

3. ANALISA DAN PEMBAHASAN

3.1 Identifikasi Masalah

Kegiatan pengisian presensi dapat membantu seorang manajer dalam merekap jumlah gaji bagi seorang pegawai. Namun sistem presensi bagi pegawai yang bekerja di lapangan, terkadang susah untuk di realisasikan. Karena pegawai lapangan bekerja tidak menetap di satu lokasi, sehingga absensi terkadang di lakukan secara manual dan dipantau oleh seorang manajer secara manual.

Kelemahan dari sistem absensi manual adalah membuat pekerjaan manajer menjadi lebih banyak, karena manajer harus memproses data presensi secara manual, dan melakukan pemantauan secara manual juga.

Beberapa perusahaan menyadari jika sistem manual tersebut sangat membuang waktu dan tenaga manajer, sehingga disusun sebuah sistem presensi dengan sidik jari. Namun sistem presensi dengan sidik jari kurang relevan dengan pegawai yang bekerja turun langsung ke lapangan, karena pegawai tidak menetap di satu lokasi secara permanen.

Dari masalah tersebut penulis menyadari harus ada solusi yang lebih tepat untuk melakukan kegiatan presensi dengan lebih efisien bagi karyawan yang bekerja di luar kantor. Penulis menyadari bahwa hal utama dalam sistem presensi adalah bagaimana cara agar data yang diterima valid dan bukan laporan presensi palsu, kemudian bagaimana cara agar data yang diterima manajer adalah data yang telah diolah, sehingga memudahkan manajer dalam memproses data tersebut menjadi hal yang lebih berguna dan dapat diteruskan ke sebuah sistem informasi lainnya seperti sistem informasi penggajian karyawan.

Dan untuk masalah yang tak luput dari perhatian penulis adalah bagaimana cara agar sistem informasi ini dapat digunakan oleh semua perusahaan. Oleh karena itu penulis membuat sistem informasi ini, dengan fleksibel dan terintegrasi dengan tiga tingkatan level akses.

Tingkatan level 1 atau admin dapat mengatur semua hal dari sistem presensi *FreeSent* yang penulis buat. Sehingga admin dapat melakukan berbagai hal seperti mengganti judul dari sistem presensi, mengubah logo, roles, mengatur user, mengatur tingkatan level user, mengatur menu, mengatur nama tingkatan level dan lain sebagainya.

Tingkatan level 2 atau manajer dapat mengatur karyawan yang dimilikinya, mengakses rekapan presensi, mengunduh rekapan dalam bentuk excel, mengatur rekapan presensi jika dinilai ada yang mencurigakan atau kesalahan.

Tingkatan level 3 atau user/pegawai dapat melihat rekapan dari presensi yang user laporkan ke sistem.

3.2 Analisa

1. Analisa Kebutuhan Sistem

Dalam membuat aplikasi ini, penulis membutuhkan beberapa kebutuhan *software, framework, dependency package, hardware, server dan smartphone* sebagai berikut:

1. Kebutuhan *software* yang digunakan, antara lain:
 - a. Sistem Operasi Ubuntu 16.04 xenial
 - b. Text Editor Atom versi 1.24.1
 - c. Browser Google Chrome Versi 65.0.3325.181
 - d. Apache2 HTTP Server
2. Kebutuhan *framework* yang digunakan, antara lain:
 - a. Laravel versi 5.5
 - b. React-Native

3. Kebutuhan *dependency package* yang digunakan, antara lain:
 - a. barryvdh/laravel-cors
 - b. fideloper/proxy
 - c. spatie/laravel-fractal
 - d. yajra/laravel-datatables-oracle
 - e. maatwebsite/excel
 - f. react-native-elements
 - g. react-native-vector-icons
 - h. react-navigation
 - i. native-base
4. Kebutuhan *hardware* yang digunakan, antara lain:
 - a. Intel Core i5-5200U CPU @ 2.7GHz
 - b. Kapasitas RAM 11929MB
 - c. Kapasitas Hardisk 500GB

5. Kebutuhan *Smartphone* yang digunakan, antara lain:

Digunakan lima (3) tipe *Smartphone* Android untuk menguji aplikasi

Tabel 3.1 : Tabel Tipe *Smartphone* untuk Menguji Aplikasi

Tipe	Merk	Ukuran Layar	RAM	Versi Android
Asus Zenfone Max Plus M1	Asus	5.7"	4 GB	7.0 (Nougat)
Asus Zenfone 5	Asus	5.0"	2 GB	4.4 (Kitkat)
Samsung Galaxy J5 (2016)	Samsung	5.2"	2 GB	6.0.1 (Marshmallow)

2. Analisa Kebutuhan Server

Penulis memerlukan server agar end-point api yang ada dapat diakses bagi semua, serta dapat melakukan pengujian sistem dengan menggunakan *production build* langsung di *server*.

