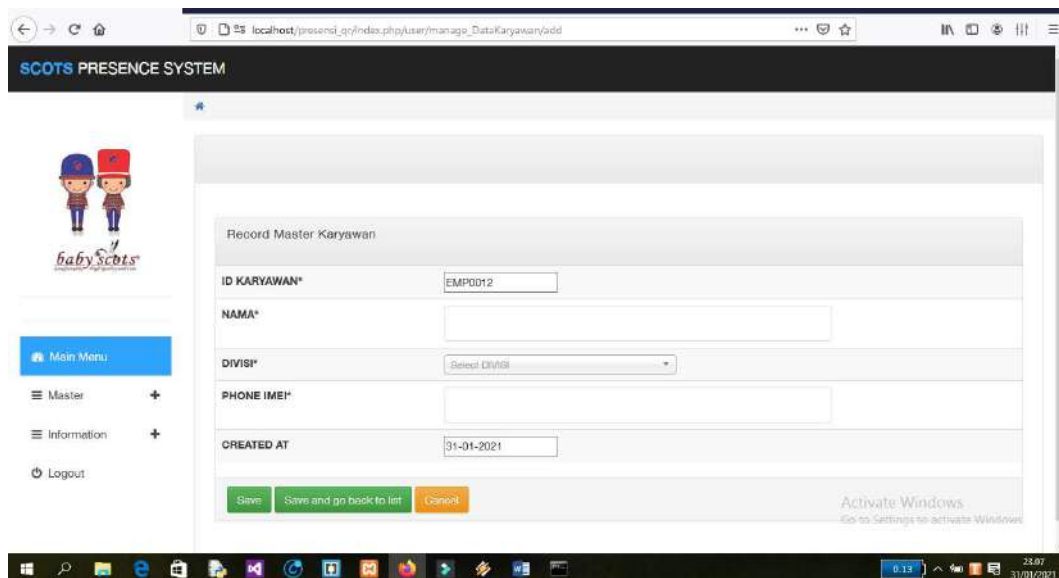
Gambar 4.6 Tampilan halaman utama

### 3. Tampilan Menu Master Karyawan



Gambar 4.7 Tampilan menu master karyawan

### 4. Tampilan Form Tambah Data Karyawan



Gambar 4.8 Tampilan form tambah data karyawan

## 5. Tampilan *Form Update Data Karyawan*

SCOTS PRESENCE SYSTEM

Record Master Karyawan

ID KARYAWAN\* EMP0001

NAMA\* Anna

DIVISI\* Accounting

PHONE IMEI\* 0

UPDATED AT 31-01-2021

Update changes Update and go back to list Cancel

Gambar 4.9 Tampilan *Form update Karyawan*

## 6. Tampilan Menu Master Divisi

SCOTS PRESENCE SYSTEM

+ Add Master Divisi

Show 10 entries

Search:

ID DIVISI	NAMA DIVISI	CREATED AT	UPDATED AT	Actions
DIV0001	FINANCE	02/01/2021	02/01/2021	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
DIV0002	IT	02/01/2021	02/01/2021	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
DIV0003	Accounting	02/01/2021	02/01/2021	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
DIV0004	MARKETING	02/01/2021	02/01/2021	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
DIV0005	AUDIT	02/01/2021	02/01/2021	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
DIV0006	PRODUKSI	02/01/2021	02/01/2021	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
DIV0007	PROCUREMENT	02/01/2021	02/01/2021	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

Search ID DIVISI Search NAMA DIVISI Search CREATED AT Search UPDATED AT

Clear Filter

Displaying 1 to 7 of 7 rows

Gambar 4.10 Tampilan menu master divisi

## 7. Tampilan *Form* Tambah Divisi

The screenshot shows a web browser window with the URL `localhost/presensi_qr/index.php/user/manage_DataDivisi/add`. The page title is "SCOTS PRESENCE SYSTEM". On the left, there is a sidebar menu with the "babyScots" logo and options: "Main Menu", "Master", "Information", and "Logout". The main content area displays a form titled "Record Master Divisi" with the following fields:

- ID DIVISI\***: A text input field containing the value "DIV0008".
- NAMA DIVISI**: An empty text input field.
- CREATED AT\***: A date input field showing "31-01-2021".

