

# **1. PENDAHULUAN**

## **1.1. Latar Belakang**

Perkembangan jaman seperti sekarang ini, banyak hal yang telah dilakukan oleh makhluk hidup dengan tingkat tertinggi dalam melakukan berbagai hal di kehidupan sehari-hari. Terobosan yang dilakukan manusia banyak terjadi pada bidang teknologi. Dari tahun ke tahun, perkembangan teknologi di berbagai penjuru dunia semakin berkembang ke arah yang lebih baik dan tentunya bermanfaat bagi kehidupan manusia.

Banyak macam teknologi yang telah dikembangkan atau diciptakan oleh manusia untuk menunjang kehidupannya sehari-hari. Baik itu untuk masalah ekonomi, kesehatan, sosial, budaya, pendidikan, hukum, pemerintah, kesehatan, keamanan dan masih banyak lagi. Yang akan diangkat dalam penelitian saya ini adalah dalam bidang sosial atau kehidupan sehari-hari yang semua orang pasti menjalaninya.

Masalah yang akan diambil adalah dari yang dekat dengan kita dan masyarakat umum banyak menggunakannya atau lebih tepatnya mendiaminya dalam kebutuhan mereka sehari-hari. Dengan membuat suatu alat yang dapat memonitoring rumah atau lebih tepatnya alat-alat listrik yang berada di rumah menjadi otomatis atau menjadi alat yang mudah untuk digunakan atau di kontrol. Karena dengan kebutuhan manusia yang dari hari ke harinya membutuhkan keringanan dalam melakukan kegiatannya. Oleh karena itu, sebagai penulis bermaksud membuat "*Smart House Berbasis Web*", yang dapat mengontrol nyala matinya lampu di rumah dan membuat pengaman pada pintu rumah dengan

memberikan alarm yang akan menyala ketika ada panas manusia yang berada pada area sensor yang telah dipasang, membuat jemuran otomatis dengan sensor hujan yang terhubung dengan servo *continuous*, membuat kipas otomatis berdasarkan suhu yang diterima sensor suhu dan membuat pagar yang dapat di buka tutup dengan sistem yang semuanya itu dapat di monitoring atau di kontrol lewat suatu web yang telah di integerasikan dengan alat yang sudah di buat menggunakan *RaspberryPi 3B*.

### **1.2. Rumusan Masalah**

Berdasarkan latar belakang usul penelitian ini, maka dapat dirumuskan beberapa masalah yaitu:

1. Bagaimana cara merancang dan membangun *Smart House Berbasis Web*?
2. Bagaimana melakukan pemantauan dan pengontrolan *Smart House* melalui *Web*?

### **1.3. Batasan Masalah**

Adapun batasan-batasan yang dimiliki aplikasi *SmartHouse* berbasis WEB ini. Beberapa hal yang dibatasi bagi adalah sebagai berikut:

- a. Aplikasi ini menggunakan tools *RaspberryPi 3B*.
- b. Aplikasi ini hanya bisa berjalan dengan hardware yang sudah di sinkronisasi.
- c. Aplikasi ini menggunakan WEB sebagai ruang kontrol dan monitoring alat tersebut dan tidak berbasiskan desktop.
- d. Sistem ini dibangun menggunakan menggunakan bahasa pemrograman *PHP*, *CSS*, *HTML*, *JavaScript*, *Python* , dan *database MySQL*.

e. Sistem ini dikhususkan untuk kenyamanan pengguna rumah dengan ketentuan alat yang digunakan adalah:

- Sensor PIR *Motion* dan Sirine sebagai *alarm* pintu rumah.
- Sensor Hujan Dan Motor Stepper sebagai jemuran otomatis.
- Lampu rumah yang bisa dinyalakan melalui sistem.
- Sensor Suhu dan Kipas Angin yang saling terhubung menjadi kipas otomatis berdasarkan ketentuan suhu ruangan.
- Servo 180 yang dipasang di pagar rumah agar dapat membuka pagar melalui sistem.
- Lampu balkon otomatis berdasarkan cahaya yang ditangkap oleh sensor cahaya.

f. Pengguna sistem hanya *end user*.

#### **1.4. Tujuan dan Manfaat**

Adapun tujuan yang akan dicapai dari penelitian ini, adalah:

1. Memberikan gambaran tentang *Smart House Berbasis Web*.
2. Merancang dan membangun *Smart House Berbasis Web*.
3. Memberikan kemudahan dan keamanan dalam pemakaian hunian rumah.