Kebutuhan *server* yang digunakan, antara lain:

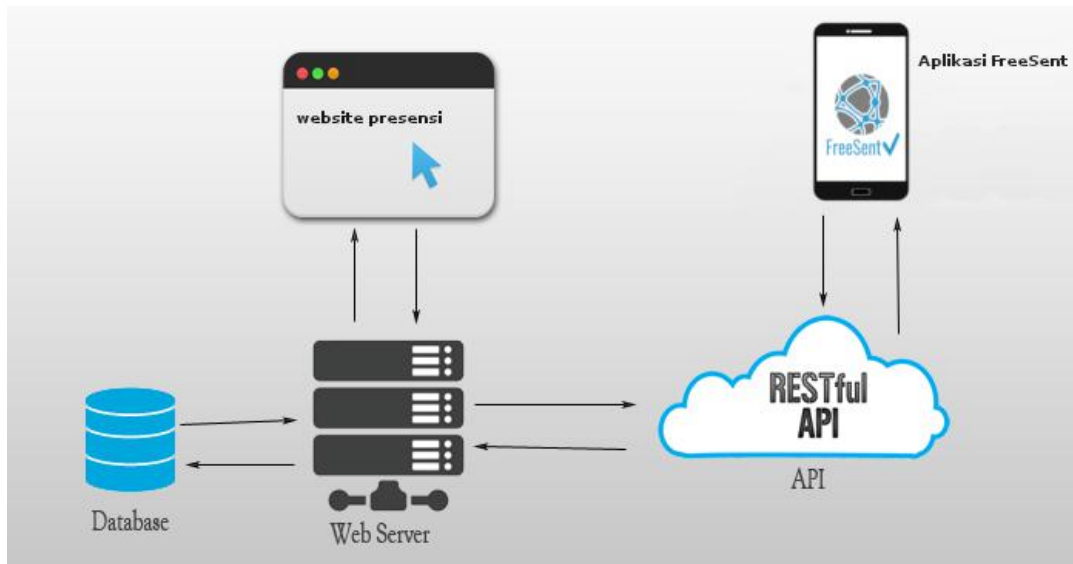
- Kapasitas Penyimpanan 500MB
- Kapasitas Penyimpanan Database 318MB
- Kapasitas Ram 1024MB
- PHP 7.0
- Akses git
- Akses ssh

3. Analisa Rest-API

Secara umum aplikasi sistem presensi yang dibuat akan menggunakan sistem *Rest-API* untuk aplikasi android bagi user, sedangkan untuk website bagi manajer penulis tidak menggunakan sistem *Rest-API*.