At the bottom of the form, there are three buttons: "Save" (green), "Save and go back to list" (green), and "Cancel" (orange). Below the form, there is a watermark that says "Activate Windows Go to Settings to activate Windows".

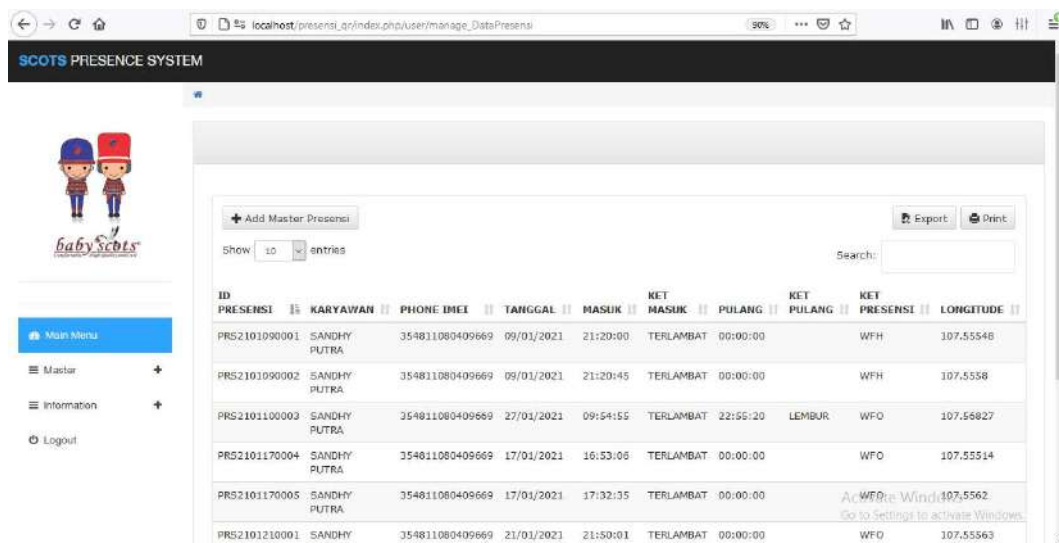
Gambar 4.11 Tampilan *form* tambah divisi

## 8. Tampilan *Form* Update Divisi

This screenshot is identical to the one above, showing the "Record Master Divisi" form in the SCOTS PRESENCE SYSTEM. The form fields and buttons are the same, but the context is for updating an existing division. The sidebar menu and browser window details are also consistent with the previous image.

Gambar 3.12 Tampilan *form* update divisi

## 9. Tampilan Menu Master Presensi



SCOTS PRESENCE SYSTEM

babyScots

Main Menu

- Master
- Information
- Logout

Add Master Presensi

Show 10 entries

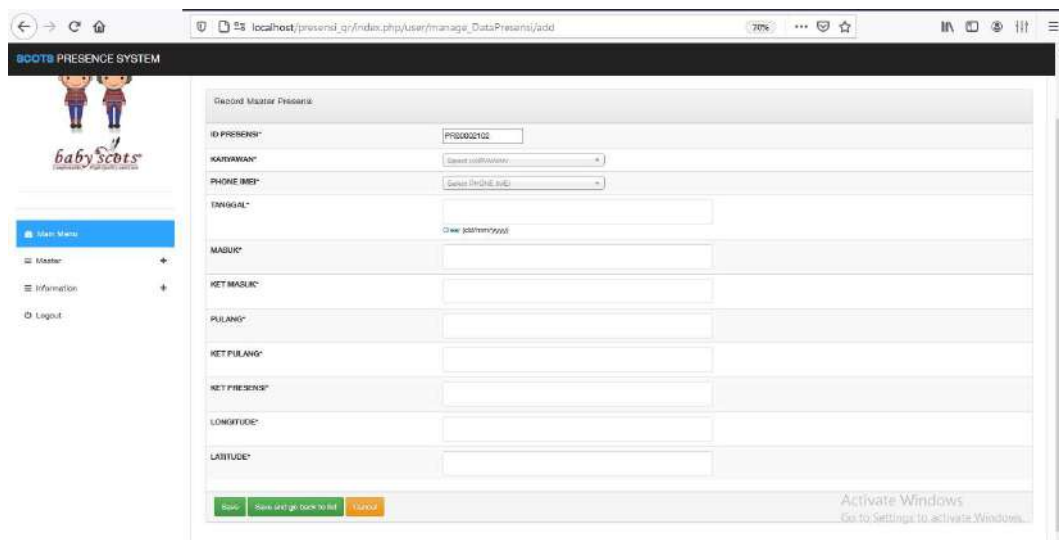
Search:

ID	PRESENSI	KARYAWAN	PHONE IMEI	TANGGAL	MASUK	KET MASUK	PULANG	KET PULANG	KET PRESENSI	LONGITUDE
PRS2101090001		SANDHY PUTRA	354811080409669	09/01/2021	21:20:00	TERLAMBAT	00:00:00		WFH	107.55548
PRS2101090002		SANDHY PUTRA	354811080409669	09/01/2021	21:20:45	TERLAMBAT	00:00:00		WFH	107.55558
PRS2101100003		SANDHY PUTRA	354811080409669	27/01/2021	09:54:55	TERLAMBAT	22:55:20	LEMBUR	WFO	107.56827
PRS2101170004		SANDHY PUTRA	354811080409669	17/01/2021	16:53:08	TERLAMBAT	00:00:00		WFO	107.55514
PRS2101170005		SANDHY PUTRA	354811080409669	17/01/2021	17:32:35	TERLAMBAT	00:00:00		WFO	107.55562
PRS2101210001		SANDHY PUTRA	354811080409669	21/01/2021	21:50:01	TERLAMBAT	00:00:00		WFO	107.55563

Export Print

Gambar 4.13 Tampilan menu master presensi

## 10. Tampilan Form Tambah Presensi



SCOTS PRESENCE SYSTEM

babyScots

Main Menu

- Master
- Information
- Logout

Record Master Presensi

ID PRESENSI\*

KARYAWAN\*

PHONE IMEI\*

TANGGAL\*

MASUK\*

KET MASUK\*

PULANG\*

KET PULANG\*

KET PRESENSI\*

LONGITUDE\*

LATITUDE\*

Save Save and go back to list Cancel

Activate Windows  
Go to Settings to activate Windows.

Gambar 4.14 Tampilan form tambah presensi

## 11. Tampilan *Form Update Presensi*

SCOTS PRESENCE SYSTEM

Record Master Presensi

ID PRESENSI\* PR52101170005

KARTAWAN\* SANDHY PUTRA

PHONE BRIEF\* 0841-00041000

UNIKDAL\* 17012021

MADUK\* 01-05-05

KET MADUK\* TOPLAWASAT

PULANG\* 170000

KET PULANG\* LEMBUR

KET PRESENSI\* WFO

LONGITUDE\* 107.8862

LATITUDE\* -7.01717

UPDATED AT\* 01-01-2021

Activate Windows  
Go to Settings to activate Windows.

Gambar 4.15 Tampilan *form update presensi*

## 12. Tampilan Menu MasterUser

SCOTS PRESENCE SYSTEM

+ Add Master User

Show 10 entries

Search:

ID USER	NAMA	USERNAME	PASSWORD	LEVEL	CREATED AT	UPDATED AT	Actions
US00001	Sandhy	adm_sandhy	sandhy123	1	02/01/2021	09/01/2021	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
US00002	Putra	adm_putra	putra123	2	02/01/2021	09/01/2021	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
US00009	SAPUTRA	adm_Saputra	1234	2	02/01/2021	09/01/2021	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

Search ID USER Search NAMA Search USERNAME Search PASSWORD Search LEVEL Search CREATE Search UPDATE

Clear Filter

Displaying 1 to 3 of 3 items

Activate Windows  
Go to Settings to activate Windows.

Gambar 4.16 Tampilan menu master *user*

### 13. Tampilan *Form Tambah User*

The screenshot shows the 'Record Master User' form in the SCOTS PRESENCE SYSTEM. The form is titled 'Record Master User' and contains the following fields:

- ID USER\*: USR0010
- NAMA\*: (empty)
- USERNAME\*: (empty)
- PASSWORD\*: (empty)
- LEVEL\*: (empty)
- CREATED AT\*: 31-01-2021

At the bottom of the form, there are three buttons: 'Save' (green), 'Save and go back to list' (green), and 'Cancel' (orange). The left sidebar shows the 'Main Menu' with options: Master, Information, and Logout. The browser address bar shows 'localhost/presensi\_qr/index.php/user/manage\_DataUser/add'.