Adapun manfaat yang akan dicapai dari penelitian ini adalah :

##### **1. Bagi Pemilik Rumah**

- Mengefektifkan penggunaan perangkat yang terhubung dengan sistem oleh pengguna.
- Memberikan keamanan lebih terhadap pengguna dalam menempati rumah.

## 2. Bagi Mahasiswa

- Menerapkan ilmu yang didapat selama perkuliahan.
- Menambah pengalaman dalam membangun suatu sistem informasi.

### 1.5. Sistematika Penulisan

Sistematika penulisan skripsi ini ditulis untuk memudahkan pembaca dalam memahami isi dari skripsi. Berikut merupakan isi dari skripsi penelitian:

#### **I. PENDAHULUAN**

Bab ini membahas latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat, dan sistematika penulisan.

#### **II. TINJAUAN PUSTAKA**

Bab ini berisi teori-teori dasar yang berkaitan dengan penelitian dan laporan tugas akhir.

#### **III. METODE PENELITIAN**

Bab ini membahas metode yang akan digunakan saat penelitian, serta membahas hal-hal yang berkaitan dengan penelitian, yaitu waktu dan tempat penelitian, data dan alat penelitian, dan jadwal penelitian.

#### **IV. HASIL DAN PEMBAHASAN**

Bab ini membahas tentang perancangan sistem informasi, perancangan basisdata, perancangan tampilan, implementasi perancangan dan testing sistem.

#### **V. KESIMPULAN DAN SARAN**

Bab ini menjelaskan tentang kesimpulan dari hasil penelitian dan saran untuk pengembangan lebih lanjut.

## **2. TINJAUAN PUSTAKA**

### **2.1. *Smart House***

Smart home System adalah sebuah sistem berbantuan komputer yang akan memberikan segala kenyamanan, keselamatan, keamanan dan penghematan energi, yang berlangsung secara otomatis dan terprogram melalui komputer, pada gedung atau rumah tinggal anda.

Dapat digunakan untuk mengendalikan hampir semua perlengkapan dan peralatan di rumah Anda, mulai dari pengaturan tata lampu hingga ke berbagai alat-alat rumah tangga, yang perintahnya dapat dilakukan dengan menggunakan suara, sinar merah infra, atau melalui kendali jarak jauh (remote). Penerapan sistem ini memungkinkan Anda untuk mengatur suhu ruangan melalui termostat pada sistem pemanas atau penyejuk hawa, sehingga memberikan suasana " adanya kehidupan " meski sebenarnya Anda dan seisi rumah sedang tidak ada di tempat. (sumber: <http://subari.blogspot.co.id/2008/03/smart-home-sistem-pintar-di-rumah.html>).

### **2.2. *Raspberry Pi***

*Raspberry Pi* adalah suatu *mikroprosesor* yang memiliki ukuran seperti kartu kredit. *Raspberry Pi* ini dapat melakukan banyak hal seperti pada sebuah komputer dekstop meskipun ukurannya sangat kecil.



**Gambar 2.1 Raspberry Pi Model 3B**

(Sumber : alphr.com, 2016)

### **2.2.1. Model Raspberry Pi**

Model model dari *rasberry pi* yaitu :

#### **1) Raspberry Pi A+**

*Raspberry Pi A+* adalah versi dari *Raspberry Pi* yang rendah *spec* dan harga. versi ini hanya memiliki satu *Port USB*, konsumsi daya yang rendah, tidak ada *port Ethernet* dan hanya *256mb ram*.

Versi dari Pi lebih cocok untuk proyek-proyek yang tidak memerlukan sejumlah besar *power* untuk pemrosesan, anda dapat menggunakannya untuk proyek-proyek seperti robotika, pesawat *remote control* / mobil dan *project sistem embeded*.

## 2) *Raspberry Pi B dan B+*

*Raspberry Pi B+* dan *B* adalah versi sebelumnya dari *raspi* yang kini telah digantikan oleh *Raspberry Pi 2*. *Versi B +* memiliki satu *CPU core*, 4 *Port USB*, slot kartu *micro SD* dan konsumsi daya yang rendah. Hal ini meningkatkan pada model sebelumnya *B* yang hanya memiliki 2 *Port USB*, konsumsi daya yang lebih tinggi, ukuran *SD Card Slot* dan beberapa hal lainnya.

