

# Implementasi Absensi Karyawan Menggunakan Algoritma Haversine dengan Global Positioning System Berbasis Android

Fajar Antono<sup>1</sup>, Saruni Dwiasnati<sup>2\*</sup>

<sup>1,2</sup> Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana

<sup>1,2</sup> Jl. Raya, RT.4/RW.1, Meruya Sel., Kec. Kembangan, Jakarta, Daerah Khusus Ibukota Jakarta 11650

<sup>1</sup>fajar.antono988@gmail.com, <sup>2\*</sup>saruni.dwiasnati@mercubuana.ac.id

*Intisari*— Perusahaan yang berkembang merupakan perusahaan yang memiliki tata kelola perusahaan yang baik yaitu dengan penerapan presensi dan manajemen jam kerja karyawan. Perkembangan perusahaan juga didukung oleh perkembangan teknologi dimana teknologi informasi saat ini berkembang pesat dan cepat yaitu teknologi mobile. Komunikasi mobile dapat mempermudah masyarakat untuk memperoleh informasi yang cepat dengan akses internet. PT. Bangsawan Cyberindo merupakan perusahaan yang masih menerapkan presensi dan manajemen jam karyawan secara manual. Namun dalam pelaksanaannya, pihak manajemen masih kesulitan dalam memantau dan merekapitulasi data presensi dan jam kerja karyawan. Sehingga perusahaan membutuhkan suatu sistem yang tepat dan berguna agar dapat membantu pihak manajemen. Dengan demikian, Penerapan digitalisasi sistem presensi ini menggunakan fungsi Global Positioning System (GPS) pada perangkat Android dan menerapkan algoritma Haversine Formula untuk perhitungan jarak antara titik posisi kantor ke titik posisi user. Untuk menampilkan peta lokasi kantor pusat dan cabang terdekat, sistem terintegrasi dengan Google Map. Hasil pengujian dapat disimpulkan bahwa perhitungan jarak antara sistem dengan perhitungan yang dilakukan secara manual hanya berbeda  $\pm 0,0018$  meter.

*Kata kunci*—Android, Algoritma Haversine Formula, Global Positioning System, Presensi, Teknologi Mobile.

*Abstract*— A developing company is a company that has good corporate governance, namely by applying the presence and management of employee working hours. The development of the company is also supported by technological developments where information technology is currently growing rapidly and rapidly, namely mobile technology. Mobile communication can make it easier for people to get fast information with internet access. PT. Bangsawan Cyberindo is a company that still applies employee clock attendance and management manually. But in its implementation, management still has difficulty in monitoring and recapitulating attendance data and employee work hours. So that the company needs an appropriate and useful system in order to help management. Thus, the application of digitalization of this presence system uses the Global Positioning System (GPS) function on Android devices and applies the Haversine Formula algorithm to calculate the distance between the position of the office position and the user's position. To display a map of the location of the nearest head office and branch, the system is integrated with Google Map. The test results can be concluded that the calculation of the distance between systems with calculations carried out manually is only different  $\pm 0,0018$  meters.

*Keywords*—Android, Haversine Formula Algorithm, Presence, Mobile Technology, Global Positioning System

## I. PENDAHULUAN

Pada era globalisasi saat ini, perusahaan yang berkembang merupakan perusahaan yang memiliki tata kelola perusahaan yang baik. Apabila perusahaan memiliki tata kelola perusahaan yang baik, maka perusahaan tersebut dapat menentukan serta mencapai tujuan serta sasaran dari perusahaan tersebut. Salah satu aspek yang harus dipenuhi oleh perusahaan adalah kejelasan fungsi, struktur, sistem dan pertanggungjawaban organ perusahaan sehingga pengelolaan perusahaan terlaksana secara efektif[1]. Pada era modern ini smartphone telah menjadi kebutuhan sekunder bagi setiap orang, jumlah pengguna smartphone pun semakin meningkat. Berdasarkan data yang diperoleh dari [gs.statcounter.com](http://gs.statcounter.com) menunjukkan bahwa pengguna mobile device menempati posisi pertama dengan 51.95% melampaui pengguna desktop dan tablet dengan 43.98% dan 4.07%[2]. Salah satu contoh kegiatannya yaitu dengan penerapan presensi dan manajemen karyawan agar kejelasan sistem dan

kinerja karyawan pada perusahaan dapat dikelola dengan baik dan benar.

Dengan perkembangan teknologi yang semakin pesat saat ini, salah satunya adalah perkembangan teknologi komunikasi dengan smart phone. Smart phone memiliki banyak fitur didalamnya termasuk fitur GPS (Global Positioning System) dan fitur LBS (Location Based Service) untuk mengirimkan lokasi smart phone berada secara cepat dan akurat, maka dari itu penulis tertarik untuk mengembangkan sistem absensi berbasis smart phone khususnya Android [3].

PT. Bangsawan Cyberindo merupakan suatu organisasi atau perusahaan yang bergerak dalam bidang jasa pengembangan perangkat lunak yang berfokus pada peledakan jasa/barang di pemerintahan. Menurut pihak manajemen perusahaan, kebijakan perusahaan terkait presensi dan jam kerja karyawan di perusahaan ini masih terkendala, dikarenakan sebagian besar pekerjaan karyawan pada PT. Bangsawan Cyberindo ini berada diluar kantor atau onsite di tempat klien, dengan sistem presensi yang berjalan sekarang

ini masih dilakukan secara manual, dalam pelaksanaan kewajiban pencatatan kehadirannya setiap karyawan harus mengisi daftar hadir saat datang dan pulang kerja. Namun dalam pelaksanaannya kebanyakan dari karyawan akan langsung menuju ke lokasi pekerjaan klien daripada harus datang ke kantor terlebih dahulu untuk melakukan presensi. Dalam hal ini karena kebanyakan karyawan yang mengabaikan presensi kehadiran, dari pihak manajemen masih kesulitan dalam memantau dan merekapitulasi data presensi karyawannya.