Gambar 4.17 Tampilan *form tambah user*

### 14. Tampilan *Form Update User*

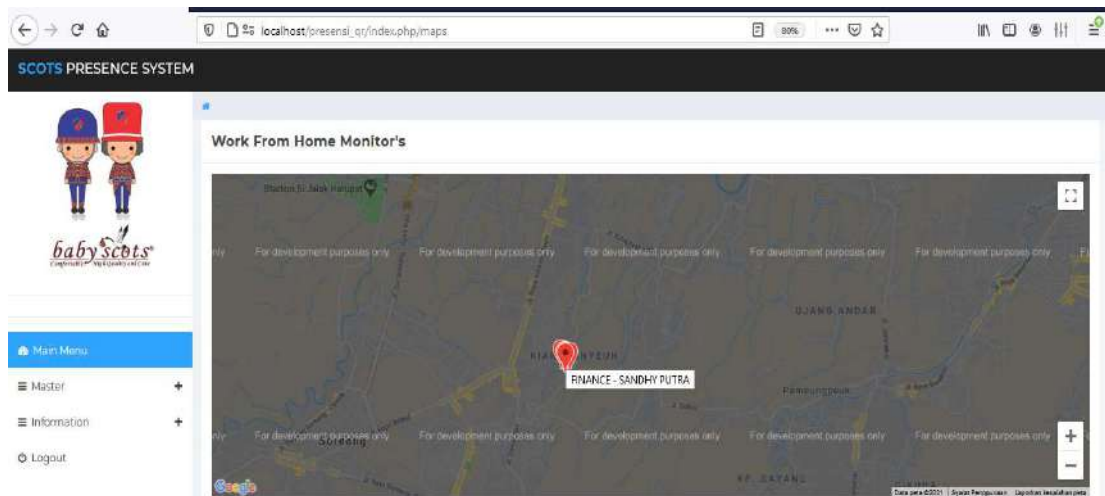
The screenshot shows the 'Record Master User' form in the SCOTS PRESENCE SYSTEM, specifically for updating an existing user. The form is titled 'Record Master User' and contains the following fields:

- ID USER\*: USR00001
- NAMA\*: Sandhy
- USERNAME\*: adm\_sandhy
- PASSWORD\*: sandhy123
- LEVEL\*: 1
- UPDATED AT\*: 31-01-2021

At the bottom of the form, there are three buttons: 'Update changes' (green), 'Update and go back to list' (green), and 'Cancel' (orange). The left sidebar shows the 'Main Menu' with options: Master, Information, and Logout. The browser address bar shows 'localhost/presensi\_qr/index.php/user/manage\_DataUser/edit/USR00001'.