## 3) *Raspberry Pi 2*

*Raspberry Pi 2* adalah versi terbaru dari *Pi* dan versi tercepat dari *Pi* saat artikel ini ditulis. *Raspberry Pi 2* dan *versi B+* adalah versi paling populer yang dapat anda temukan karena kekuatan pemrosesan dan jumlah *port* yang bisa anda dapatkan. *Raspberry Pi 2* adalah penggantian *B +* dan memiliki fitur *900 MHz quad core CPU dan 1 GB ram*. Sisa dari spesifikasi tetap sama seperti apa yang akan Anda temukan di model sebelumnya yaitu *Raspberry Pi B+*.

## 4) *Raspberry Pi 3*

*Raspberry Pi 3* merupakan versi terbaru dari *rasberry Pi2*. Pada versi ini *rasberry* menambahkan jumlah *Pin GPIO* menjadi 40 dan menambahkan modul *WIFI*.

### 2.2.2. **Komponen Raspberry Pi 3**

#### 1) *Port USB*

*Raspberry Pi 3* memiliki 4 buah *Port USB*, tidak seperti pendahulunya yang hanya memiliki 2 buah *Port USB*. Bertambahnya jumlah *Port USB* ini

tentu membantu pengguna *Raspberry Pi* dalam berbagai hal misalnya ada slot tambahan sehingga tidak hanya cukup untuk memasukkan *keyboard* dan *mouse USB*, namun juga bisa ditambah kamera webcam atau penyimpanan *USB* seperti *flashdisk* maupun *hardisk*.

## 2) *Port HDMI*

*Port HDMI* ini digunakan untuk menghubungkan *Raspberry Pi* ke monitor. *Raspberry Pi* merupakan mikroprosesor yang berarti dapat digunakan seperti layaknya komputer biasa, oleh karena itu untuk menggunakan *Raspberry Pi* diperlukan sebuah monitor yang nantinya dihubungkan melalui *port HDMI* ini. Sebenarnya *Raspberry Pi* dapat digunakan secara *remote* dengan *SSH (Secure Shell)* ataupun *Virtual Network Computing (VNC)*, namun untuk konfigurasi awal tetap diperlukan adanya monitor.

## 3) *Port Audio jack 3,5*

*Audio jack* ini digunakan untuk menghubungkan *Raspberry Pi* dengan speaker. Seperti halnya komputer, *Raspberry Pi* juga dapat memproses suara. *Audio jack* ini digunakan untuk mengeluarkan suara yang diproses ke speaker.

## 4) *Port RJ 45*

*Port ethernet* tersedia pada *Raspberry Pi* sebagai media untuk menyambungkan *Raspberry Pi* ke suatu jaringan baik itu lokal maupun internet.



#### 5) *Micro SD Slot*

*Raspberry Pi 3* dilengkapi dengan *slot micro sd*, berbeda dengan pendahulunya yang dilengkapi dengan *slot sd card* yang berukuran lebih besar. *Micro sd* yang dimasukkan kedalam *Raspberry Pi* haruslah sudah berisi *Operating System (OS)* sehingga *Raspberry Pi* dapat digunakan. *OS* yang direkomendasikan adalah Raspbian yang berasal dari pengembangan *OS Debian Linux*. Namun *OS* apa saja dapat digunakan oleh *Raspberry Pi* seperti *Windows*, dan berbagai macam *distro Linux* lainnya.

#### 6) *Modul WIFI*

Tidak hanya *bluetooth* dan *port ethernet*, *Raspberry Pi* juga dilengkapi *modul WiFi* sebagai tambahan penghubung dengan perangkat lain. Tidak seperti model pendahulunya yang tidak dilengkapi *WiFi*, *Raspberry Pi 3* ini dilengkapi *WiFi 802.11n* yang memudahkan penggunaannya untuk langsung terhubung ke internet.

#### 7) *Modul Bluetooth*

*Raspberry Pi* dilengkapi dengan *bluetooth 4,1*, Dengan adanya modul *bluetooth* pada *Raspberry Pi* tentu akan memudahkan pengguna dalam menghubungkan *Raspberry Pi* dengan perangkat lainnya.

#### 8) *CSI*

*CSI* ini digunakan untuk menghubungkan *modul* kamera dengan *Raspberry Pi*. Terdapat 2 *modul* kamera yang resmi dikeluarkan oleh pabrik *Raspberry Pi* yaitu *modul* kamera *Raspberry Pi* dan *modul* kamera *PiNoIR*. Perbedaan kedua *modul* tersebut terletak pada kepekaan kamera dengan

cahaya inframerah dimana *modul NoIR* tidak dapat menangkap cahaya inframerah sehingga cocok digunakan untuk penglihatan pada malam hari.