Oleh sebab itu, pada penelitian ini dibuat aplikasi agar dapat melakukan presensi yang memiliki mobilitas tinggi dengan didukung smartphone berbasis android yang terintegrasi dengan fitur *Global Positioning System* (GPS) sebagai alat untuk memberikan informasi sesuai lokasi user berada, dan juga dengan penerapan algoritma *Haversine Formula* yang diperlukan dalam penelitian untuk menemukan solusi bagi masalah-masalah dan kondisi di atas. Android adalah sistem operasi yang dikeluarkan oleh Google khususnya untuk *smartphone* dan *tablet*[4]. *Haversine Formula* adalah rumus yang tepat dalam menghitung jarak antara dua titik yakni dengan inputan latitude dan longitude, sebagai titik awal dan akhir, maka akan dihitung jarak antara titik-titik yang berada didekatnya.

## II. METODOLOGI PENELITIAN

### 2.1 Metode Pengumpulan Data

Dalam penelitian ini menggunakan beberapa metodologi untuk mengumpulkan data yaitu sebagai berikut:

1. Wawancara, Dengan menggunakan metode ini penulis mencoba berdiskusi dengan pihak manajemen dan karyawan PT. Bangsawan Cyberindo dan memberikan beberapa pertanyaan untuk mengetahui permasalahan yang ada saat ini dan saling memberi masukan terkait sistem presensi yang akan dibuat.
2. Studi literatur, Dengan menggunakan metode ini penulis mempelajari dan mencari referensi mengenai pembuatan aplikasi presensi berbasis android. Referensi didapat dari buku, jurnal, artikel maupun penelitian sebelumnya.

### 2.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini diantaranya adalah :

#### 1. Metode *Haversine Formula*

Metode *Haversine* digunakan untuk menghitung jarak antara titik di permukaan bumi menggunakan garis lintang (*longitude*) dan garis bujur (*latitude*) sebagai variabel inputan. *Haversine formula* adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi). berdasarkan bujur dan lintang. Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari R 6.367, 45 km, dan lokasi dari 2 titik di koordinat bola (lintang dan bujur) masing-masing adalah lon1, lat1, dan lon2, lat2[5].

### Rumus Haversine

$$\begin{aligned} x &= (\text{lon2}-\text{lon1}) * \cos((\text{lat1}+\text{lat2})/2); \\ y &= (\text{lat2}-\text{lat1}); \\ d &= \text{sqrt}(x*x+y*y)*R \end{aligned}$$

### Keterangan:

x = Longitude (Lintang)  
y = Latitude (Bujur)  
d = Jarak  
R = Radius Bumi = 6371 km  
1 derajat = 0.0174532925 radian

### Contoh Perhitungan Rumus Haversine

Lokasi 1:

lat1 = -0.790175, lon1 = 119.800801

Lokasi 2:

lat2 = -0.8989, lon2 = 119.8428

lat1 = -0.790175 \* 0.0174532925  
radian = -0.013791155 radian  
lon1 = 119.800801 \* 0.0174532925  
radian = 2.090918422 radian

lat2 = -0.8989 \* 0.0174532925  
radian = -0.01569 radian  
lon2 = 119.8428 \* 0.0174532925  
radian = 2.091651 radian

$$\begin{aligned} x &= (\text{lon2}-\text{lon1}) * \cos((\text{lat1}+\text{lat2})/2) \\ &= (2.091651-2.090918422) * \cos((-0.013791155 + - \\ &0.01569)/2) \\ &= 0.0007329412 \\ y &= (\text{lat2}-\text{lat1}) \\ &= (-0.01569- (-0.013791155)) \\ &= -0.001897609 \\ d &= \text{sqrt}(x*x + y*y) * R \\ &= \text{sqrt}((0.0007329412*0.0007329412)+(-0.001897609*- \\ &0.001897609))*6371 \\ &= \text{sqrt}(0.0000041381) * 6371 \\ &= \mathbf{12.96012927 \text{ km}} \end{aligned}$$

#### 2. Metode Perancangan Sistem

Metode rekayasa peranti lunak yang digunakan peneliti adalah metode *Extreme Programming* (XP)[6], *Extreme Programming* telah dipublikasikan sejak 1999 oleh Kent Beck dengan menggunakan metode pendekatan berorientasi objek. Pada Gambar 1 terdapat 4 kerangka kegiatan dalam metode ini, dengan urutan proses sebagai berikut:

##### a. *Planning* (Perencanaan)

Pada tahap ini, pengumpulan data dan kebutuhan sangat penting untuk membantu kinerja pembuatan aplikasi. Dimulai dengan merancang gambaran umum aplikasi yang diperlukan seperti fitur dan fungsinya yang didiskusikan bersama pengguna aplikasi tersebut. Hal ini nantinya akan menciptakan sebuah user stories.

##### b. *Design*

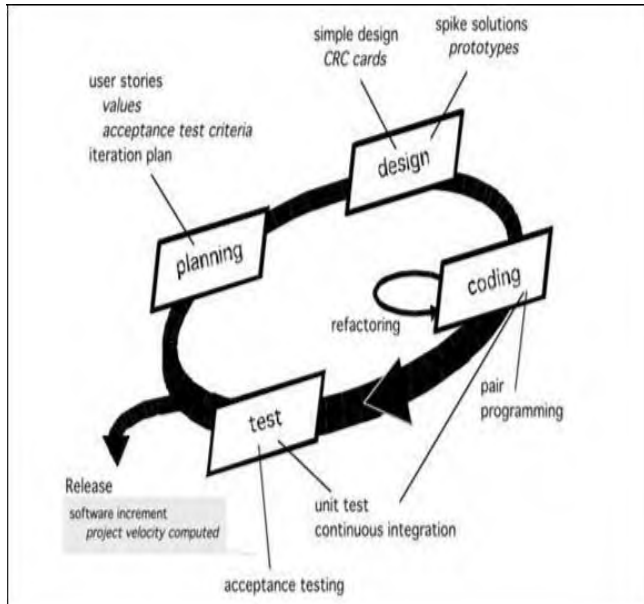
Pada tahap ini peneliti melakukan desain setelah kebutuhan selesai dikumpulkan secara lengkap.

c. *Coding*

Pada proses ini desain program diterjemahkan kedalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji dengan tujuan agar program dapat diuji dengan baik.

d. *Test (Pengujian)*

Pada proses ini adalah melakukan unit testing untuk pengujian kode supaya pengujian dapat dilakukan berulang-ulang dengan mudah setiap ada perubahan kode.