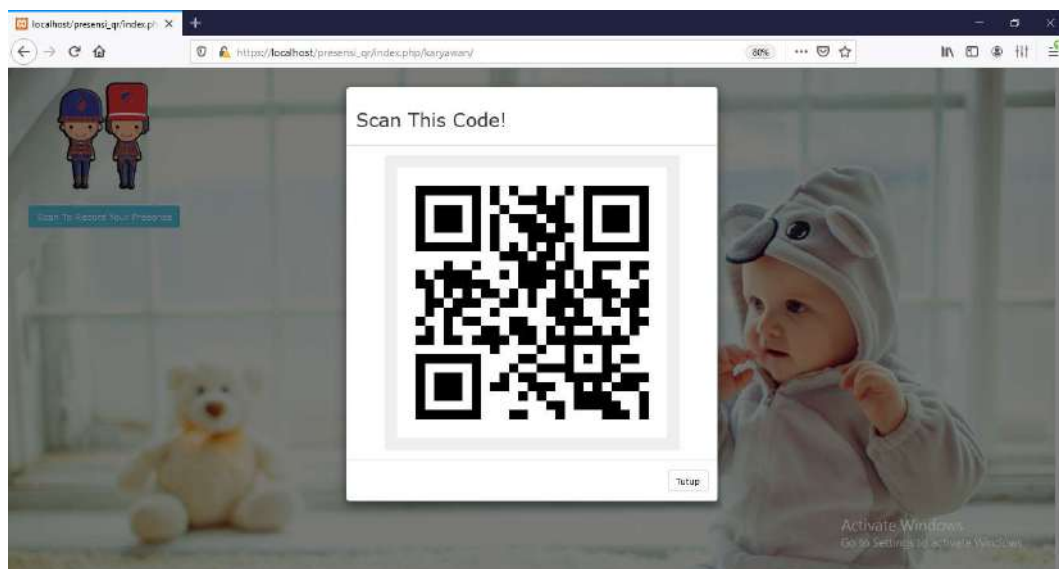
Gambar 4.18 Tampilan *form update user*

## 15. Tampilan Peta Sebaran *Work From Home*



Gambar 4.19 Tampilan peta sebaran *work from home*

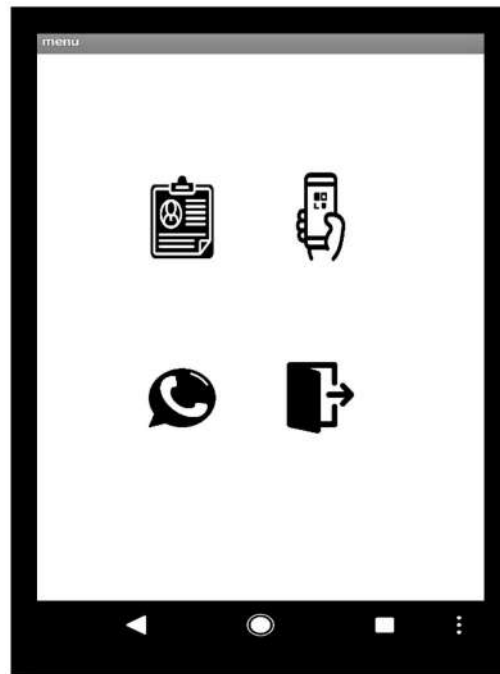
## 16. Tampilan Halaman *Scan* Presensi



Gambar 4.20 Tampilan halaman *scan* presensi



### 17. Tampilan Menu Utama Aplikasi Presensi



Gambar 4.21 Tampilan menu utama aplikasi presensi

### 18. Tampilan Menu Scan Presensi



Gambar 4.22 Tampilan menu *scan* presensi

## 19. Tampilan Menu Kartu Presensi

Kartu

Tanggal Awal 2021-1-1

Tanggal Akhir 2021-1-29

Lihat Kartu

**KARTU PRESENSI**  
**EMP0002 - SANDHY PUTRA**

Tanggal	Masuk	Pulang	KET. Masuk	KET. Pulang
2021-01-27	07:54:55	17:12:20	OK	LEMBUR
2021-01-09	08:20:00	16:30:00	OK	OK
2021-01-17	08:45:35	17:00:00	TERLAMBAT	LEMBUR
2021-01-21	08:50:01	18:00:00	TERLAMBAT	LEMBUR
2021-01-28	08:12:06	17:40:00	OK	LEMBUR

Gambar 4.23 Tampilan menu kartu presensi

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil analisis selama mengerjakan Kerja Praktek di CV. Indoscots Baby Utama yang berjudul Perancangan Sistem Presensi *Quick Response Code* berbasis *web* dan *android client*, maka penulis mendapat kesimpulan yaitu:

1. Sistem dapat menghindari kecurangan karyawan dalam pencatatan jam kehadiran ketika jam masuk dan jam pulang kerja.
2. Adanya fitur peta sebaran *work from home* membuat pihak perusahaan mendapat informasi terhadap posisi karyawan yang sedang bekerja di rumah.
3. Sistem presensi sudah sepenuhnya terkomputerisasi.

#### **5.2 Saran**

Untuk meningkatkan kinerja Sistem Presensi *Quick Response Code* , maka penulis mengajukan beberapa saran kepada pihak yang mengelola, antara lain :

1. Sistem yang telah dibuat ini dapat dijadikan sebagai pemecahan masalah dan sistem ini harus dikembangkan lebih baik lagi.
2. Diperlukan kebijakan perusahaan terhadap penggunaan sistem ini untuk pengamanan terhadap data yang bersifat penting dan rahasia seperti data nomor IMEI ponsel dan data koordinat karyawan yang melakukan kegiatan presensi.