#### 9) *Display Serial Interface (DSI)*

Berbeda dengan *CSI*, *Display Serial Interface* ditujukan agar *Raspberry Pi* dapat terhubung dengan *modul interface* lain seperti *LCD*.

#### 10) *GPIO*

*GPIO* merupakan suatu interface agar *Raspberry Pi* dapat terhubung ke dunia fisik baik itu melalui sensor maupun *aktuator*. Pin ini merupakan pin digital yang didesain sebagai *general purpose* atau dapat digunakan sebagai apa saja. Ada 40 *pin GPIO* pada *Raspberry Pi 3*, tidak seperti pendahulunya yang hanya memiliki 26 pin. Pada *GPIO* ini terdapat beberapa jenis pin seperti *ground*, *power*, dan *pin I/O (input/output)*.

#### 11) *Power Suply*

*Raspberry Pi* memerlukan tegangan DC 5V dan 2A. Jika tegangan yang dimasukkan ke *Raspberry Pi* kurang dari yang ditentukan, *Raspberry Pi* tetap akan menyala hanya saja akan ada sedikit gangguan seperti sebagian *port* tidak dapat digunakan atau gangguan lainnya. Jika tegangan yang dimasukkan melebihi ketentuan, dapat terjadi korsleting pada *Raspberry Pi*. Hal ini tentu saja berbahaya, oleh karena itu dianjurkan untuk mengecek tegangan yang masuk sebelum mulai menggunakan *Raspberry Pi*.

### 2.2.3. Keuntungan menggunakan *Raspberry Pi*

Kelebihan *Raspberry* ini bisa digunakan untuk project-project misalnya :

### 1. NAS (Network Attached Storage)

Yaitu sebagai file server berbasis IP. Pemakai rumahan skrg bisa memiliki NAS yang bisa berjalan 24-jam sehari dengan konsumsi listrik sekitar 50rb/tahun.

### 2. Media Server

Jika ingin punya HD Media player ,alat yang bisa memainkan file-file music, gambar, dan video di dalam hardisk, *Raspberry Pi* ini bisa di jadikan Media player dengan menggunakan port HDMI ataupun RCA. Anda bisa menonton Streaming TV online, bisa nonton Youtube ataupun memutar film kesukaan anda.

### 3. Print Server

Print Server adalah Alat yang di hubungkan dengan Printer,menjadikan printer anda bisa di gunakan bersama-sama dalam sebuah jaringan LAN. *Raspberry Pi* juga bisa di jadikan print server ,dengan menghubungkan printer yang ingin di jadikan printer Jaringan.

### 4. Server Torent -Bittorent

*Raspberry Pi* juga bisa di jadikan Server untuk download Torent.

### 5. Download manager

*Raspberry Pi* bisa dijadikan sebagai Komputer yang mendownload file-file film yang bisa kita tinggal tanpa mengkhawatirkan konsumsi listrik.

## 6. Wifi Internet Radio Player

Raspberry bisa juga di gunakan untuk memainkan music yang di streaming dari Internet Radio, bisa mengatur volume, mengatur list lagu dari handphone (android, Blackberry).

## 7. Reporting Dashboard

Bayangkan di ruangan Manager, CEO anda, ada satu monitor dengan kemampuan Screen touch, dan dibelakang monitor di pasang Raspberry+ Usb Wifi Mini, Lalu anda install browser Chromium, dan setting setiap kali Raspberry menyala maka langsung otomatis membuka website internal anda yaitu berisi Dashboard Report misalnya.

## 8. Home Automation

Anda bisa membuat Home Automation untuk mengontrol lampu , penyiraman taman, kipas angin, AC dan lain nya, dan bisa di atur baik dari layar LCD maupun Handphone anda.

## 9. Server Hosting website

Raspberry juga bisa menjadi Server untuk hosting website anda, berbasis html, php dan mysql. anda bisa membuat website misalnya untuk company profile ataupun wordpres sebagai CMS.

### **2.3. Sistem Tertanam**

Sistem tertanam adalah sebuah sistem dan aplikasi yang mengandung sedikitnya sebuah central information processing unit (CPU) yang dapat diprogram –umumnya dalam bentuk microcontroller, microprocessor atau pun digital signal processor chip yang digunakan oleh individu yang seringkali tidak menyadari

keberadaannya. Sistem ini dapat pula didefinisikan sebagai peralatan elektronik berukuran relatif kecil dan berbasis komputer dalam bentuk chip. Komputer atau CPU sebagai ‘otak’ sistem inilah yang berperan sentral memberikan kemudahan dalam pengoperasian serta meningkatkan fungsi dinamis dan otomatis sistem.