Gambar 1 Alur Pengembangan Sistem *Extreme Programming*

III. HASIL DAN PEMBAHASAN

Hasil dan pembahasan pada penelitian ini meliputi:

3.1. Perancangan *Database*

Pada aplikasi ini menggunakan database sebagai penyimpanan informasi dengan menggunakan tabel sebagai berikut :

a. Tabel *Users*

Tabel 1. Tabel *Users*

Nama Kolom	Tipe Data	Panjang	Keterangan
id	int	11	ID Urut Pengguna
name	varchar	50	Nama pengguna
username	varchar	50	Username Pengguna
password	varchar	100	Password Pengguna
email	varchar	50	Alamat E-mail Pengguna
phone	varchar	20	No. Handphone Pengguna
gender	enum	0	Jenis Kelamin Pengguna
date_of_birth	date	0	Tanggal Lahir Pengguna
province	int	11	Provinsi Pengguna
city	int	11	Kota Pengguna
address	text	0	Alamat Pengguna
usertype	int	2	Tipe Pengguna
active	enum	0	Status pengguna aktif
delete	enum	0	Status pengguna terhapus
photo	varchar	100	Foto profil Pengguna
ip_address	varchar	45	IP address Pengguna
salt	varchar	255	Salt Kunci Pengguna
activation_code	varchar	40	Kode Aktivasi Pengguna
forgotten_password	varchar	40	Kode Lupa Password
forgotten_password	int	11	Waktu Lupa Password
remember_code	varchar	40	Kode Ingatkan Saya
last_login	int	11	Log login Terakhir Pengguna
created_on	int	11	Waktu Pembuatan Pengguna
modified	timestamp	0	Waktu Perubahan Pengguna

Tabel *users* merupakan tempat penyimpanan data untuk *user administrator* yang dapat mengakses ke dalam *dashboard admin berbasis web*.

b. Tabel *Branch*

Tabel 2. Tabel *Branch*

Nama Kolom	Tipe Data	Panjang	Keterangan
id	int	11	ID urut kantor cabang
company	int	11	Nama perusahaan
branch_name	varchar	50	Nama kantor cabang
email	varchar	50	Alamat email kantor cabang
fax	varchar	30	Fax kantor cabang
phone	varchar	20	Telepon kantor cabang
province	int	11	Provinsi kantor cabang
city	int	11	Kota kantor cabang
subdistrict	int	11	Kecamatan kantor cabang
village	int	11	Kelurahan kantor cabang
address	text	0	Alamat kantor cabang
latitude	float8	40	Latitude kantor cabang
longitude	float8	40	Longitude kantor cabang
radius	varchar	10	Radius kantor cabang
is_deleted	enum	0	Status kantor cabang terhapus
primary	enum	0	Kantor cabang sebagai kantor pu

Tabel *branch* merupakan tempat penyimpanan data untuk kantor lokasi cabang atau penetapan area yang digunakan untuk penetapan lokasi presensi karyawan.

c. Tabel *Employees*

Tabel 3. Tabel *Employees*

Nama Kolom	Tipe Data	Panjang	Keterangan
id	int	32	ID Urut Karyawan
id_employee	varchar	30	No. Identitas Karyawan
fullName	varchar	50	Nama Lengkap Karyawan
username	varchar	50	Username Karyawan
password	varchar	100	Password Karyawan
email	varchar	40	Alamat E-mail Karyawan
phone	varchar	20	Nomor Handphone Karyawan
gender	Enum	0	Jenis Kelamin Karyawan
place_of_birth	varchar	30	Tempat Lahir Karyawan
date_of_birth	date	0	Tanggal Lahir Karyawan
province_id	int	11	Provinsi Karyawan
city_id	int	11	Kota Karyawan
subdistrict_id	int	11	Kecamatan Karyawan
village_id	int	11	Kelurahan Karyawan
address	text	0	Alamat Karyawan
company_id	int	11	Nama Perusahaan Karyawan
branch_id	int	11	Lokasi Cabang Karyawan
division_id	int	11	Divisi Karyawan
departement_id	int	11	Departemen Karyawan
position_id	int	11	Posisi Karyawan
join_start_date	date	0	Tanggal Masuk Karyawan
gcm_token	text	0	Kunci token karyawan
avatar	varchar	100	Foto Profile Karyawan
random	text	0	Kode Lupa Password
join_end_date	date	0	Tanggal Keluar Karyawan
active	enum	0	Status Aktif Karyawan
is_deleted	enum	0	Status Karyawan Terhapus

Tabel *employees* merupakan tempat penyimpanan data untuk karyawan agar dapat masuk ke dalam sistem presensi berbasis Android dan dapat melakukan kegiatan seperti *check in* dan *check out* untuk presensi kehadiran karyawan tersebut.