## DAFTAR PUSTAKA

- Abdulloh, R. (2018). *7 in 1 Pemrograman Web untuk Pemula*. Jakarta : Elex Media Komputindo.
- Herlinah, & Musliadi, K. (2019). *Pemrograman Aplikasi Android dengan Android Studio, Photoshop, dan Audition*. Jakarta : Elex Media Komputindo.
- Huda, M. M. K. (2019). *Teknologi Komputer: Pemanfaatan teknologi komputer untuk mempermudah penyelesaian tugas dan pekerjaan yang dihadapi*. Bogor : bisakimia.
- Kadir, A. (2017). *Pemrograman Arduino & Android Menggunakan App Inventor*. Jakarta : Elex Media Komputindo.
- Google Developer Team. (2016). *Learn to develop Android Applications*. Google Developer Team
- Kadir, A. (2018). *Langkah Mudah Pemrograman Android Menggunakan App Inventor 2 Ultimate*. Jakarta : Elex Media Komputindo.
- Lubis, A. (2016). *Basis Data Dasar*. Yogyakarta : Deepublish.
- Muslihudin, M., & Oktafianto. (2016). *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*. Surakarta : Penerbit Andi.
- Rachmadi, T. (2020). *Sistem Basis Data*. Jakarta : TIGA Publisher.
- Takdirillah, R. (2020, July 5). *Inilah Urutan Versi Android dari Awal Hingga Terbaru (Lengkap)*. Dicoding.Com. <https://www.dicoding.com/blog/urutan-versi-android/> (diakses pada 19 Desember 2020 Jam 21.42 WIB)
- Widodo, I. W. (2015). *Membangun Web Super Cepat dengan CodeIgniter GroceryCRUD dan TankAuth*. In Wahyu Widodo.
- Yanto, R. (2016). *Manajemen Basis Data Menggunakan MySQL*. Yogyakarta : Deepublish.
- Yudhanto, Y., & Azis, A. (2019). *Pengantar Teknologi Internet of Things (IoT)*. Yogyakarta : UNSPress.
- Sulistiono, H. (2018). *Coding Mudah dengan CodeIgniter, JQuery, Bootstrap, dan Datatable*. Jakarta : Elex Media Komputindo.
- Suyuti, M. A., & Nur, R. (2018). *Perancangan Mesin-Mesin Industri*. Yogyakarta : Deepublish.
- Rosa, A., & Salahudin.(2018). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung : Informatika.
- Munawar.(2018). *ANALISIS PERANCANGAN SISTEM BERORIENTASI OBJEK DENGAN UNIFIED MODELLING LANGUAGE*. Bandung : Informatika.

- Rahima. (2016). Pengaruh Penerapan Strategi Question Student Have Disertai Modul Berbasis Masalah Terhadap Aktivitas Belajar Mahasiswa Pada Perkuliahan Kalkulus Peubah Banyak 2. *Lemma*, 2(2), 145055.
- Hidayat, R., & Wagyana, A. (2019). Rancang Bangun Sistem Presensi Menggunakan Face Recognition dengan Metode Eigenface. *Setrum: Sistem Kendali-Tenaga-Elektronika-Telekomunikasi-Komputer*, 7(2), 278–287.
- Khoiriyah, N. L., Marisa, F., Wijaya, I. D. (2018). *Rancang Bangun Sistem Presensi Online Berbasis Granted Validitas Data*. 3(1), 53–61.
- Purwaningtias, F. (2018). E-Commerce Penjualan Berbasis Metode Ooad. *Jurnal Cendikia*, 16(1 April), 1–5.
- Hasanuddin. (2016). SISTEM INFORMASI KEUANGAN DENGAN METODE OBJECT ORIENTED ANALISYS DESIGN. *Technologia: Jurnal Ilmiah*, 7(2).
- Sugiantoro, B., & Fuad, H. (2015). Pengembangan QR Code Scanner Berbasis Android Untuk Sistem Informasi Museum Sonobudoyo Yogyakarta. *Telematika: Jurnal Informatika Dan Teknologi Informasi*, 12(2), 134–145.