

BAB 2

LANDASAN TEORI

2.1 Tempat Penelitian

PT. Metro Permata Raya merupakan perusahaan yang bergerak di bidang *property*. Produk yang kami hasilkan merupakan perumahan, *apartment* dan Ruko / Soho. PT. Metro Permata Raya sendiri merupakan salah satu anak perusahaan dari Margahayuland Development yang beralamat di Jalan Soekarno Hatta No. 693, Bandung, Jawa Barat 40286 Indonesia. Contact dari perusahaan ini (+62 22) 7533535. Perusahaan ini berdiri pada tanggal 10 Januari 2004 dengan nama PT. Metro Utama Raya. Logo perusahaan PT. Metro Permata Raya dapat dilihat pada Gambar 2.1 Logo Perusahaan.



PT. METRO PERMATA RAYA

Gambar 2.1 Logo Perusahaan

2.1.1 Visi dan Misi

Visi dan misi dari PT. Metro Permata Raya adalah sebagai berikut :

1. Visi

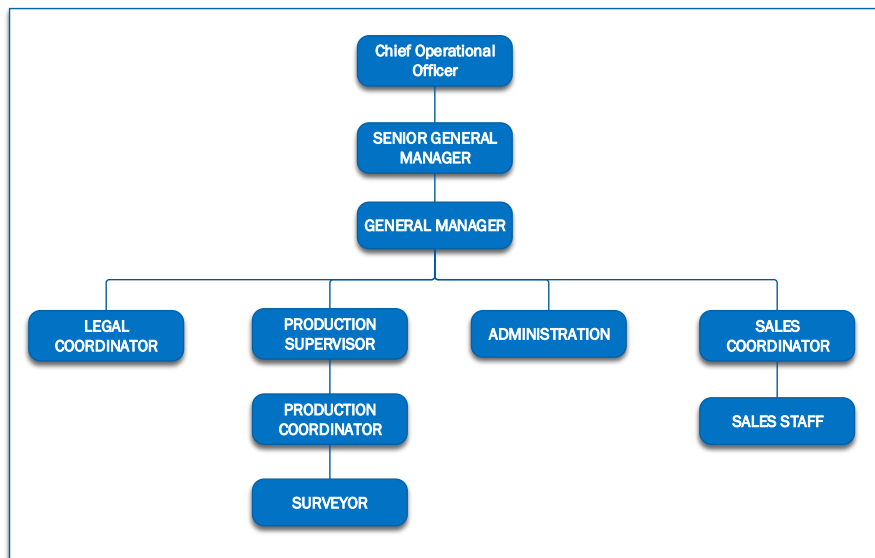
Menjadi perusahaan *property* yang terdepan dan berkelanjutan yang memberikan kepuasan dan kesenangan bagi pihak-pihak yang berkepentingan.

2. Misi

Ingin menciptakan sebuah tempat yang akan menjadi “*A great place to shop, work and invest*” dengan slogan ‘*build, grow & care*’.

2.1.2 Struktur Organisasi

Struktur organisasi PT. Metro Permata Raya dapat dilihat pada Gambar 2.2 Struktur Organisasi PT. Metro Permata Raya.



Gambar 2.2 Struktur Organisasi PT. Metro Permata Raya

2.1.3 Proyek yang sedang berjalan

Proyek-proyek PT. Metro Permata Raya adalah sebagai berikut :

1. De Marrakesh

De Marrakesh berlokasi di Jl. Derwati No 1, Bandung. Di bangun di atas lahan seluas 12,5 Ha dengan konsep unik yang pertama kali di Bandung, dimana bangunan rumah, fasilitas, dan lingkungan perumahan di bangun dengan “*Moroccan Style*”. Penghuni dapat menikmati suasana lingkungan nuansa *Maroko extraordinary* dengan pemandangan *riverside* dan kolam-kolam unik menjadikan De Marrakesh sebagai icon baru. Selain itu, kawasan hunian De Marrakesh terletak di lokasi strategis & prospektif, persis di jalan utama Derwati, Ciwastra *prime location* dan dilalui 3 transportasi umum sehingga mempermudah penghuninya menuju ke tempat aktivitas sehari-hari. Logo proyek De Marrakesh dapat dilihat pada Gambar 2.3 Logo Proyek De Marrakesh.



Gambar 2.3 Logo Proyek De Marrakesh

2. East Garden Residences

East Garden Residences berlokasi di Jl. Galumpit, Cileunyi, Kab. Bandung. East Garden Residences merupakan produk hunian terbaru dengan konsep *Modern Green Development*. Nuansa Modern diciptakan dengan memadukan keunggulan lokasi, fasilitas publik dan konsep lingkungan hijau. Logo proyek East Garden dapat dilihat pada Gambar 2.4 Logo Proyek East Garden Residences.



Gambar 2.4 Logo Proyek East Garden Residences

3. The Pakuan Mansion

The Pakuan Mansion berlokasi di Jl. Asri No. 1, Padasuka, Bandung. The Pakuan Mansion merupakan produk *Masterpiece* dengan konsep *Nuansa Rumah Bangsawan*. Nuansa kebangsawanan diciptakan dengan *men'stilasi'* keunggulan tradisi & teknologi modern. Dengan

adanya The Pakuan Mansion diharapkan dapat menjadi solusi kebutuhan eksklusivitas bertempat tinggal di sekitar Gedung Sate khususnya di Padasuka Bandung. Logo proyek pakuan mension dapat dilihat pada Gambar 2.5 Logo Proyek The Pakuan Mansion.



Gambar 2.5 Logo Proyek The Pakuan Mansion

2.2 Konsep Dasar Sistem

Pemahaman tentang sistem memiliki peranan yang penting dalam melakukan sebuah penelitian terhadap sistem yang akan diteliti serta untuk mencapai tujuan dan sasaran yang ingin dicapai. Menurut Abdul Kadir (2003:54), Sistem adalah sekumpulan elemen yang saling terpadu yang dimaksudkan untuk mencapai suatu tujuan [7]. Sedangkan menurut Muhammad Fakhri Husein dan Amin Wibowo (2002:210), Sistem adalah seperangkat komponen yang saling berhubungan dan saling bekerja sama untuk mencapai beberapa tujuan [8].

2.3 Internet

Internet atau singkatan dari *Inter-Network* adalah sekumpulan jaringan komputer yang saling terhubung dengan menggunakan protokol pertukaran paket (*packet switching communication protocol*) seperti *Transmission Control Protocol/Internet Protocol Suite* (TCP/IP). Internet menyediakan layanan telekomunikasi dan jutaan sumber daya informasi yang dapat diakses di seluruh dunia [9].

Jaringan yang membentuk internet bekerja berdasarkan suatu set protokol standar yang digunakan untuk menghubungkan jaringan komputer dan mengamati lalu lintas dalam jaringan. Protokol ini mengatur format data yang diijinkan, penanganan kesalahan (*error handling*), lalu lintas pesan, dan standar

komunikasi lainnya. Protokol standar pada internet dikenal sebagai TCP/IP (*Transmission Control Protocol/Internet Protocol*). Protokol ini memiliki kemampuan untuk bekerja diatas segala jenis komputer, tanpa terpengaruh oleh perbedaan perangkat keras maupun sistem operasi yang digunakan. Sebuah sistem komputer yang terhubung secara langsung ke jaringan memiliki nama domain dan alamat IP (*Internet Protocol*) dalam bentuk numerik dengan format tertentu sebagai pengenalan.