d. Tabel *Presence*

Tabel 4. Tabel *Presence*

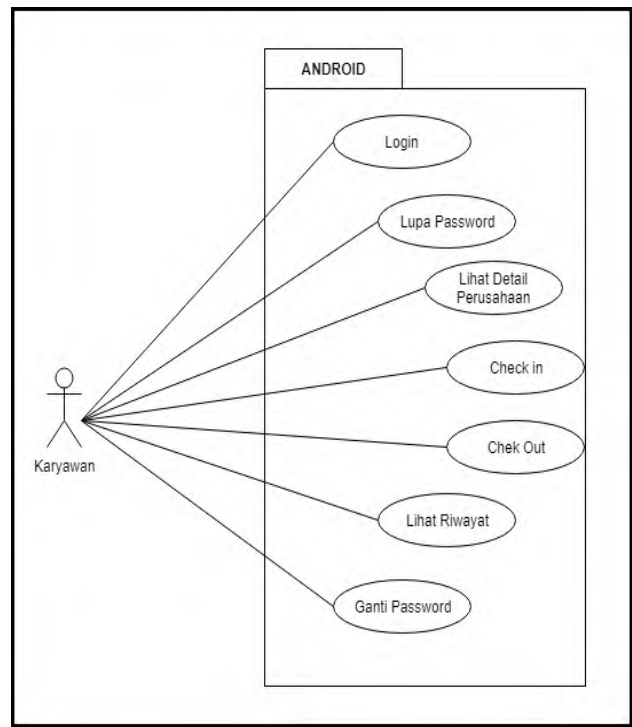
Nama Kolom	Tipe Data	Panjang	Keterangan
id	int	11	ID urut presensi
branch_id	int	11	Kantor cabang presensi karyawan
employee	int	11	Nama Karyawan
date	date	0	Tanggal presensi
photo_checkin	varchar	100	Foto masuk presensi karyawan
check_in	time	0	Jam masuk presensi
photo_checkout	varchar	100	Foto pulang presensi karyawan
check_out	time	0	Jam pulang presensi
latitude	float8	50	Latitude device karyawan
address	text	0	Alamat device karyawan
longitude	float8	50	Longitude device karyawan
province	varchar	100	Provinsi device karyawan
city	varchar	100	Kota device karyawan
state	varchar	100	Negara device karyawan
postal_code	varchar	10	Kode pos device karyawan
remark	text	0	Keterangan presensi karyawan

Tabel *presence* merupakan tempat penyimpanan data untuk *log* presensi karyawan jam masuk, jam pulang, presensi di cabang mana, dan di koordinat berapa karyawan tersebut melakukan presensi.

3. 2. Use Case Diagram

Perancangan ini akan menggambarkan aktifitas suatu *user* yang menggunakan sistem aplikasi dari pengamatan sudut pandang sederhana. Terdapat 2 rancangan *use case diagram* dalam sistem ini antara lain *use case* aplikasi android dan *use case* aplikasi *web* yang terdapat pada Gambar 2 dan Gambar 3.

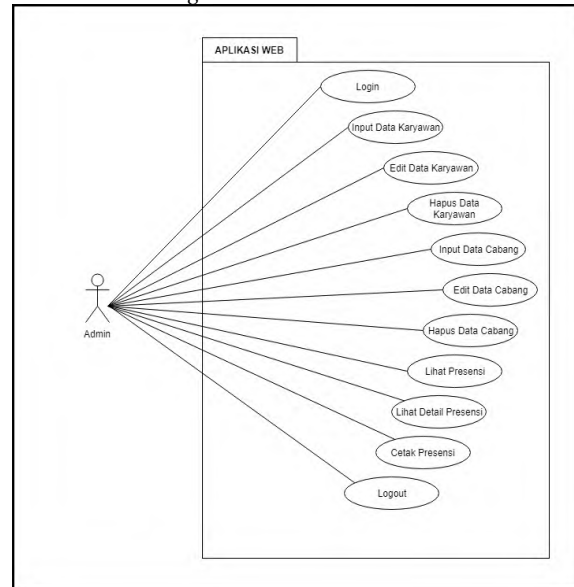
a. Use Case Diagram Karyawan



Gambar 2. Use Case Diagram Karyawan

Pada Gambar diatas, menunjukkan bahwa karyawan dapat melakukan aktivitas login, lupa password, lihat detail perusahaan, *check in*, *check out*, lihat riwayat, ganti *password*, dan *logout* pada saat mengoperasikan sistem aplikasi presensi pada android.

b. Use Case Diagram Admin



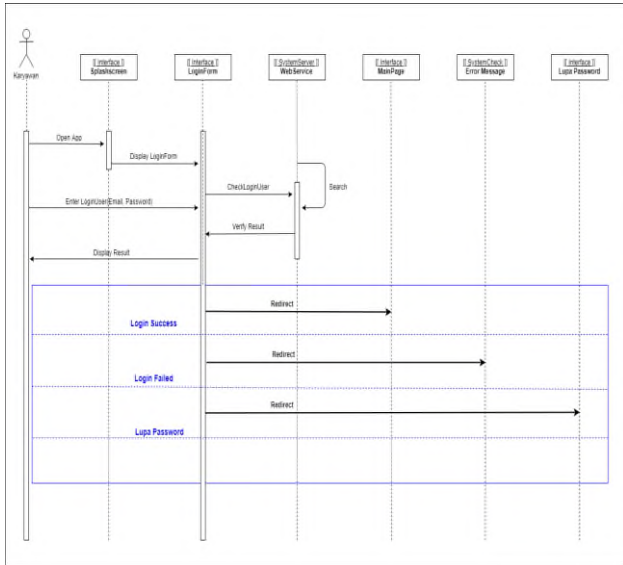
Gambar 3. Use Case Diagram Admin

Pada gambar diatas, aplikasi *web* dapat diakses oleh admin. admin dapat melakukan aktivitas *login*, input data kayawan, Edit data karyawan, Hapus data karyawan, input data cabang,

edit data cabang, Hapus data cabang, Lihat presensi, Lihat detail presensi, Cetak hasil presensi, dan Logout.