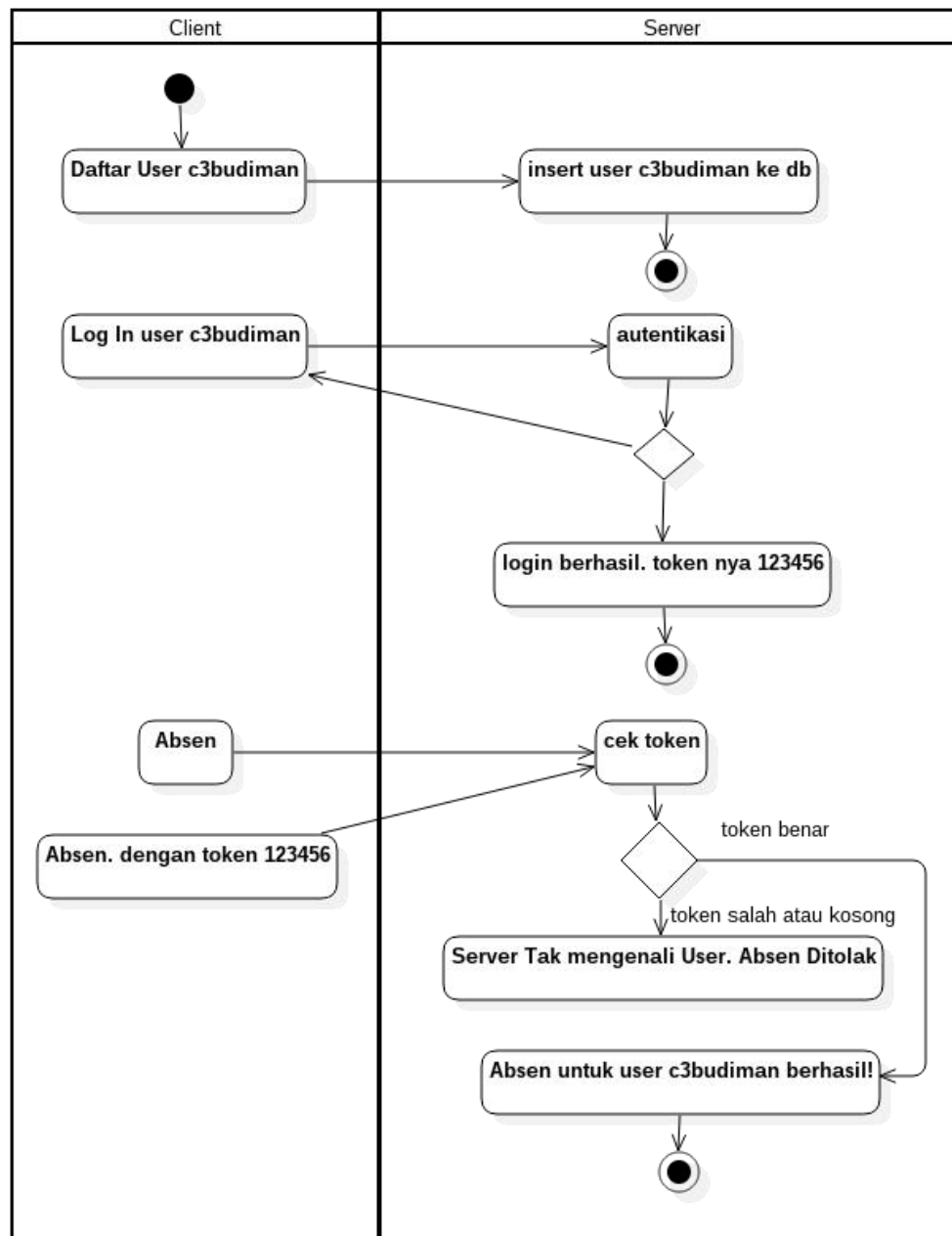
Bagi yang tidak familiar dengan sistem *Rest-API*, penulis akan menjelaskan bagaimana client yang disini adalah sebuah aplikasi *Android*, berkomunikasi dengan end-point api yang ada di server.

Rest Api akan membantu aplikasi android dalam penyimpanan dan pengaksesan data, seperti terlihat pada gambar 3.1 dibawah ini, Aplikasi FreeSent akan membuat request dalam protokol http yang berisi request/response dalam bentuk json ke dalam endpoint RESTful Api yang nantinya akan diteruskan ke web server untuk diolah dengan laravel yang kemudian akan memasukkan atau memanggil data yang ada di database.



Gambar 3.1 Deployment Diagram FreeSent

Activity Diagram Rest-api menggambarkan aliran kerja atau aktivitas client atau frontend dalam berinteraksi dengan server secara umum, untuk transaksi secara khusus nanti akan dijelaskan dibagian perancangan.