2.4 *Internet of Things (IoT)*

Internet of Things (IoT) merupakan teknologi yang memungkinkan adanya pengendalian, komunikasi, dan kerjasama antara berbagai jenis perangkat keras melalui internet, IoT muncul sebagai bentuk perubahan dan perkembangan teknologi informasi dan jaringan. IoT bukan hanya terkait pengendalian jarak jauh, tapi IoT juga berkaitan dengan bagaimana proses untuk berbagi data, memvirtualisasi segala hal nyata kedalam bentuk internet dan lain-lain [7].

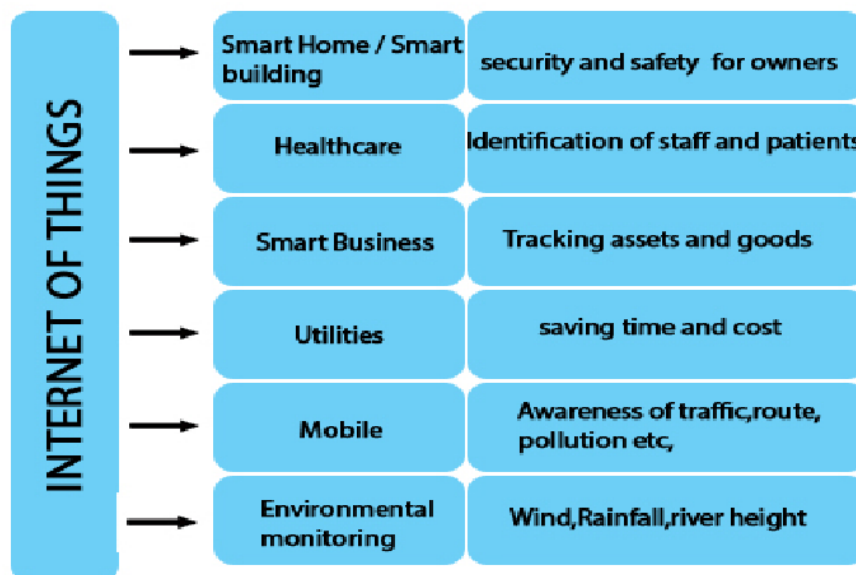
"A *Things*" pada *Internet of Things* dapat didefinisikan sebagai subjek misalkan orang dengan monitor implant jantung, hewan peternakan dengan *transponder biochip*, sebuah mobil yang telah dilengkapi *built-in* sensor untuk memperingatkan pengemudi ketika tekanan ban rendah. Sejauh ini, IoT paling erat hubungannya dengan komunikasi *machine-to-machine* (M2M) di bidang manufaktur dan listrik, perminyakan, dan gas. Produk dibangun dengan kemampuan komunikasi M2M yang sering disebut dengan sistem cerdas atau "*smart*". (contoh: *smart label, smart meter, smart grid sensor*).

Meskipun konsep ini kurang populer hingga tahun 1999, namun IoT telah dikembangkan selama beberapa dekade. Alat Internet pertama, misalnya, adalah mesin *Coke* di Carnegie Melon University di awal 1980-an. Para programmer dapat terhubung ke mesin melalui Internet, memeriksa status mesin dan menentukan apakah ada atau tidak minuman dingin yang menunggu mereka, tanpa harus pergi ke mesin tersebut.

Istilah IoT (*Internet of Things*) mulai dikenal tahun 1999 yang saat itu disebutkan pertama kalinya dalam sebuah presentasi oleh Kevin Ashton, *cofounder and executive director of the Auto-ID Center* di MIT.

Dengan semakin berkembangnya infrastruktur internet, maka kita menuju babak berikutnya, di mana bukan hanya smartphone atau komputer saja yang dapat terkoneksi dengan internet. Namun berbagai macam benda nyata akan terkoneksi dengan internet. Sebagai contohnya dapat berupa : mesin produksi, mobil, peralatan elektronik, peralatan yang dapat dikenakan manusia (*wearables*), dan termasuk benda nyata apa saja yang semuanya tersambung ke jaringan lokal dan global menggunakan sensor dan atau aktuator yang tertanam.

Pengaplikasian dari IoT bisa diklasifikasikan menjadi berbagai macam kegunaan yang bisa dilihat di Gambar 2.6 Contoh Pengaplikasian IoT.



Gambar 2.6 Contoh Pengaplikasian IoT [10]

Teknologi *internet of things* sangat luar biasa. Jika sudah direalisasikan, teknologi ini tentu akan sangat memudahkan pekerjaan manusia. Manusia tidak akan perlu lagi mengatur mesin saat menggunakannya, tetapi mesin tersebut akan dapat mengatur dirinya sendiri dan berinteraksi dengan mesin lain yang dapat berkolaborasi dengannya. Hal ini membuat mesin-mesin tersebut dapat bekerja

sendiri dan manusia dapat menikmati hasil kerja mesin-mesin tersebut tanpa harus repot-repot mengatur mereka

Kevin Ashton seorang pelopor teknologi yang juga membuat sistem standar global untuk RFID dan sensor lainnya mengatakan bahwa hampir semua data yang beredar di internet berasal dari hasil input atau hasil capture yang dilakukan oleh manusia ke dalam sistem. Dari sudut pandang sistem, manusia adalah obyek yang lambat, rawan kesalahan, pengantar data yang tidak efisien dan memiliki batasan dalam hal kualitas dan kuantitas, bahkan kadang mencoba menterjemahkan dan mengubah data tersebut. Sebagai alternatif akan lebih efisien jika sistem dapat terkoneksi dengan sensor yang dapat menterjemahkan kejadian di dunia nyata secara langsung. Jadi, di masa depan, sistem tidak memerlukan perantara manusia dan tersambung secara langsung ke sensor dan internet untuk mencatat data yang diambil dari dunia nyata

2.5 Raspberry Pi

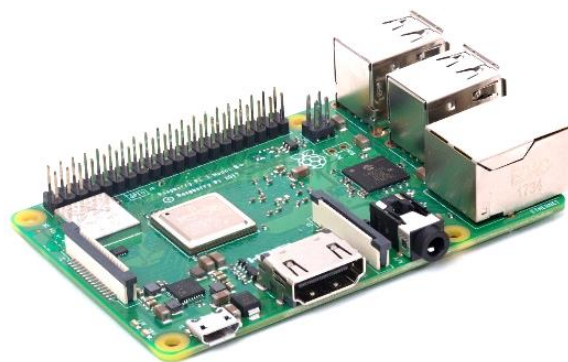
Raspberry Pi merupakan sebuah komputer berukuran kecil dan memiliki 2 model yaitu model A dan model B. Fungsi utama dari Raspberry Pi adalah sebagai komputer yang memungkinkan untuk browsing, membuat lapran, presentasi, bermain game, atau sekedar mendengarkan music dan nonton film. Raspberry Pi menggunakan *Python* sebagai bahasa pemrograman resminya [11].

Raspberry Pi adalah sebuah *SingleBoard Computer* yang menggunakan *system on a chip* (SoC) dari Broadcomm BCM2835, juga sudah termasuk prosesor ARM1176JZF-S 700 MHz, GPU Video Core IV dan RAM sebesar 256 MB (untuk Rev. A) dan 512 MB (untuk Rev. B) [12]. Dengan spesifikasi tersebut maka Raspberry Pi bisa digunakan untuk membuat dokumen, menghitung, menggambar, browsing internet, download, mencetak dokumen, memutar video, memutar musik, bermain game, dan lain-lain. Raspberry Pi yang memiliki ukuran kecil sangat praktis untuk dibawa kemana saja dan harganya relatif lebih murah dari pada komputer atau laptop.