3. 4. Sequence Diagram

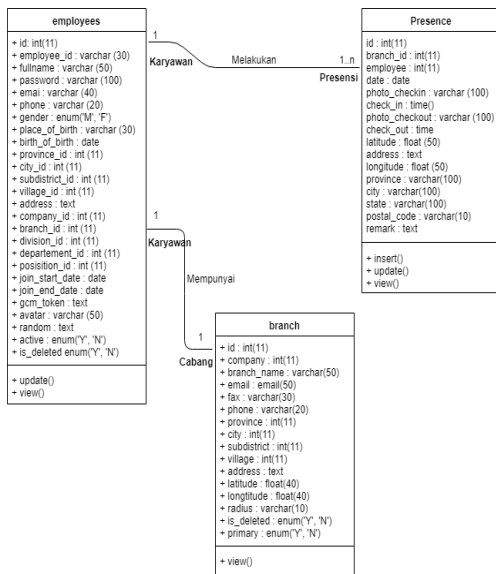
Sequence diagram akan menjelaskan cara kerja sistem dengan suatu urutan yang terperinci. Perancangan ini akan menunjukkan tahap-tahapan yang menggambarkan cara kerja sistem yang akan digunakan. Berikut adalah sequence diagram pada saat login.



Gambar 4. Sequence Diagram Login

3. 5. Class Diagram

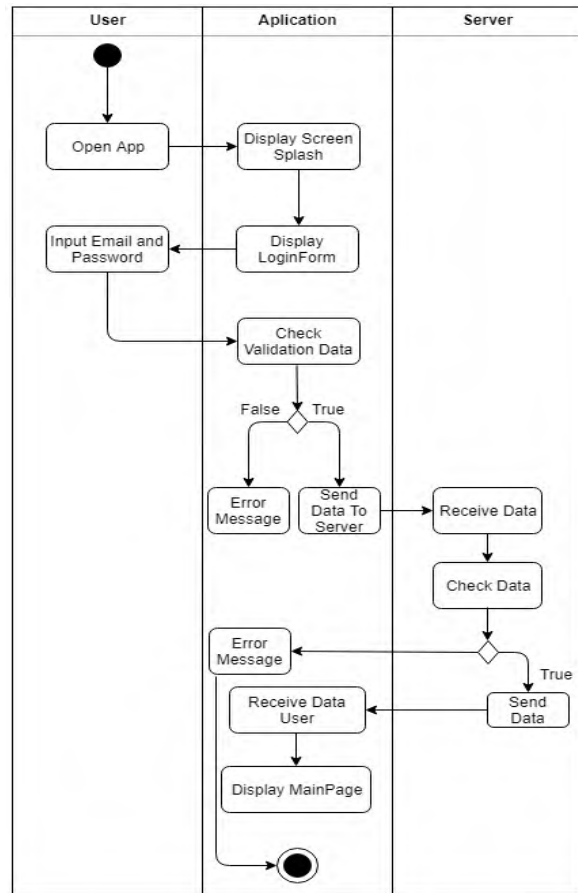
Pada perancangan ini, menjelaskan hubungan antara kelas yang digunakan dalam sistem aplikasi. Kelas “employees” berisi data-data karyawan yang akan melakukan presensi pada kelas “presence” dan setiap karyawan akan menentukan lokasi sesuai dengan lokasi terdekat karyawan melakukan absensi di cabang pada kelas “branch”. Perancangan class diagram dapat dilihat pada Gambar 5.



Gambar 5. Class Diagram

3. 6. Activity Diagram

Perancangan activity diagram ini akan dijelaskan dalam urutan-urutan seperti activity diagram login berikut ini :



Gambar 6. Activity Diagram Login

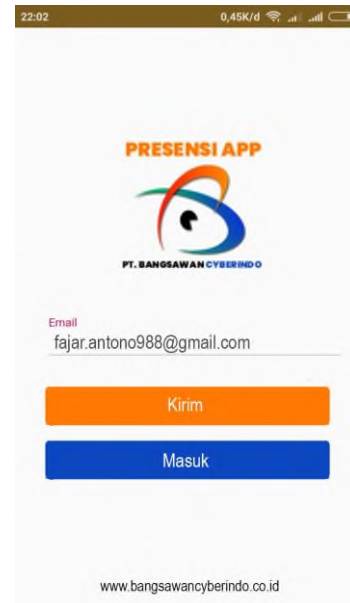
3. 7 Implementasi Sistem

Halaman Splash Screen

Merupakan tampilan awal aplikasi ketika pertama kali aplikasi dijalankan. Tampilan ini berguna sebagai indikator pengguna berhasil menjalankan aplikasi lalu mengalihkan ke halaman login.



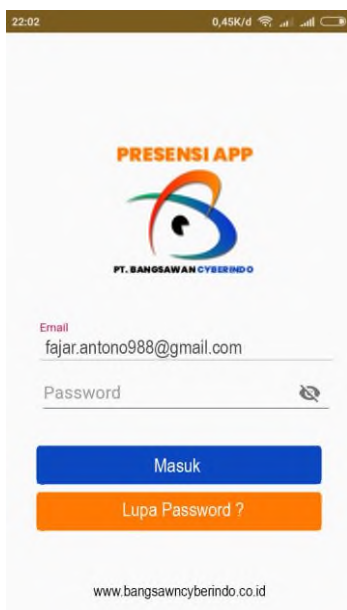
Gambar 7. Halaman *Splash Screen*



Gambar 9. Halaman lupa password

### Halaman *Login*

Pada tampilan *interface* ini, pengguna harus memasukkan data yang telah didaftarkan oleh administrator berupa *email* dan *password* agar dapat masuk ke halaman *interface* berikutnya.



Gambar 8. Halaman *Login*

### Halaman *Lupa Password*

Pada tampilan *interface* ini, pengguna harus memasukkan data berupa *email* yang telah didaftarkan oleh administrator. Setelah pengguna memasukan *email* yang terdaftar pada sistem, maka sistem akan mengirimkan sebuah *link* untuk merubah *password* ke *email* pengguna tersebut.