Sistem tertanam biasanya digunakan sebagai komponen inti dari produk lain, dan umumnya berada di dalam atau tertanam pada komponen atau peralatan yang lebih besar. Sistem tertanam dirancang untuk tujuan khusus melakukan satu atau banyak tugas dalam komputasi yang real-time. (Sumber: <https://30riyadh.wordpress.com/2012/06/07/pengertian-sistem-tertanam/>).

## **2.4. Framework**

### **2.4.1. Pengertian Framework**

*Framework* adalah seperangkat pustaka dari sebuah perangkat lunak yang telah terintegrasi (seperti *class*, objek dan komponen-komponen) yang berkolaborasi untuk menyediakan desain arsitektur dari sebuah aplikasi yang dapat digunakan kembali untuk membangun aplikasi sejenis yang terkait (sommerville:2011).

*Framework* dibedakan menjadi beberapa jenis yaitu :

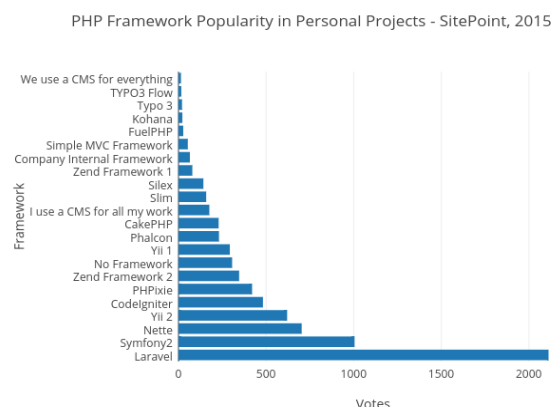
1. *System Infrastructur Framework* adalah *framework* yang mendukung pengembangan infrastruktur sistem seperti komunikasi, *user interface*, dan *compiler*.
2. *Middleware Integration Framework* adalah *framework* yang terdiri dari satu set standar dan kelas objek terkait yang mendukung komponen komunikasi

dan pertukaran informasi. Contoh: Microsoft NET. Framework dan Enterprise Java Beans (EJB).

3. *Enterprise Application Framework* adalah *framework* yang spesifik pada domain aplikasi seperti telekomunikasi atau sistem keuangan. *Framework* ini mendukung tujuan *End-Development User*.
4. *Web Application Framework* (WAF) adalah jenis *framework* yang terbaru saat ini. Arsitektur WAF yang biasanya didasarkan pada *Model-Views-Controller (MVC) pattern*. *Web application framework* biasanya menyertakan satu atau lebih kerangka kerja khusus yang mendukung fitur aplikasi secara spesifik (Sommerville, 2011).

#### 2.4.2. *Laravel Framework*

*Laravel framework* dibuat oleh Taylor Otwell tahun 2011. Meskipun tergolong baru, laravel telah mencuri perhatian banya programmer dunia. Dikarenakan para pengembang *framework* masih belum puas dengan *framework* yang telah ada.



**Gambar 2.2 Persentase Survei Kepopuleran laravel *Framework***

(Sumber : nicolasruslim.com, 2016)

Berikut ini akan dijelaskan kelebihan Laravel dibandingkan dengan framework PHP yang lain :

- |                      |                              |
|----------------------|------------------------------|
| a. Expressif         | e. Kompatibel dengan PHP 5.3 |
| b. Sempel            | keatas                       |
| c. Tersedia Composer | f. Dokumentasi yang lengkap  |
| d. Open Source       |                              |

Framework laravel memiliki beberapa fitur sebagai berikut :

1. Bundles yaitu sebuah fitur dengan system pengemasan modular dan berbagai bundle telah tersedia untuk di gunakan.
2. Eloquent ORM merupakan penerapan PHP lanjutan dari pola “active record” menyediakan metode internal untuk mengatasi kendala hubungan antara objek database. Pembangunan query Laravel Fluent didukung Eloquent.
3. Application Logic merupakan bagian dari aplikasi yang di kembangkan, baik menggunakan Controllers maupun sebagai bagian dari deklarasi Route. Sintaks yang digunakan untuk mendefinisikannya hampir sama dengan yang digunakan oleh framework Sinatra.
4. Reverse Routing, mendefinisikan hubungan antara Link dan Route, sehingga jika suatu saat ada perubahan pada route secara otomatis akan tersambung dengan link yang relevan. Ketika link yang dibuat dengan menggunakan nama-nama dari Route yang ada, secara otomatis Laravel akan membuat URI yang sesuai.