Raspberry Pi tidak memerlukan energi listrik yang besar untuk beroperasi. Kita bisa membandingkan daya listrik dengan komputer di rumah yang

membutuhkan daya listrik mulai dari 200 *watt* [12]. Raspberry Pi hanya memerlukan daya yang kecil, oleh karena itu Raspberry Pi dapat dinyalakan hanya dengan charger handphone. Raspberry Pi dikembangkan oleh Raspberry pi Foundation. Ide dari pembuatan Raspberry pi muncul ketika sekelompok mahasiswa dari Universitas Cambridge, yaitu Eben Upton, Rob Mullins, Jack Lang, dan Alan Mycroft. Pada awalnya Raspberry Pi digunakan untuk membantu orang-orang di Afrika untuk belajar komputer dengan biaya yang sangat murah. Sehingga orang-orang yang tidak mempunyai cukup dana untuk membeli komputer bisa beralih ke Raspberry Pi sebagai alternatif. Dalam perkembangannya Raspberry Pi melakukan banyak inovasi sehingga menarik banyak perhatian selain itu kemampuan Raspberry pi juga sangat bisa diandalkan.

Kelebihan dari Raspberry Pi adalah terdapat pin-pin yang dapat diprogram untuk membantu berbagai macam pekerjaan, pin-pin ini dapat diprogram untuk mengumpulkan data-data atau mengendalikan sesuatu yang sudah deprogram dan sudah terhubung dengan Raspberry Pi. Raspberry Pi juga memiliki komunitas yang sangat besar yang dapat membantu dan menginspirasi kita dalam membuat project. Gambar dari Raspberry Pi dapat dilihat pada Gambar 2.8 Pin GPIO Raspberry Pi 3 Model B. gambar *board* Raspberry Pi dapat dilihat pada Gambar 2.7 Raspberry Pi 3 Model B



Gambar 2.7 Raspberry Pi 3 Model B [11]

2.1.4 Port I/O Raspberry Pi

Pada dasarnya port input dan output dari Raspberry Pi sama dengan port yang ada pada komputer, berikut daftar komponen dari Raspberry Pi :

1. Port USB

Pada Raspberry Pi terdapat 4 Port USB yang bisa digunakan untuk menyambungkan *keyboard*, *mouse*, modem, dan lain-lain. Raspberry Pi versi 1, 2, 3 masih mendukung USB 2.0. Setiap port USB menyediakan power sebesar 500mA [12].

2. Port LAN

Raspberry Pi juga memiliki Port LAN untuk dapat terhubung dengan jaringan local atau internet melalui kabel, dengan memanfaatkan Port LAN ini Raspberry Pi dapat terhubung dengan jaringan dan berbagi data.

3. Port HDMI

Port HDMI adalah port untuk menghubungkan Raspberry Pi ke *monitor* atau TV, salah satu kelebihan dari Raspberry Pi bisa menampilkan layar *full HD*.

4. Port composite untuk audio dan video (jack 3.5 mm)

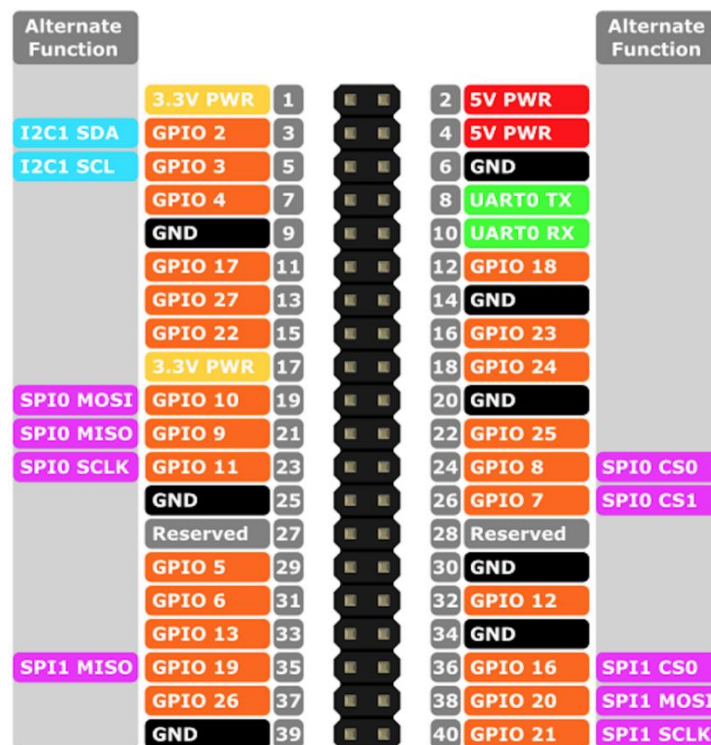
Melalui port ini kita bisa menggunakan menghubungkan Raspberry Pi dengan *speaker* atau *headset* untuk mendengarkan musik atau lainnya.

5. Slot Micro SD

Slot Micro SD adalah tempat untuk memasukan Micro SD yang berfungsi sebagai media penyimpanan Raspberry pi, sama halnya seperti harddisk pada komputer. Semua yang kita kerjakan di dalam Raspberry Pi akan disimpan di micro SD. Raspberry Pi membutuhkan kapasitas minimal 4GB untuk dapat beroperasi. Raspberry Pi memberikan sebuah utilitas baru untuk dapat menyalin (*clone*) Micro SD card. Sehingga lebih mudah bagi kita untuk backup atau memindahkan Micro SD ke unit Raspberry Pi yang lain. Micro SD rentan rusak (*corrupt*) apabila proses shutdown tidak berjalan sebagaimana mestinya.

6. Pin GPIO (*General Purpose Input Output*)

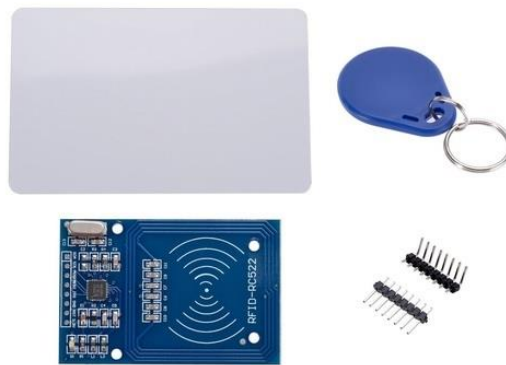
Pin-pin ini seperti pintu masuk dan keluarnya data atau sinyal. Masuk artinya digunakan untuk membaca atau mengumpulkan data yang ditangkap oleh sensor berupa sinyal input, sedangkan keluar berarti untuk mengeluarkan hasil berupa sinyal output. Menggunakan fungsi output adalah sama seperti menggunakan saklar lampu di rumah kita, jika kita tekan ke posisi ON, maka lampu menyala karena ada arus listrik yang mengalir ke lampu. Sebaliknya, arus listrik terputus saat kita tekan posisi OFF. Dengan Raspberry Pi, Saklar-saklar ini dikendalikan dengan bahasa program. Setiap pi memiliki identitas berupa penomoran dan juga fungsi spesifik. Berikut adalah Gambar 2.8 Pin GPIO Raspberry Pi 3 Model B.