### Halaman *Beranda*

Pada tampilan *interface* ini, pengguna memasuki halaman utama aplikasi dimana pada halaman ini terdapat daftar cabang lokasi kantor. Ketika pengguna klik salah satu cabang maka akan berpindah ke halaman *presence* dengan indikator cabang tersebut.

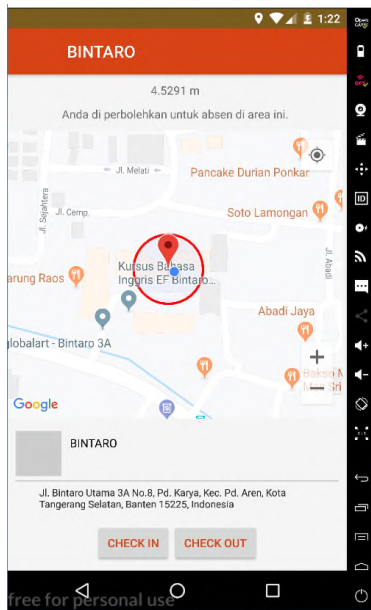


Gambar 10. Halaman *Beranda*

### Halaman *Presensi*

Pada tampilan *interface* ini, pengguna dapat melihat letak posisi pengguna tersebut dimana, didalam tampilan ini terdapat googlemap yang sudah di *marker* dengan lokasi cabang beserta jarak radiusnya. Pada halaman ini terdapat fungsi *check in* dan *check out*, dimana jika pengguna berada di

area kantor dan masih tercakup oleh radius. Maka pengguna tersebut bisa melakukan fungsi presensi jam masuk (*check in*) dan jam pulang (*check out*).



Gambar 11. Halaman Presensi

### Halaman Profil Perusahaan

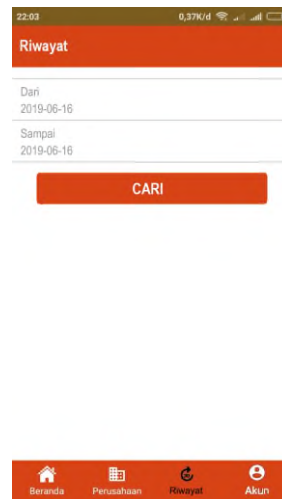
Pada tampilan *interface* ini, pengguna memasuki halaman informasi data perusahaan.



Gambar 12. Halaman Profil Perusahaan

### Halaman Riwayat

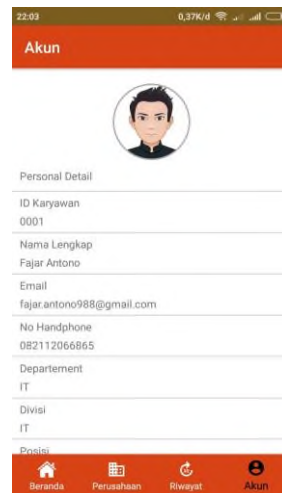
Pada tampilan *interface* ini, pengguna dapat melakukan pencarian presensi berdasarkan dengan tanggal. Hasil pencarian ini menampilkan dengan detail seperti tanggal, jam masuk, jam pulang, dan melakukan presensi di lokasi cabang mana.



Gambar 13. Halaman Riwayat

### Halaman Akun

Pada tampilan *interface* ini, pengguna dapat mengetahui informasi tentang data dirinya sebagai karyawan di perusahaan tersebut.



Gambar 14. Halaman Akun

### 3. 8 Pengujian Perhitungan Jarak

Pada pengujian perhitungan jarak akan diambil lima sampel titik posisi *user* kemudian akan dibandingkan dengan perhitungan pada sistem aplikasi dengan perhitungan secara manual. Berikut data sampelnya dapat dilihat pada table 5.

Tabel 5. Data sampel Posisi

No	Nama Kantor Cabang	Lokasi Kantor		Lokasi Karyawan	
		Latitude	Longitude	Latitude	Longitude
1	Kantor Pusat Bintaro	-6.268362	106.7375519	-6.268377	106.73759
2	Kantor Cabang Pinang Griya	-6.2194152	106.6918382	-6.2194298	106.6918318
3	Kantor BSD	-6.30476	106.7375519	-6.30458	106.7375511
4	KEMHANN (Onsite)	-6.1771136	106.821903	-6.1771128	106.821942
5	BKN (Onsite)	-6.258535	106.8678262	-6.258538	106.8678452

Lalu dilakukan pengujian perhitungan menggunakan rumus pada sistem yang telah diuji dengan menggunakan perintah SQL dan rumus perhitungan secara manual dengan rumus sebagai berikut :

1. Rumus perhitungan jarak pada sistem (Perintah SQL)

```
SELECT (6371 *
ACOS(SIN(RADIANS(lat1)) *
SIN(RADIANS(lat2)) +
COS(RADIANS(long1 - long2)) *
COS(RADIANS(lat1)) *
COS(RADIANS(lat2)))) AS distance
```

2. Rumus perhitungan jarak secara manual

```
x=(lon2-lon1)*cos((lat1+lat2)/2);
y=(lat2-lat1);
d= sqrt(x*x+y*y)*R
```

Apabila sudah diketahui kedua rumus tersebut, penulis melakukan perbandingan tersebut dalam Tabel 6.