5. Restful Controllers, memberikan sebuah pilihan untuk memisahkan logika dalam melayani HTTP GET dan permintaan POST.
6. Class Auto Loading, menyediakan otomatis loading untuk class-class PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah loading yang tidak perlu.
7. View Composers adalah kode unit logical yang dapat dijalankan ketika sebuah View di load.
8. IoC Composers memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip control pembalik, dengan pilihan contoh dan referensi dari objek baru sebagai Singletons.
9. Migrations menyediakan versi sistem control untuk skema database, sehingga memungkinkan untuk menghubungkan perubahan adalah basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak database, mempermudah dalam penempatan dan memperbarui aplikasi.
10. Unit Testing mempunyai peran penting dalam framework Laravel, dimana unit testing ini mempunyai banyak tes untuk mendeteksi dan mencegah regresi. Unit testing dapat dijalankan melalui fitur “artisan command line”.
11. Automatic pagination menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke laravel.



## 2.5. Aplikasi berbasis Web

### 2.5.1. Pengertian Aplikasi

Menurut *Kamus Besar Bahasa Indonesia*, aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Dari pengertian-pengertian tersebut dapat disimpulkan bahwa aplikasi merupakan program siap pakai yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya dengan tujuan untuk membantu mencapai tujuan pengguna.

Sedangkan menurut Jogiyanto HM(2005) Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output* (Jogiyanto HM, 2005).

### 2.5.2. Pengertian Web

*Web* adalah kumpulan komputer pribadi, *web browser*, koneksi ke ISP, komputer *server*, *router*, dan *switch* yang digunakan untuk mengalirkan informasi dan menjadi wahana pertama berbagai pihak terkait (Budi Sutedjo,2007).

*Web* dibagi menjadi beberapa jenis, yaitu:

- a. *Web Search Engine*, adalah *web* yang memiliki kemampuan untuk melakukan pencarian dokumen berdasarkan kata kunci tertentu.
- b. *Web Portal*, adalah *web* yang berisi kumpulan *link*, *search engine*, dan informasi.
- c. *Web Perusahaan*, adalah *web* yang mendeskripsikan suatu perusahaan, layanan, fasilitas, dan segala sesuatu tentang perusahaan.

### 2.5.3. Pengertian Aplikasi berbasis Web

Menurut Abdul Kadir (2014), Berdasarkan pemrosesan, aplikasi web dibagi menjadi dua, yaitu:

#### 1) Pemrosesan disisi server

Pemrosesan diserver berarti bahwa kode yang menyusun aplikasi web di proses disisi server. Pemrosesan aplikasi disisi server seringkali melibatkan PHP, JSP, atau ASP. Kode yang dibuat dengan bahasa diatas akan di eksekusi di server dan hasilnya diberikan ke klien untuk diproses oleh klien. Kode seperti itu diperlukan antaralain untuk memungkinkan pengelolaan data secara dinamis atau untuk merahasiakan hal tertentu , misalnya *password*.

#### 2) Pemrosesan disisi klien

Pemrosesan disisi klien berarti kode yang menyusun aplikasi web diproses disisi klien / dilakukan oleh browser. Kode yang diterima oleh klien adalah kode yang diproses di sisi klien. Kode tersebut berbentuk skrip yang akan di proses oleh browser, yang dapat berisi HTML, JavaScript, dan CSS. Awalnya hanya HTML yang digunakan untuk membentuk halaman web. Namun dalam perkembangan selanjutnya , aplikasi web dibedakan menjadi tiga lapisan, yaitu lapisan konten, lapisan presentasi dan lapisan perilaku.

Aplikasi berbasis web merupakan aplikasi yang berjalan menggunakan web browser sehingga tidak perlu menginstalnya terlebih dahulu. Banyak keuntungan yang didapatkan jika menggunakan aplikasi berbasis web, antara lain pengguna dapat menggunakannya dimanapun dan kapanpun, dapat dibuka dengan perangkat

apapun baik dengan komputer, laptop, maupun handphone yang terkoneksi dengan internet.

## **2.6. Sensor**

Sensor adalah sesuatu yang digunakan untuk mendeteksi adanya perubahan lingkungan fisik atau kimia. Variabel keluaran dari sensor yang diubah menjadi besaran listrik disebut Transduser. Pada saat ini, sensor tersebut telah dibuat dengan ukuran sangat kecil dengan orde nanometer. Ukuran yang sangat kecil ini sangat memudahkan pemakaian dan menghemat energi.