Gambar 2.8 Pin GPIO Raspberry Pi 3 Model B

2.6 RFID

RFID (*Radio Frequency Identification*) merupakan teknologi digital dalam bentuk tag dan reader yang berbasis jaringan *wireless* (gelombang radio) untuk proses transfer data, identifikasi objek dan informasi elektronik lainnya [13]. Gambar RFID *reader* dan Tag RFID dapat dilihat pada Gambar 2.9 RFID MFRC522.



Gambar 2.9 RFID MFRC522 [13]

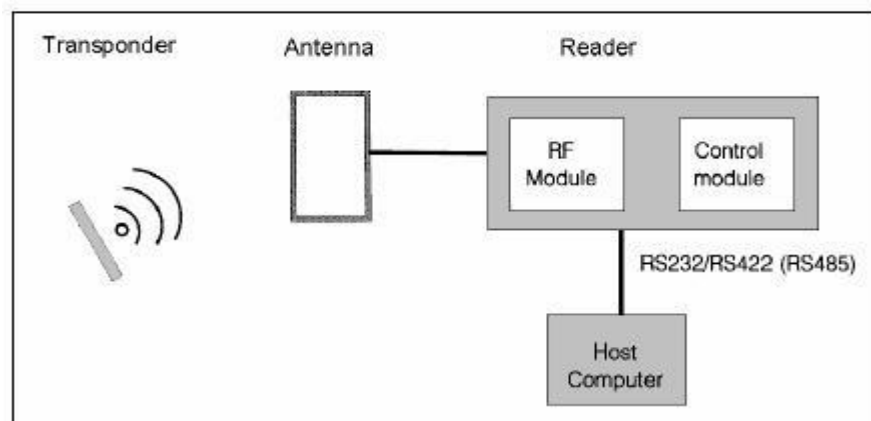
Teknologi RFID mudah digunakan, dan sangat cocok untuk operasi otomatis. RFID mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain. RFID dapat disediakan dalam perangkat yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read/Write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi lingkungan dan menyediakan tingkat integritasi yang tinggi [14].

Pada sistem RFID umumnya, tag atau transponder ditempelkan pada suatu objek. Setiap tag dapat membawa informasi yang unik, di antaranya: serial number, model, warna, tempat perakitan, dan data lain dari objek tersebut. Ketika tag ini melalui medan yang dihasilkan oleh pembaca RFID yang kompatibel, tag akan mentransmisikan informasi yang ada pada tag kepada pembaca RFID, sehingga proses identifikasi objek dapat dilakukan [14].

Sistem RFID terdiri dari empat komponen, di antaranya seperti dapat dilihat pada Gambar 2.10 Sistem RFID.

1. *Tag*: Ini adalah *device* yang menyimpan informasi untuk identifikasi objek. *Tag* RFID sering juga disebut sebagai transponder.
2. Antena: untuk mentransmisikan sinyal frekuensi radio antara pembaca RFID dengan *tag* RFID.
3. Pembaca RFID (*Micro-Reader*): adalah alat yang kompetibel dengan *tag* RFID yang akan berkomunikasi secara wireless dengan *tag*.
4. Software Aplikasi: adalah aplikasi pada sebuah *workstation* atau PC yang dapat membaca data dari *tag* melalui pembaca RFID. Baik *tag* dan pembaca RFID diperlengkapi dengan antena sehingga dapat menerima dan memancarkan gelombang elektromagnetik.

Gambaran dari sistem RFID dapat dilihat pada Gambar 2.10 Sistem RFID



Gambar 2.10 Sistem RFID

Walaupun teknologi RFID telah hadir selama hampir 20 tahun, belum ada standar data tunggal untuk satuan maupun untuk aplikasi industri. Sebagai tambahan terhadap biaya per label, ketiadaan suatu standar data yang jelas juga menjadi suatu faktor yang membatasi penggunaan RFID secara luas.

Berdasarkan cara daya tag, tag RFID (*Radio Frequency Identification*) dapat digolongkan menjadi:

1. Tag Pasif: yaitu tag yang daya diperoleh dari medan yang dihasilkan oleh pembaca RFID. Rangkaianannya lebih sederhana, harganya lebih murah, ukurannya kecil, dan lebih ringan. Kelemahannya adalah tag

hanya dapat mengirimkan informasi dalam jarak yang terbatas 4 - 5m ketika menggunakan frekuensi UHF (860 MHz– 930 MHz).

2. Tag Semi-Pasif: yaitu tag yang memiliki baterai terintegrasi dan oleh karena itu tidak memerlukan energi dari medan pembaca untuk menggerakkan chip itu. Ini memungkinkan tag untuk berfungsi dengan tingkatan sinyal yang lebih rendah, menghasilkan jarak yang lebih besar sampai kepada 100 meter. Jaraknya terbatas karena tag tidak mempunyai pemancar terintegrasi, dan masih perlu menggunakan medan pembaca untuk komunikasi kembali ke pembaca itu.
3. Tag Aktif: yaitu tag yang catu dayanya diperoleh dari baterai, sehingga akan mengurangi daya yang diperlukan oleh pembaca RFID dan tag dapat mengirimkan informasi dalam jarak yang lebih jauh (sampai beberapa kilometer). Kelemahan dari tipe tag ini adalah harganya yang mahal dan ukurannya yang lebih besar karena lebih kompleks. Semakin banyak fungsi yang dapat dilakukan oleh tag RFID maka rangkaiannya akan semakin kompleks dan ukurannya akan semakin besar.

2.1.5 Tag RFID

Tag RFID telah sering dipertimbangkan untuk digunakan sebagai barcode pada masa yang akan datang. Pembacaan informasi pada tag RFID tidak memerlukan kontak sama sekali. Karena kemampuan rangkaian terintegrasi yang modern, maka tag RFID dapat menyimpan jauh lebih banyak informasi dibandingkan dengan barcode. Fitur pembacaan jamak pada teknologi RFID sering disebut sebagai anti-collision.

Pembagian kelas RFID adalah berdasarkan kemampuan untuk membaca dan menulis data. EPC (Electronic Product Code) global mengelompokkannya menjadi lima kelas yaitu :

Class 0 – Read Only – Factory Programmed

Tipe ini adalah tipe yang paling sederhana, yang hanya mengandung nomor seri, EPC ditulis sekali ke dalam tag selama produksi. Datanya kemudian tidak

dapat diubah lagi. Kelas 0 juga digunakan untuk menentukan kategori tag yang disebut EAS (*Electronic Article Surveillance*) atau alat anti pencuri, yang tidak mempunyai nomor seri, hanya mendeteksi keberadaannya saat melewati antena.

Class 1

Dalam kelas ini, tag dihasilkan dengan tidak ada data yang ditulis ke dalam memori. Data kemudian bisa ditulis baik oleh pabrik tag atau oleh pemakai tetapi hanya sekali. Selanjutnya tag tidak dapat ditulis lagi dan tag hanya dapat dibaca. Tag jenis ini umumnya digunakan sebagai identifikasi sederhana.

Class 2

Tipe ini adalah tipe yang paling fleksibel, di mana pemakai mempunyai akses untuk membaca dan menulis data ke dalam memori tag. Tag ini biasa digunakan untuk mencatat data, dan oleh karena itu berisi lebih banyak memori dibanding dengan yang hanya digunakan untuk pengenalan sederhana.