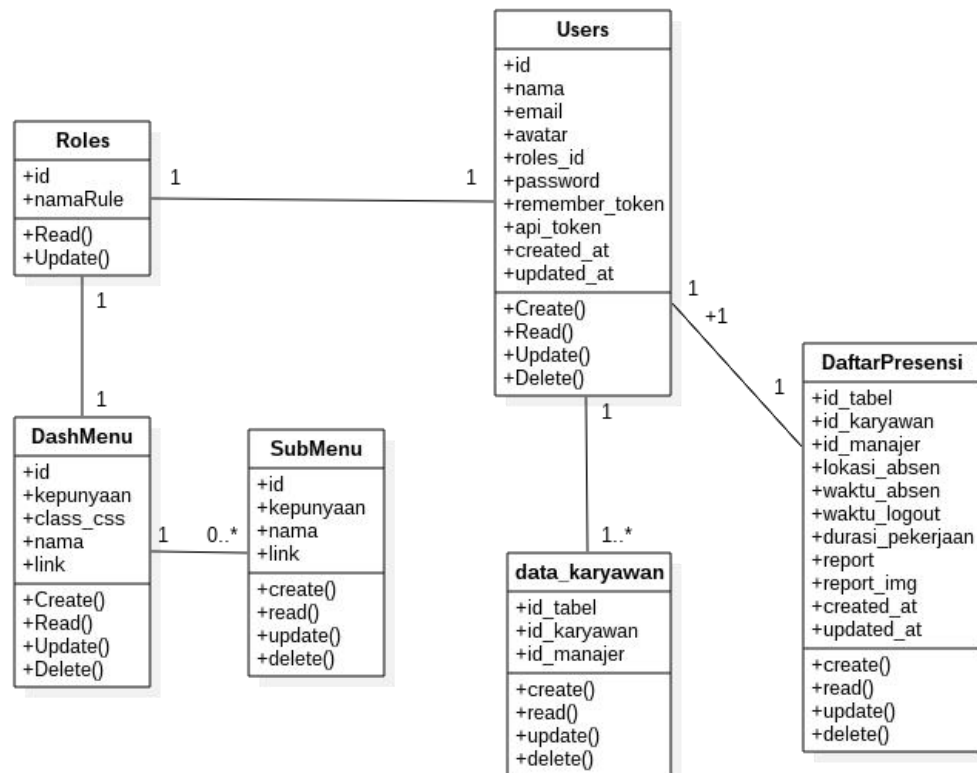


Gambar 3.2 Activity Diagram Interaksi Client-Server dengan konsep Rest-Api.

3.3 Perancangan Database

1. Rancangan Class Diagram

Dalam perancangan database, hal yang pertama penulis pikirkan adalah struktur dan relasi dari database, karena itu yang akan menentukan bagaimana sistem dapat dibentuk. *Class Diagram* menggambarkan kelas yang ada pada sebuah sistem. Masing-masing kelas terdapat atribut dan metoda. Atribut akan menggambarkan keadaan suatu sistem, sedangkan metoda digunakan untuk memanipulasi data.



Gambar 3.3 Class Diagram *FreeSent*

2. Rancangan File Migrasi

Dalam perancangan database, penulis menggunakan mysql dengan bantuan laravel-migration, sehingga seandainya akan mengganti database dari mysql ke oracle atau yang lainnya. Dapat dengan mudah menyesuaikan dan tidak harus membuatnya lagi.

Selain tabel, file migrasi juga dapat digunakan untuk mengisi record ke dalam tabel, sehingga data yang harus ada seperti menu navigasi bisa diletakkan secara langsung.

Namun dalam pembuatannya penulis harus melakukan pengkodean dalam bahasa laravel. Setiap tabel dan atribut nya akan dimasukkan ke dalam kode laravel, yang disebut schema.

Untuk membuat schema dan model dengan otomatis dapat menggunakan “php artisan make:model ‘nama_tabel’ -m”. Kemudian terdapat file migration yang diciptakan secara otomatis di bagian database->migrations. Untuk memasukkan hasil pengkodean ke database dapat menggunakan command “php artisan migrate”, jika terdapat kesalahan dapat menggunakan command “php artisan migrate:rollback” yang akan mengosongkan database.

A. Tabel users

```
Schema::create('users', function (Blueprint $table) {
    $table->increments('id');
    $table->string('nama');
    $table->string('email')->unique();
    $table->unsignedInteger('roles_id')->nullable();
    $table->string('avatar');
    $table->string('password');
    $table->string('api_token');
    $table->rememberToken();
    $table->timestamps();
});
```


B. Tabel roles

```
Schema::create('roles', function(Blueprint $kolom) {  
    $kolom->increments('id');  
    $kolom->string('namaRule');  
});
```

C. Relasi antara user dan roles

```
Schema::table('users', function(Blueprint $kolom){  
    $kolom->foreign('roles_id')->references('id')->  
>on('roles')->onDelete('cascade')->onUpdate('cascade');  
});
```

D. Tabel dashmenu

```
Schema::create('dashmenu', function (Blueprint $table) {  
    $table->increments('id');  
    $table->unsignedInteger('kepunyaan')->nullable();  
    $table->string('class_css');  
    $table->string('nama');  
    $table->string('link');  
});
```

E. Tabel submenu

```
Schema::create('submenu', function (Blueprint $table) {  
    $table->increments('id');  
    $table->unsignedInteger('kepunyaan')->nullable();  
    $table->string('nama');  
    $table->string('link');  
});
```