Sensor yang digunakan pada penelitian ini sebagai berikut:

### **2.6.1. Sensor PIR**



**Gambar 2.3. Sensor PIR**

(Sumber : <http://khourumuslimah.blogspot.co.id/2015/06/pengertian-dan-cara-kerja-sensor-pir.html>)

Sensor PIR (Passive Infra Red) adalah sensor yang digunakan untuk mendeteksi adanya pancaran sinar infra merah. Sensor PIR bersifat pasif, artinya sensor ini tidak memancarkan sinar infra merah tetapi hanya menerima radiasi sinar infra merah dari luar. Sensor suhu dan kelembaban (DHT11)

Sensor ini biasanya digunakan dalam perancangan detektor gerakan berbasis PIR. Karena semua benda memancarkan energi radiasi, sebuah gerakan akan terdeteksi ketika sumber infra merah dengan suhu tertentu (misal: manusia) melewati sumber infra merah yang lain dengan suhu yang berbeda (misal: dinding), maka sensor akan membandingkan pancaran infra merah yang diterima setiap satuan waktu, sehingga jika ada pergerakan maka akan terjadi perubahan pembacaan pada sensor.

Sensor PIR terdiri dari beberapa bagian yaitu :

1. Fresnel Lens

Lensa Fresnel pertama kali digunakan pada tahun 1980an. Digunakan sebagai lensa yang memfokuskan sinar pada lampu mercusuar. Penggunaan paling luas pada lensa Fresnel adalah pada lampu depan mobil, di mana mereka membiarkan berkas parallel secara kasar dari pemantul parabola dibentuk untuk memenuhi persyaratan pola sorotan utama. Namun kini, lensa Fresnel pada mobil telah ditiadakan diganti dengan lensa plain polikarbonat. Lensa Fresnel juga berguna dalam pembuatan film, tidak hanya karena kemampuannya untuk memfokuskan sinar terang, tetapi juga karena intensitas cahaya yang relative konstan diseluruh lebar berkas cahaya.

2. IR Filter

IR Filter dimodul sensor PIR ini mampu menyaring panjang gelombang sinar infrared pasif antara 8 sampai 14 mikrometer, sehingga panjang gelombang yang dihasilkan dari tubuh manusia yang berkisar antara 9

sampai 10 mikrometer ini saja yang dapat dideteksi oleh sensor. Sehingga Sensor PIR hanya bereaksi pada tubuh manusia saja.

### 3. Pyroelectric Sensor

Seperti tubuh manusia yang memiliki suhu tubuh kira-kira 32 derajat celcius, yang merupakan suhu panas yang khas yang terdapat pada lingkungan. Pancaran sinar inframerah inilah yang kemudian ditangkap oleh Pyroelectric sensor yang merupakan inti dari sensor PIR ini sehingga menyebabkan Pyroelectric sensor yang terdiri dari galium nitrida, caesium nitrat dan litium tantalate menghasilkan arus listrik. Mengapa bisa menghasilkan arus listrik? Karena pancaran sinar inframerah pasif ini membawa energi panas. Material pyroelectric bereaksi menghasilkan arus listrik karena adanya energi panas yang dibawa oleh infrared pasif tersebut. Prosesnya hampir sama seperti arus listrik yang terbentuk ketika sinar matahari mengenai solar cell.

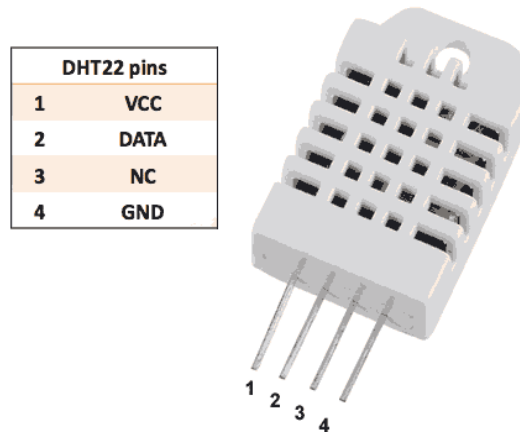
### 4. Amplifier

Sebuah sirkuit amplifier yang ada menguatkan arus yang masuk pada material pyroelectric.

### 5. Komparator

Setelah dikuatkan oleh amplifier kemudian arus dibandingkan oleh komparator sehingga menghasilkan output.