Class 3

Tag ini berisi sensor terintegrasi untuk parameter perekaman seperti temperatur, tekanan, dan gerakan, yang dapat direkam dengan menulis ke dalam memori tag. Pembacaan sensor dapat diambil tanpa ada pembaca, tag dapat berupa tag aktif maupun tag semi-pasif.

Class 4

Tipe ini seperti miniatur radio yang dapat berkomunikasi dengan tag dan alat lain tanpa adanya suatu pembaca. Ini berarti mereka dengan sepenuhnya aktif dengan sumber baterai mereka sendiri.

2.7 Real-Time-Clock (RTC)

Real-time clock yang biasa disingkat dengan sebutan RTC merupakan sebuah *module* yang berfungsi untuk menyimpan data waktu pada komputer yang umumnya berupa sirkuit, RTC pada umumnya memiliki catu daya yang terpisah

dari catu daya komputer, RTC memiliki catu daya berupa baterai litium sehingga dapat tetap berfungsi ketika catu daya komputer terputus. Tipe RTC yang digunakan dalam penelitian ini adalah DS3231 yang memiliki spesifikasi sebagai berikut :

1. *Real time clock* (RTC) menyimpan data-data detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun valid hingga 2100.
2. Komunikasi antarmuka *serial two-wire* (I2C).
3. Sinyal keluaran gelombang kotak terprogram (*programmable squarewave*).
4. Ketahanan suhu 0°C hingga 70°C (komersial) dan -40°C hingga +85°C (industrial).
5. Memiliki *crystal oscillator internal*.

Gambar dari *real time clock* dapat dilihat pada Gambar 2.11 *Real Time Clock* (RTC) DS3231



Gambar 2.11 Real Time Clock (RTC) DS3231

RTC DS3231 memiliki 6 pin dengan deskripsi sebagai berikut :

1. 32K, berfungsi sebagai keluaran frekuensi 32 KHz.
2. SQW, berfungsi sebagai sinyal kotak (*square wave*) keluaran.
3. SCL, berfungsi sebagai serial data *clock*.
4. SDA, berfungsi sebagai serial data.
5. VCC, berfungsi sebagai catu daya positif.
6. GND, berfungsi sebagai catu daya negatif.

2.8 Pi Camera

Modul *Pi Camera* merupakan salah satu *accessories* pendukung Raspberry Pi keluaran distributor *RS Components & Premier Farnell / Element 14*. Kamera berukuran 5mp ini dilengkapi dengan kabel datar fleksibel yang berfungsi untuk dihubungkan pada konektor CSI yang terletak antara port ethernet dan port HDMI pada Raspberry Pi. Pada sistem operasi Raspbian, mengaktifkan pi camera dapat dilakukan dengan melakukan *install* dan *upgrade* versi terbaru dari Raspbian OS, kemudian jalankan *Raspi-config* dan pilih *Camera Option*.

Pi Camera dapat mengambil gambar dengan ukuran 1080p, 720p dan merekam video dengan ukuran sebesar 640x480p. Pi Camera memiliki dimensi 25mm x 20mm x 9mm. Kamera ini dapat bekerja dengan semua model Raspberry Pi 1, 2 dan 3. Dapat diakses melalui MMAL dan API V4L, juga terdapat banyak sekali *third-party libraries* yang mendukung, termasuk *Pi Camera Python library* [11].



Gambar 2.12 Pi Camera [11].

2.9 Pengenalan wajah (*Face Recognition*)

Face Recognition (pengenalan wajah) adalah aplikasi komputer atau mobile dimana aplikasi ini berjalan secara otomatis untuk mengidentifikasi atau memverifikasi seseorang dari gambar digital yang bersumber dari video. Pengenalan citra wajah dibagi menjadi 2 jenis, yaitu sistem *future based* dan sistem *image-based*. Pada sistem *future based* menggunakan fitur yang diekstraksi dari komponen wajah (mata, hidung, mulut, dll) yang kemudian dihubungkan dan

dimodelkan secara geometris. Sedangkan sistem *image-based* menggunakan informasi mentah dari piksel citra yang direpresentasikan dalam metode tertentu, seperti *principal component analysis* (PCA), transformasi *wavelet* yang kemudian digunakan untuk klasifikasi identitas citra [15].

Pada umumnya sistem *face recognition* mencakup empat modul utama yaitu: deteksi, *alignment*, ekstraksi fitur dan pencocokan. Proses lokalisasi dan normalisasi (deteksi wajah dan *alignment*) adalah langkah-langkah sebelum proses pengenalan wajah (ekstraksi fitur wajah dan pencocokan) dilakukan. Deteksi wajah adalah langkah awal untuk melakukan identifikasi wajah atau *face recognition*. Sebuah pendeteksi wajah yang ideal seharusnya mampu mengidentifikasi dan menentukan lokasi dan luas semua wajah yang ada di dalam sebuah gambar tanpa memperhatikan pose, skala, orientasi, umur, dan ekspresi. Deteksi wajah melakukan segmentasi area citra wajah dengan bagian latar [16].

Proses *alignment* (normalisasi) bertujuan untuk mempermudah dalam proses pendeksian. Apabila wajah sudah dinormalisasi, ekstraksi fitur dilakukan untuk mengambil data informasi mengenai posisi poin titik yang berguna untuk mengenali wajah satu sama lain. Pencocokan wajah dilakukan dengan cara melakukan pencocokan fitur yang telah diekstraksi dari citra wajah masukan dengan kumpulan data *training* dan uji coba citra wajah yang telah tersimpan pada database.

2.10 Algoritma Eigenface

Eigenface adalah kumpulan dari *eigenvector* yang digunakan untuk *computer vision* pada pengenalan wajah manusia, diambil dari analisis statistic dari banyak gambar wajah. *Eigenface* adalah salah satu algoritma pengenalan wajah yang didasarkan pada *Principal Component Analysis* (PCA) yang dikembangkan di MIT. Secara keseluruhan algoritma ini cukup sederhana, training image direpresentasikan dalam gabungan *vector* yang digabung menjadi sebuah matriks tunggal [15].

Algoritma Pengenalan wajah dimulai dengan matriks kolom dari wajah yang diinputkan ke dalam *database*. Rata-rata vector citra (*mean*) dari matriks kolom

dihitung dengan cara membaginya dengan jumlah banyaknya citra yang disimpan di dalam database. *Eigenface* sendiri merupakan sekumpulan *standaerdize face integredient* yang diperoleh dari *analisis statistic* dari banyak gambar wajah. Untuk menghasilkan nilai *eigenface* sekumpulan citra digital dari wajah manusia diambil pada kondisi pencahayaan sama kemudian dinormalisasikan dan diproses pada resolusi yang sama misal ($M \times N$).

2.11 Python

Python merupakan bahasa pemrograman yang freeware atau perangkat bebas dalam arti tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan source code-nya, debugger dan profiler, antarmuka, fungsi sistem, GUI (*Graphics User Interface*), dan baris datanya. Python menjadi bahasa resmi yang terintegritas dalam Raspberry pi. Nama Raspberry Pi sendiri merujuk pada kata “Python”, dikatakan bahwa python adalah bahasa natural Raspberry Pi [11]. Gambar logo python dapat dilihat pada Gambar 2.13 Logo Python.



Gambar 2.13 Logo Python

Python mendukung multi paradigma pemrograman, utamanya namun tidak dibatasi pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan

bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi.