F. Relasi antara dashmenu dan submenu

```
Schema::table('submenu', function(Blueprint $kolom){
    $kolom->foreign('kepunyaan')->references('id')-
>on('dashmenu')->onDelete('cascade')->onUpdate('cascade');
});
```

G. Tabel setting_situses

```
Schema::create('setting_situses', function (Blueprint $table) {
    $table->increments('id');
    $table->string('namaSitus');
    $table->string('slogan');
    $table->string('favicon');
    $table->string('logo');
    $table->string('alamatSitus');
    $table->timestamps();
});
```

H. Tabel setting_situses

```
Schema::create('setting_situses', function (Blueprint $table) {
    $table->increments('id');
    $table->string('namaSitus');
    $table->string('slogan');
    $table->string('favicon');
    $table->string('logo');
    $table->string('alamatSitus');
    $table->timestamps();
});
```

I. Tabel data_karyawan

```
Schema::create('data_karyawan', function (Blueprint $table) {
    $table->unsignedInteger('id_karyawan')->nullable();
    $table->unsignedInteger('id_manajer')->nullable();
    $table->string('nama_karyawan');

});
```

J. Relasi data_karyawan dan users

```
Schema::table('data_karyawan', function(Blueprint $kolom){
    $kolom->foreign('id_karyawan')->references('id')->on('users')->
    >onDelete('cascade')->onUpdate('cascade');
    $kolom->foreign('id_manajer')->references('id')->on('users')->
    >onDelete('cascade')->onUpdate('cascade');
});
```

K. Tabel Daftar_presensis

```
Schema::create('daftar_presensis', function (Blueprint $table) {
    $table->increments('id');
    $table->unsignedInteger('id_manajer')->nullable();
    $table->unsignedInteger('id_karyawan')->nullable();
    $table->string('lokasi_absen');
    $table->timestamp('waktu_absen')->nullable();
    $table->timestamp('waktu_logout')->nullable();
    $table->time('durasi_pekerjaan')->nullable();
    $table->string('report');
    $table->string('report_img');
    $table->timestamps();

});
```

L. Relasi daftar_presensis dan user

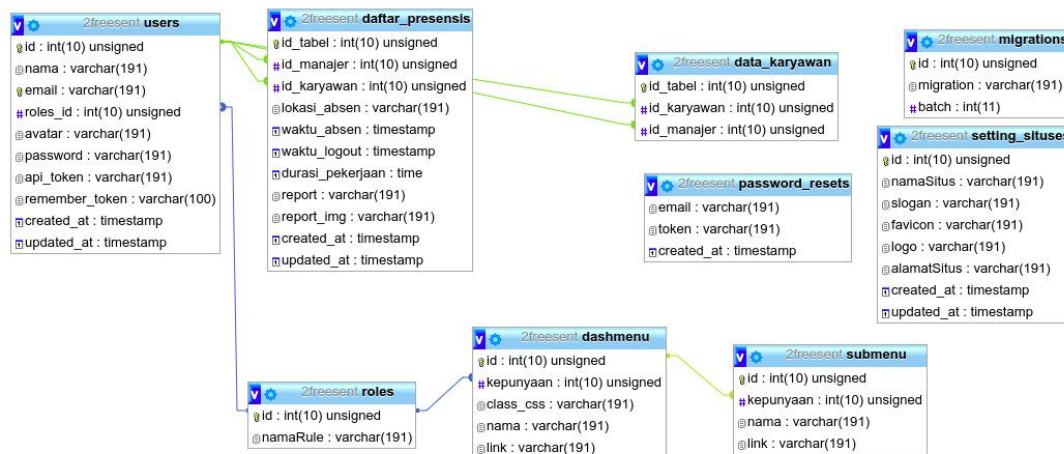
```
Schema::table('daftar_presensis', function(Blueprint $kolom){
    $kolom->foreign('id_manajer')->references('id')->on('users')->onDelete('cascade')->onUpdate('cascade');
    $kolom->foreign('id_karyawan')->references('id')->on('users')->onDelete('cascade')->onUpdate('cascade');
});
```