Tabel 6. Perbandingan Perhitungan Rumus

No	Perhitungan Rumus pada Sistem	Perhitungan Rumus Secara Manual
1	<pre>SELECT (6371 * ACOS(SIN(RADIANS(- 6.268362)) * SIN(RADIANS(-6.268377)) + COS(RADIANS(106.7375519 - '106.73759')) * COS(RADIANS(-6.268362)) * COS(RADIANS(- 6.268377)))) AS distance  <b>HASIL :</b> <b>0.004529120 Km</b> <b>4.5291 m</b></pre>	<p><b>Lokasi 1</b> Lat1 = -6,268362 , Lon1 = 106,7375519</p> <p><b>Lokasi 2:</b> Lat2 = -6,268377, Lon2= 106,73759</p> <p>Lat1 = -6,268362 * 0.0174532925 Lat1 = -0.109403555 Lon1 = 106,7375519 * 0.0174532925 Lon1 = 1.862921714</p> <p>Lat2 = -6,268377 * 0.0174532925 Lat2 = -0.109403817 Lon2 =106,73759 * 0.0174532925 Lon2 = 1.862922379</p> <p><math>x = (lon2-lon1) * \cos((lat1+lat2)/2)</math></p>

		$x = (1.862922379 - 1.862921714) * \cos((-0.109403555 + (-0.109403817))/2)$ $x = 0.000000660995$ $y = (lat2-lat1)$ $y = (-0.109403817 - (-0.109403555))$ $y = -0.000000261799$ $d = \sqrt{x*x + y*y} * R$ $d = \sqrt{(0.000000660995*0.000000660995) + (-0.000000261799*-0.000000261799)} * 6371$ $d = 0.004529477 \text{ km}$ <p><b>d = 4.5295 m</b></p>
2.	<pre>SELECT (6371 * ACOS(SIN(RADIANS(- 6.2194152')) * SIN(RADIANS(-6.2194298')) + COS(RADIANS(106.6918382 - '106.6918318')) * COS(RADIANS(- 6.2194152')) * COS(RADIANS(- 6.2194298')))) AS distance  <b>HASIL :</b> <b>0.001770995 Km</b> <b>1.7710 m</b></pre>	<p><b>Lokasi 1</b> Lat1 = -6,2194152 , Lon1 = 106,6918382</p> <p><b>Lokasi 2:</b> Lat2 = -6,2194298, Lon2= 106,6918318</p> <p>Lat1 = -6,2194152 * 0.0174532925 Lat1 = -0.108549273 Lon1 = 106,6918382 * 0.0174532925 Lon1 = 1.862123859</p> <p>Lat2 = -6,2194298 * 0.0174532925 Lat2 = -0.108549527 Lon2 = 106,6918318 * 0.0174532925 Lon2 = 1.862123748</p> <p><math>x = (lon2-lon1) * \cos((lat1+lat2)/2)</math>  <math>x = (1.862123748 - 1.862123859) * \cos((-0.108549273 + (-0.108549527))/2)</math>  <math>x = -0.000000111044</math>  <math>y = (lat2-lat1)</math>  <math>y = (-0.108549527 - (-0.108549273))</math>  <math>y = -0.000000254818</math></p> $d = \sqrt{x*x + y*y} * R$ $d = \sqrt{(-0.000000111044*-0.000000111044) + (-0.000000254818*-0.000000254818)} * 6371$ $d = 0.001770897 \text{ km}$ <p><b>d = 1.7709 m</b></p>
3.	<pre>SELECT (6371 * ACOS(SIN(RADIANS(- 6.30476')) * SIN(RADIANS(- 6.30458')) + COS(RADIANS(106.7375519 - '106.7375511')) * COS(RADIANS(-6.30476')) * COS(RADIANS(-6.30458')))) AS distance  <b>HASIL :</b></pre>	<p><b>Lokasi 1:</b> Lat1 = -6,30476, Lon1 = 106,7375519</p> <p><b>Lokasi 2:</b> Lat2 = -6,30458, Lon2= 106,7375511</p> <p>Lat1 = -6,30476 * 0.0174532925 Lat1 = -0.11003882 Lon1 = 106,7375519 * 0.0174532925</p>



	<p><b>0.02001514 Km</b> <b>20.0151 m</b></p>	<p>Lon1 = 1.862921714</p> <p>Lat2 = -6,30458 * 0.0174532925 Lat2 = -0.110035679 Lon2 = 106,7375511* 0.0174532925 Lon2 = 1.8629217</p> <p><math>x = (\text{lon2} - \text{lon1}) * \cos((\text{lat1} + \text{lat2})/2)</math> <math>x = (1.8629217 - 1.862921714) * \cos((-0.11003882 + (-0.110035679))/2)</math> <math>x = -0.0000000138782</math></p> <p><math>y = (\text{lat2} - \text{lat1})</math> <math>y = (-0.110035679 - (-0.11003882))</math> <math>y = 0.00000314159</math></p> <p><math>d = \sqrt{x^2 + y^2} * R</math> <math>d = \sqrt{((-0.0000000138782 - 0.0000000138782) + (0.00000314159 * 0.00000314159)) * 6371}</math> <math>d = 0.020015282 \text{ km}</math> <b>d = 20.0152 m</b></p>
4.	<p>SELECT (6371 * ACOS(SIN(RADIANS(-6.1771136')) * SIN(RADIANS(-6.1771128')) + COS(RADIANS('106.821903' - '106.821942')) * COS(RADIANS(-6.1771136')) * COS(RADIANS(-6.1771128')))) AS distance</p> <p><b>HASIL :</b> <b>0.004314075 Km</b> <b>4.3141 m</b></p>	<p><b>Lokasi 1:</b> Lat1 = -6,1771136, Lon1 = 106,821903</p> <p><b>Lokasi 2:</b> Lat2 = -6,1771128, Lon2 = 106,821942</p> <p>Lat1 = -6,1771136 * 0.0174532925 Lat1 = -0.10781097 Lon1 = 106,821903 * 0.0174532925 Lon1 = 1.864393918</p> <p>Lat2 = -6,1771128* 0.0174532925 Lat2 = -0.107810957 Lon2 = 106,821942 * 0.0174532925 Lon2 = 1.864394599</p> <p><math>x = (\text{lon2} - \text{lon1}) * \cos((\text{lat1} + \text{lat2})/2)</math> <math>x = (1.864394599 - 1.864393918) * \cos((-0.10781097 + (-0.107810957))/2)</math> <math>x = 0.000000676726</math></p> <p><math>y = (\text{lat2} - \text{lat1})</math> <math>y = (-0.107810957 - (-0.10781097))</math> <math>y = 0.0000000139626</math></p> <p><math>d = \sqrt{x^2 + y^2} * R</math> <math>d = \sqrt{((0.000000676726 * 0.000000676726) + (0.0000000139626 * 0.0000000139626)) * 6371}</math> <math>d = 0.004312342 \text{ km}</math></p>