2.12 JSON (JavaScript Object Notation)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 – Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data.

Penulis menggunakan metode JSON dalam pengiriman data yang dilakukan, karena JSON memiliki beberapa kelebihan - kelebihan dibandingkan XML, kelebihan – kelebihan tersebut adalah:

1. Format Penulisan

Untuk merepresentasikan sebuah struktur data yang rumit dan berbentuk hirarkis penulisan JSON relatif lebih terstruktur dan mudah.

2. Ukuran

Ukuran karakter yang dibutuhkan JSON lebih kecil dibandingkan XML untuk data yang sama. Hal ini tentu berpengaruh pula pada kecepatan pertukaran data, walaupun tidak signifikan untuk data yang kecil, namun cukup berarti jika koneksi yang digunakan relatif lambat untuk mengakses aplikasi web kaya fitur yang memanfaatkan pertukaran data. Di sini JSON lebih unggul dibandingkan XML, kecuali jika data dikompresi terlebih dahulu sebelum dikirimkan, perbedaan JSON dan XML yang telah dikompresi tidaklah signifikan.

3. Browser Parsing

Proses *parsing* merupakan proses pengenalan token atau bagian-bagian kecil dalam rangkaian dokumen XML/JSON. Contohnya, terdapat data text dalam format JSON. Data tersebut harus di-parsing terlebih dahulu sebelum dapat diakses dan dimanipulasi. *Browser parsing* berarti proses *parsing* yang terjadi pada sisi *client/browser*.

2.13 Flask

Flask adalah sebuah *web framework* yang ditulis dengan bahasa Python dan tergolong sebagai jenis *microframework*. Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. Dengan menggunakan Flask dan bahasa Python, pengembang dapat membuat sebuah web yang terstruktur dan dapat mengatur *behaviour* suatu web dengan lebih mudah. Logo Flask *Framework* dapat dilihat pada Gambar 2.14 Logo Flask.



Gambar 2.14 Logo Flask

Flask termasuk pada jenis *microframework* karena tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya. Sebagian besar fungsi dan komponen umum seperti validasi *form*, *database*, dan sebagainya tidak terpasang secara *default* di Flask. Hal ini dikarenakan fungsi dan komponen-komponen tersebut sudah disediakan oleh pihak ketiga dan Flask dapat menggunakan ekstensi

yang membuat fitur dan komponen-komponen tersebut seakan diimplementasikan oleh Flask sendiri.

Selain itu, meskipun Flask disebut sebagai *microframework*, bukan berarti Flask mempunyai kekurangan dalam hal fungsionalitas. *Microframework* disini berarti bahwa Flask bermaksud untuk membuat *core* dari aplikasi ini sesederhana mungkin tapi tetap dapat dengan mudah ditambahkan. Dengan begitu, fleksibilitas serta skalabilitas dari Flask dapat dikatakan cukup tinggi dibandingkan dengan *framework* lainnya

Web framework Flask ditulis menggunakan bahasa Python, sehingga sebelum Flask dapat digunakan, maka developer harus menginstall Python pada perangkat yang akan digunakan. Oleh sebab itu, *web developer* yang akan menggunakan Flask sebagai *web framework* untuk *web development* harus setidaknya mempelajari bahasa pemrograman Python terlebih dahulu, sebelum dapat menggunakan Flask seutuhnya

Dalam melakukan instalasi Flask pada sebuah perangkat, dibutuhkan PIP yang biasanya sudah terinstal pada Python versi 3.4 ke atas. PIP adalah sebuah *package management system* yang biasa digunakan untuk mengatur dan menginstall *package* yang berisi modul-modul Python. PIP digunakan untuk menginstall Flask karena Flask ditulis dan dikembangkan dengan bahasa dan modul-modul pemrograman Python. Dengan menggunakan PIP, semua hal yang dibutuhkan untuk instalasi Flask akan diunduh dan dipasang dalam satu perintah.

2.14 PHP

Personal Home Page atau sekarang biasa disebut PHP (*Hypertext Processor*) adalah *script* yang dijalankan di sisi server dengan konsep yang berbeda dengan JavaScript yang dijalankan pada sisi klien. Dalam penggunaan PHP, kode yang menyusun program tidak perlu dibagikan ke pemakai, yang berarti bahwa kerahasiaan kode dapat dilindungi sehingga menguntungkan bagi penggunanya.

Secara khusus, PHP dirancang untuk membentuk web dinamis yang dapat membentuk suatu tampilan berdasarkan permintaan terkini. Sebagai contoh, pengguna bisa menampilkan isi database ke halaman web. Pada prinsipnya,

PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *Cold Fusion* ataupun *Perl*.

Kenyataan bahwa PHP bisa digunakan untuk mengakses berbagai macam database seperti Access, Oracle, kode PHP dapat disisipkan pada kode HTML merupakan hal menarik yang didukung oleh PHP tetapi tidak mungkin dilakukan oleh JavaScript. Selain itu PHP juga bisa digunakan untuk menghasilkan kode-kode HTML [17]. Logo PHP dapat dilihat pada Gambar 2.15 Logo PHP.



Gambar 2.15 Logo PHP

Kelebihan dari PHP, yaitu :

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web server yang mendukung PHP dapat ditemukan dimana-mana dari mulai apache, IIS, Lighttpd, nginx, hingga Xitami dengan konfigurasi lebih mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di beberapa mesin (*Linux, Unix, Macintosh, Windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.15 MYSQL

MySQL adalah sebuah aplikasi Relational Database Management Server (RDBMS) bersifat open source yang memungkinkan data diakses dengan cepat oleh banyak pemakai secara bersamaan dan juga memungkinkan pembatasan akses pemakai berdasarkan privilege (hak akses) yang diberikan. MySQL menggunakan bahasa SQL (structured query language) yang merupakan bahasa standar pemograman database [17]. Logo *database* MySQL dapat dilihat pada Gambar 2.16 Logo MySQL.



Gambar 2.16 Logo MySQL

MySQL dipublikasikan sejak tahun 1996, yang sebenarnya sudah dikembangkan sejak tahun 1979. Selama tiga tahun berturut-turut, MySQL telah memenangkan penghargaan Linux Journal Reader's Choice Award. MySQL sekarang tersedia di bawah lisensi *open source* dan yang bersifat komersial.

Berikut berbagai keunggulan yang ditemukan di MySQL:

1. Skalabilitas, MySQL dapat menangani database yang besar, yang implementasinya telah dibuktikan dalam organisasi seperti HP, Cisco, Yahoo, NASA, Google, dan lain sebagainya.
2. Portabilitas, MySQL dapat berjalan pada berbagai macam sistem operasi termasuk Unix, Solaris, Mac OS, Windows, Linux serta pada arsitektur yang berbeda, mulai dari low-end PC sampai high-end mainframe.
2. Konektivitas, MySQL sepenuhnya mendukung jaringan yang dapat diakses dari mana saja di internet serta pengguna dapat mengakses database MySQL secara bersamaan. MySQL juga menyediakan berbagai

macam API (*Application Program Interface*) untuk mendukung konektivitas aplikasi yang ditulis dalam bahasa C, C++, C#, Perl, PHP, Java, Python, dan lain sebagainya.