Menjalankan migrasi :

```
c3budiman@lenovo-g40s: ~/PI Cecep/absensi
c3budiman@lenovo-g40s:~/PI Cecep/absensi$ php artisan migrate
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2018_04_02_003237_sidebar
Migrated: 2018_04_02_003237_sidebar
Migrating: 2018_04_02_005020_submenu
Migrated: 2018_04_02_005020_submenu
Migrating: 2018_04_02_105929_create_setting_situses_table
Migrated: 2018_04_02_105929_create_setting_situses_table
Migrating: 2018_04_13_111307_create_karyawan_lists_table
Migrated: 2018_04_13_111307_create_karyawan_lists_table
Migrating: 2018_04_13_112013_create_daftar_presensis_table
Migrated: 2018_04_13_112013_create_daftar_presensis_table
c3budiman@lenovo-g40s:~/PI Cecep/absensi$
```

Gambar 3.4 Proses Migrasi

Hasil migrasi :



Gambar 3.5 Database *Freesent*

1. Tabel Database

Berdasarkan dari rancangan yang telah dibuat, *Platform Freesent* membutuhkan *database* berikut, yakni:

1. Tabel users

Tabel users, digunakan untuk menyimpan data-data pengguna dari situs, pengguna termasuk admin, manajer, dan karyawan. Pada tabel ini terdapat 10 *field* yang terdiri dari id, nama, email, roles_id, avatar, password, remember_token, api_token, created_at, dan updated_at. Strukturnya dijabarkan pada Tabel 3.1

Tabel 3.1 Tabel users

Field	Tipe Data	Panjang	Keterangan
id	Integer	10	Primary Key
nama	Varchar	191	
email	Varchar	191	
roles_id	int	10	
avatar	Varchar	191	
password	Varchar	191	
api_token	Varchar	191	
remember_token	Varchar	100	
created_at	timestamp		
updated_at	timestamp		

2. Tabel roles

Tabel roles, digunakan untuk menyimpan data tingkatan user/roles dari user. Tabel roles tidak berguna jika tidak digunakan oleh tabel user. Pada tabel ini terdapat 2 *field* yang terdiri dari id, dan namaRule. Strukturnya dijabarkan pada Tabel 3.2

Tabel 3.2 Tabel roles

Field	Tipe Data	Panjang	Keterangan
id	Integer	10	Primary Key
namaRule	Varchar	191	

3. Tabel dashmenu

Tabel dahsmenu, digunakan untuk menyimpan struktur navigasi pada semua tingkatan user. Pada tabel ini terdapat 5 *field* yang terdiri dari id, kepunyaan, class_css, nama, dan link. Strukturnya dijabarkan pada Tabel 3.3

Tabel 3.3 Tabel roles

Field	Tipe Data	Panjang	Keterangan
id	Integer	11	Primary Key
kepunyaan	Integer	11	
class_css	Varchar	191	
nama	Varchar	191	
link	Varchar	191	

4. Tabel submenu

Tabel submenu, digunakan untuk menyimpan struktur navigasi yang detail dari dashmenu pada semua tingkatan user. Pada tabel ini terdapat 4 *field* yang terdiri dari id, kepunyaan, nama dan link. Strukturnya dijabarkan pada Tabel 3.4

Tabel 3.4 Tabel roles

Field	Tipe Data	Panjang	Keterangan
id	Integer	11	Primary Key
kepunyaan	Integer	11	
nama	Varchar	191	
link	Varchar	191	

5. Tabel data_karyawan

Tabel data_karyawan, digunakan untuk menjelaskan hubungan antara manajer dan karyawan. Pada tabel ini terdapat 2 *field* yang terdiri dari id, id_manajer, dan id_karyawan. Strukturnya dijabarkan pada Tabel 3.5

Tabel 3.5 Tabel data_karyawan

Field	Tipe Data	Panjang	Keterangan
id_tabel	Integer	11	Primary Key
id_manajer	Integer	11	
id_karyawan	Integer	11	

6. Tabel daftar_presensi

Tabel daftar_presensi, digunakan untuk menyimpan data kehadiran dan rekapan jam kerja dari para karyawan. Pada tabel ini terdapat 11 *field* yang terdiri dari id, id_manajer, id_karyawan, lokasi_absen, waktu_absen, waktu_logout, durasi_pekerjaan, report, report_img, created_at dan updated_at. Strukturnya dijabarkan pada Tabel 3.1

Tabel 3.1 Tabel users

Field	Tipe Data	Panjang	Keterangan
Id_tabel	Integer	10	Primary Key
id_manajer	Integer	10	
id_karyawan	Integer	10	
lokasi_absen	Varchar	191	
waktu_absen	Timestamp		
waktu_logout	Timestamp		
durasi_pekerjaan	Time	100	
report	Varchar	191	
report_img	Varchar	191	
created_at	Timestamp		
updated_at	Timestamp		