		<p><b>d = 4.3123 m</b></p>
5.	<p>SELECT (6371 * ACOS(SIN(RADIANS(-6.258535')) * SIN(RADIANS(-6.258538')) + COS(RADIANS('106.8678262' - '106.8678452')) * COS(RADIANS(-6.258535')) * COS(RADIANS(-6.258538')))) AS distance</p> <p><b>HASIL :</b> <b>0.002127059 Km</b> <b>2.1271 m</b></p>	<p><b>Lokasi 1:</b> Lat1 = -6,258535 , Lon1 = 106,8678262</p> <p><b>Lokasi 2:</b> Lat2 = -6,258538, Lon2 = 106,8678452</p> <p>Lat1 = -6,258535 * 0.0174532925 Lat1 = -0.109232042 Lon1 = 106,8678262 * 0.0174532925 Lon1 = 1.86519543</p> <p>Lat2 = -6,258538* 0.0174532925 Lat2 = -0.109232094 Lon2 = 106,8678452* 0.0174532925 Lon2 = 1.865195761</p> <p><math>x = (\text{lon2} - \text{lon1}) * \cos((\text{lat1} + \text{lat2})/2)</math> <math>x = (1.865195761 - 1.86519543) * \cos((-0.109232042 + (-0.109232094))/2)</math> <math>x = 0.000000329636</math></p> <p><math>y = (\text{lat2} - \text{lat1})</math> <math>y = (-0.109232094 - (-0.109232042))</math> <math>y = -0.0000000523599</math></p> <p><math>d = \sqrt{x^2 + y^2} * R</math> <math>d = \sqrt{((0.000000329636 * 0.000000329636) + (-0.0000000523599 * -0.0000000523599)) * 6371}</math> <math>d = 0.002126441 \text{ km}</math> <b>d = 2.1264 m</b></p>

Apabila perbandingan perhitungan jarak antara sistem dan manual telah dilakukan, maka penulis telah merangkum hasil dari perbandingan tersebut pada Tabel 7.

Tabel 7. Perbandingan Hasil Perhitungan

No	Hasil Perhitungan Sistem (m)	Hasil Perhitungan Secara Manual (m)	Gambar Output
----	------------------------------	-------------------------------------	---------------

1	4.5291 m	4.5295 m	
2.	1.7710 m	1.7709 m	
3.	20.0151 m	20.0152 m	
4.	4.3141m	4.3123 m	
5.	2.1271 m	2.1264 m	

Pada tabel diatas dapat dilihat hasil perhitungan jarak menggunakan rumus pada sistem dan perhitungan jarak secara manual, maka penulis dapat menarik kesimpulan bahwa dari lima sampel yang telah dihitung memiliki selisih letak hasil sebanyak 0.0018 meter.

#### IV. KESIMPULAN

Berdasarkan analisis dan pengujian yang dilakukan, maka kesimpulan yang dapat diambil adalah:

1. Penerapan Haversine formula mampu memberikan informasi jarak dari lokasi pengguna ke lokasi kantor yang digunakan sebagai objek.
2. Hasil dari penggunaan rumus perhitungan jarak menggunakan haversine formula pada sistem maupun secara manual tidak jauh berbeda, perbedaannya  $\pm 0.0018$  meter.
3. *Haversine Formula* dapat dimanfaatkan sebagai pencarian jarak terdekat dengan cara mencari hasil yang paling kecil nilainya sebagai lokasi dengan jarak terdekat.

#### REFRENSI

- [1]. Emirzon, J., (2006). Regulatory Driven Dalam Implementasi Prinsip-Prinsip Good Corporate Governance Pada Perusahaan Di Indonesia, [https://s3.amazonaws.com/academia.edu.documents/28873301/jurnal\\_mm\\_vol\\_4\\_no\\_8\\_artikel\\_4\\_joni\\_emirzon.pdf?AWSAccessKeyId=AKI](https://s3.amazonaws.com/academia.edu.documents/28873301/jurnal_mm_vol_4_no_8_artikel_4_joni_emirzon.pdf?AWSAccessKeyId=AKI)

AIWOWYYGZ2Y53UL3A&Expires=1554603280&Signature=pbOeU6I9HGnczsUbJVIs2X8YB%2F0%3D&response-content-disposition=inline%3B%20filename%3DRegulatory\_Driven\_Dalam\_I mplementasi\_Pri.pdf, diakses tanggal 07 April 2019.

- [2] Statcounter GlobalStats, "Operating System Market Share Worldwide | StatCounter Global Stats," Statcounter GlobalStats, 2018.
- [3] H. P. Utama, O. D. Nurhayati, and I. P. Windasari, "Pembuatan Aplikasi Memantau Lokasi Anak Berbasis Android Menggunakan Location Based Service," J. Teknol. dan Sist. Komput., vol. 4, no. 1, p. 202, 2016, doi: 10.14710/jtsiskom.4.1.2016.202-208.
- [4] Wawan, Gunawan. (2019). Pengembangan Aplikasi Berbasis Android Untuk Pengenalan Huruf Hijaiyah. JURNAL INFORMATIKA. Vol.6. No.1. April. ISSN: 2355-6579. E-ISSN: 2528-2247. Hal 69-76.
- [5] Prasetyo, Dwi dan Hastuti Khafizh. (2015). Penerapan Haversine Formula Pada Aplikasi Pencarian Lokasi dan Informasi Gereja Kristen Di Semarang Berbasis Mobile, UDiNus Repository, 2.
- [6] Pressman, R.S. (2010). Software Engineering: A Practitioner's Approach, Sevent Edition, The McGraw-Hill Companies Inc., New York.