3. Keamanan, MySQL mencakup seluruh keamanan yang kuat untuk mengontrol akses data dan juga mendukung *Secure Socket Layer (SSL Protocol)*.
4. Kecepatan, MySQL dikembangkan dengan kecepatan.
5. Mudah digunakan, MySQL mudah untuk digunakan dan diimplementasikan.
6. *Open Source*, MySQLAB membuat kode MySQL tersedia untuk digunakan setiap orang. filosofi open source memungkinkan khalayak global untuk berpartisipasi dalam pengembangan.

Sebuah DBMS/RDBMS tidak dapat lepas dari SQL (*Structured Query Language*). SQL merupakan sebuah bahasa yang digunakan untuk mengelola dan berinteraksi dengan data dalam database relasional. SQL adalah bahasa database yang paling universal digunakan, dan itu telah menjadi bahasa standar untuk manajemen database. SQL bekerja sama dengan sebuah RDBMS untuk mendefinisikan struktur dari database, menyimpan data di database tersebut, memanipulasi data, mengambil data, mengontrol akses ke data, dan menjamin integritas data.

Berikut beberapa contoh perintah dasar SQL yang sering digunakan pada MySQL :

1. Create Database, perintah yang digunakan membuat database baru.
Sintaks : `CREATE DATABASE DATABASE_NAME`
2. Drop Database, perintah yang digunakan untuk menghapus database.
Sintaks : `DROP TABLE TABEL_NAME`
3. Create Tabel, perintah yang digunakan untuk membuat table baru.
Sintaks : `Create Tabel tabel_name (create_definition)`
4. Describe, perintah yang digunakan untuk mendeskripsikan tabel.
Sintaks : `Describe (Desc) tabel [column]`

5. Alter Tabel, perintah yang digunakan untuk menghapus tabel.

Sintaks : Alter [Ignor] Tabel table_name

6. Drop Tabel, perintah yang digunakan untuk menghapus tabel.

Sintaks : Drop Tabel tabel_name [tabel_name..]

7. Delete, perintah yang digunakan untuk menghapus record dri tabel.

Sintaks : Delete From tabel_name Where Where_definiition

8. Select, perintah yang digunakan untuk query ke database.

Sintaks : select * from tabel_name

2.16 XAMPP

XAMPP merupakan salah satu paket instalasi Apache, PHP, dan MySQL secara instan yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut yang memiliki fungsi sebagai server yang berdiri sendiri (*localhost*). Beberapa Fitur yang ada pada XAMPP antara lain FTP, MySQL, Apache, PHP, CGI-BIN, FTP, Mercury Mail (SMTP), Perl MyAdmin dan My Admin [17]. Logo XAMPP dapat dilihat pada Gambar 2.17 Logo XAMPP .



XAMPP

Gambar 2.17 Logo XAMPP [17]

Bagian penting dari XAMPP yang sering digunakan:

1. htdoc adalah folder tempat meletakkan file yang akan dijalankan, seperti file PHP, HTML dan *script* lainnya.
2. phpMyAdmin merupakan bagian untuk mengelola database MySQL pada browser. Dengan mengakses alamat <http://localhost/phpMyAdmin>, maka akan muncul halaman phpMyAdmin.

3. Kontrol Panel yang berfungsi untuk mengelola layanan (*service*) XAMPP. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*).

2.17 CodeIgniter

CodeIgniter adalah sebuah *framework* bahasa pemrograman PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis semua kode program dari awal [18].

CodeIgniter pertama kali dibuat oleh Rick Ellis, CEO *Ellislab, Inc.* (<http://ellislab.com>), sebuah perusahaan yang memproduksi CMS (*Content Management System*) yang cukup handal, yaitu *Expression Engine* (<http://www.expressionengine.com>). Saat ini, CodeIgniter dikembangkan dan dimaintain oleh *Expression Engine Development Team*. Logo *framework* Codeigniter dapat dilihat pada Gambar 2.18 Logo CodeIgniter.



Gambar 2.18 Logo CodeIgniter [18]

Adapun beberapa keuntungan menggunakan CodeIgniter, diantaranya:

1. Gratis

CodeIgniter berlisensi dibawah Apache/BSD opensorce.

2. Ditulis Menggunakan PHP 4

Meskipun CodeIgniter dapat berjalan di PHP 5, namun sampai saat ini kode program CodeIgniter masih dibuat dengan menggunakan PHP 4.

3. Berukuran Kecil

Ukuran CodeIgniter yang kecil merupakan keunggulan tersendiri. Dibanding dengan framework lain yang berukuran besar.

4. Menggunakan Konsep MVC

CodeIgniter menggunakan konsep MVC yang memungkinkan pemisahan layer *application-logic* dan *presentation*.

5. URL yang Sederhana

Secara default, URL yang dihasilkan CodeIgniter sangat bersih dan *Search Engine Friendly* (SEF).

6. Memiliki Paket Library yang Lengkap

CodeIgniter mempunyai library yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis web, misalnya mengakses database, mengirim email, memvalidasi form, menangani session dan sebagainya.

7. Extensible

Sistem dapat dikembangkan dengan mudah menggunakan plugin dan helper, atau dengan menggunakan hooks.

8. Tidak Memerlukan Template Engine

Meskipun CodeIgniter dilengkapi dengan template parser sederhana yang dapat digunakan, tetapi hal ini tidak mengharuskan kita untuk menggunakannya.

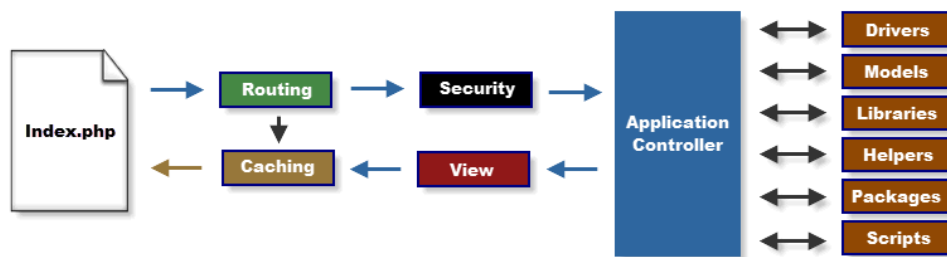
9. Dokumentasi Lengkap dan Jelas

Dari sekian banyak framework, CodeIgniter adalah satu-satunya framework dengan dokumentasi yang lengkap dan jelas.

10. Komunitas

Komunitas CodeIgniter saat ini berkembang pesat. Salah satu komunitasnya bisa dilihat di (<http://codeigniter.com/forum/>).

Proses aliran data aplikasi pada sistem dapat diilustrasikan seperti terlihat pada Gambar 2.19 Proses Aliran Data CodeIgniter.



Gambar 2.19 Proses Aliran Data CodeIgniter [18]

2.18 Web Browser

Web Browser adalah *software* yang digunakan untuk menampilkan informasi dari web server yang telah dikembangkan dengan menggunakan *Graphic User Interface* (GUI), sehingga pengguna dapat dengan mudah melakukan point dan click untuk pindah antar dokumen. Web browser yang masih menggunakan mode teks adalah Lynx, yang mengakibatkan tidak ada gambar yang dapat ditampilkan. Biasanya Lynx terdapat pada lingkungan DOS dan Unix. Akan tetapi perkembangan dari web browser dengan GUI. Saat ini ada banyak browser web GUI, antara lain Mozilla Firefox, Internet Explorer, Opera, dan Google Chrome. Dengan berusaha mendekati standar dokumen HTML yang direkomendasikan oleh W3C, beberapa browser tersebut bersaing untuk merebut pemakainya [17].

2.19 Unified Modelling Language (UML)

Pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek yaitu Unified Modeling Language (UML). UML merupakan bahasa visual yang menjadi standar untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak [19].

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep

dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak.

2.1.6 Usecase Diagram

Diagram usecase menggambarkan perilaku (*behavior*) sistem yang akan dibuat. Usecase mendeskripsikan sebuah interaksi antara satu aktor atau lebih dengan sistem yang akan dibuat. Secara umum, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi pada sistem [19].

Penamaan pada usecase didefinisikan sesederhana mungkin dan mudah untuk dipahami. Ada dua hal utama dalam use case, yaitu aktor dan use case.

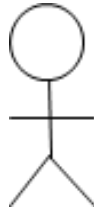
1. Aktor: merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem yang akan dibuat. Meskipun simbol dari aktor berbentuk orang, tapi aktor belum tentu merupakan orang.
2. Usecase: merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Terdapat beberapa simbol yang digunakan dalam pembuatan usecase diagram, berikut adalah symbol-simbol pada usecase diagram :

1. Aktor

Aktor dalam usecase diagram merupakan gambaran Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dikembangkan di luar sistem tersebut, sehingga meskipun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. Aktor

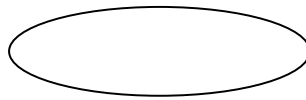
sering dinyatakan menggunakan kata benda, misal admin. Simbol aktor dapat dilihat pada Gambar 2.20 Simbol aktor.



Gambar 2.20 Simbol aktor

2. Usecase

Simbol usecase merupakan fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau actor, usecase sering dinyatakan dengan menggunakan kata kerja, misal melihat data. Simbol usecase dapat dilihat pada Gambar 2.21 Simbol Usecase.



Gambar 2.21 Simbol Usecase

3. Asosiasi

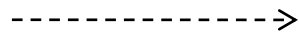
Asosiasi merupakan suatu garis yang menunjukkan komunikasi atau hubungan antara aktor dan usecase atau antar usecase satu dengan usecase lain. Simbol Asosiasi dapat dilihat pada Gambar 2.22 Simbol Asosiasi.



Gambar 2.22 Simbol Asosiasi

4. Extend

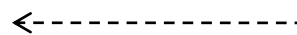
Extend menggambarkan suatu hubungan usecase tambahan ke usecase dimana usecase yang ditambahkan dapat berdiri sendiri walau tanpa usecase utama, anak panah pada garis menuju ke usecase yang dituju. Simbol extend dapat dilihat pada Gambar 2.23 Simbol Extend.



Gambar 2.23 Simbol Extend

5. Include

Include menggambarkan suatu hubungan antara usecase tambahan ke sebuah usecase dimana usecase yang ditambahkan memerlukan usecase utama untuk menjalankan fungsinya, sehingga usecase harus dijalankan terlebih dulu sebelum menjalankan usecase tambahan, arah panah menunjuk pada use case yang perlu dijalankan terlebih dahulu. Simbol include dapat dilihat pada Gambar 2.24 Simbol Include.



Gambar 2.24 Simbol Include

2.1.7 Activity Diagram

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem, tetapi bukan aktivitas aktor. Diagram aktivitas juga menggambarkan bagaimana alur sistem berawal, pilihan (*decision*) yang mungkin terjadi, dan bagaimana akhir alur sistem tersebut [19]. Simbol-simbol yang digunakan pada aktivitas diagram adalah sebagai berikut:

1. Status awal

Status awal pada diagram aktivitas menggambarkan dimulainya suatu aktivitas sistem, sebuah diagram aktivitas memiliki

sebuah status awal yang akan dilanjutkan dengan aktivitas pertama pada sistem. Simbol status awal dapat dilihat pada Gambar 2.25 Simbol Status Awal.



Gambar 2.25 Simbol Status Awal

2. Aktivitas

Simbol aktivitas pada diagram aktivitas menggambarkan suatu aktivitas yang dilakukan pada suatu system, penamaan aktivitas biasanya diawali dengan kata kerja seperti menambah data. Simbol aktivitas dapat dilihat pada Gambar 2.26 Simbol Aktivitas.



Gambar 2.26 Simbol Aktivitas

3. Percabangan

Simbol percabangan (*decision*) menggambarkan asosiasi percabangan dimana jika terdapat suatu kondisi pilihan aktivitas lebih dari satu dan menjelaskan beberapa kemungkinan yang terjadi setelahnya. Simbol percabangan dapat dilihat pada Gambar 2.27 Simbol Percabangan.



Gambar 2.27 Simbol Percabangan

4. Status Akhir

Status akhir pada diagram aktivitas menggambarkan selesainya suatu aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status akhir yang menggambarkan bahwa aktivitas pada system telah selesai. Simbol status akhir dapat dilihat pada Gambar 2.28 Simbol Status akhir.



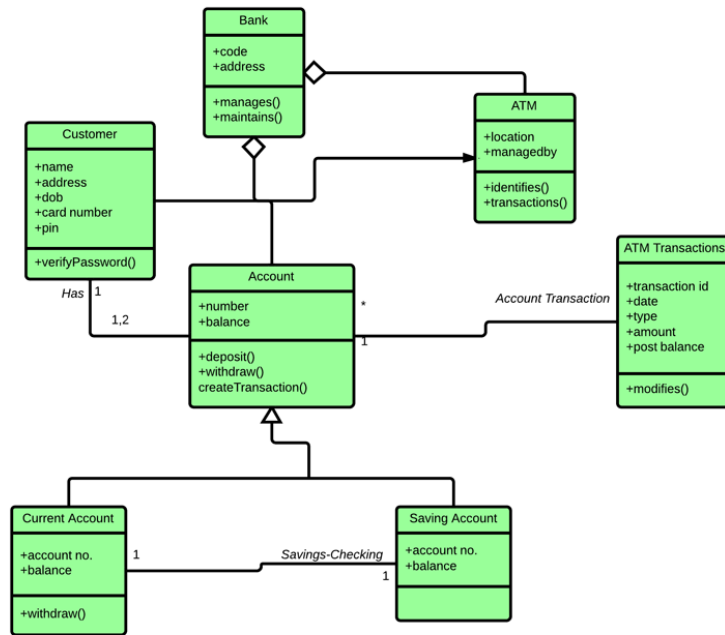
Gambar 2.28 Simbol Status akhir

2.1.8 Sequence Diagram

Sequence Diagram merupakan diagram yang digunakan untuk menggambarkan perilaku objek pada usecase dengan mendeskripsikan proses objek dengan pesan yang dikirimkan. Oleh karena itu, untuk menggambar diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah usecase beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu [19].

2.1.9 Class Diagram

Class diagram digunakan untuk menggambarkan struktur dalam objek sistem. Diagram ini menunjukkan class object yang menyusun sistem dan juga hubungan antara class object. Kelas memiliki atribut dan metode atau operasi. Atribut adalah variabel-variabel yang mendeskripsikan properti dengan bentuk sebaris teks dalam kelas tersebut, sedangkan metode adalah fungsi yang dimiliki oleh kelas yang dalam class diagram dilambangkan menggunakan simbol-simbol [19]. Gambar contoh class diagram dapat dilihat pada Gambar 2.29 Contoh Class Diagram.



Gambar 2.29 Contoh Class Diagram