### 2.6.2. Sensor Suhu DHT 22

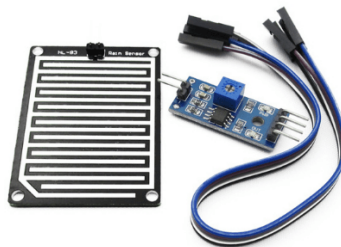


**Gambar 2.4. sensor suhu DHT22**

(Sumber : <http://www.labelektronika.com/2016/09/dht22-sensor-suhu-dan-kelembaban-arduino.html>)

DHT22 (AM2302) *Digital Capacitive Relative Humidity & Temperature Sensor Module* adalah sensor yang dapat mengukur suhu dan kelembaban udara di sekitarnya. Sensor ini lebih akurat dan presisi dalam hal pengukuran di banding DHT11, kekurangan DHT22 ,harga lebih mahal di banding DHT11

### 2.6.3. Sensor hujan



**Gambar 2.5. Sensor Hujan**

(Sumber : <http://www.ngarep.net/tutorial-arduino-mengakses-sensor-hujan/>)

Sensor hujan adalah sensor yang difungsikan untuk mendeteksi ada tidaknya kondisi rintik hujan, yang dimana dapat dimanfaatkan dalam berbagai aplikasi mulai dari yang sederhana hingga aplikasi yang kompleks.

#### **2.6.4. Sensor Cahaya**

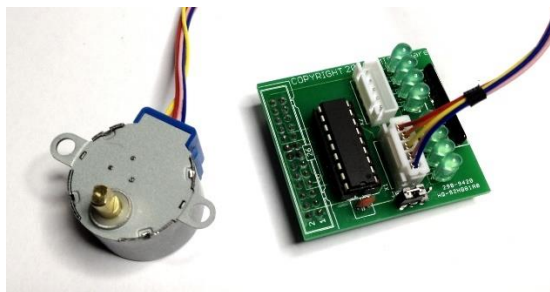


**Gambar 2.6. Sensor Cahaya**

(Sumber: [https://id.wikipedia.org/wiki/Sensor\\_cahaya](https://id.wikipedia.org/wiki/Sensor_cahaya))

Sensor cahaya adalah sebuah alat yang digunakan untuk mengubah besaran cahaya menjadi besaran listrik. Prinsip kerja dari alat ini adalah mengubah energi dari foton menjadi elektron. Idealnya satu foton dapat membangkitkan satu elektron. Sensor cahaya sangat luas penggunaannya, salah satu yang paling populer adalah kamera digital.

#### **2.7. Motor Stepper**



**Gambar 2.7. Motor Stepper**

(Sumber: <https://www.raspberrypi.org/forums/viewtopic.php?f=81&t=27928>)

Motor stepper adalah salah satu jenis motor dc yang dikendalikan dengan pulsa-pulsa digital. Prinsip kerja motor stepper adalah bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit dimana motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor stepper tersebut.

## 2.8. Servo 180



**Gambar 2.8. Servo 180**

(Sumber :Sumber: <http://trikueni-desain-sistem.blogspot.co.id/2014/03/Pengertian-Motor-Servo.html>)

Motor servo standard (servo rotation  $180^{\circ}$  ) adalah jenis yang paling umum dari motor servo, dimana putaran poros outputnya terbatas hanya  $90^{\circ}$  kearah kanan dan  $90^{\circ}$  kearah kiri. Dengan kata lain total putarannya hanya setengah lingkaran atau  $180^{\circ}$  .

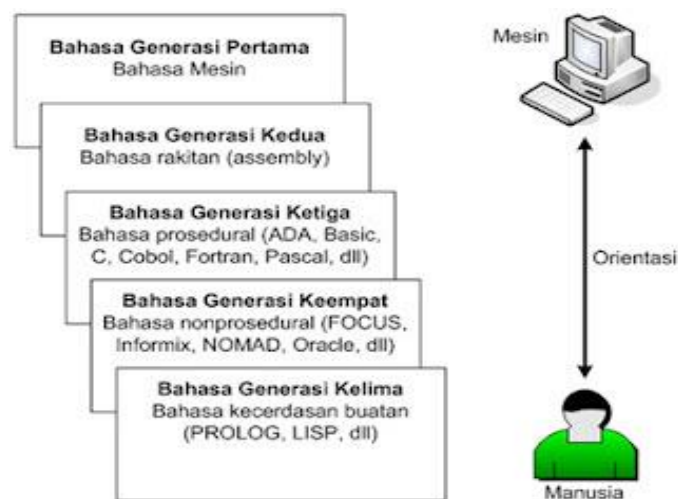
## 2.9. Bahasa Pemrograman

Program adalah sekumpulan intruksi yang digunakan untuk mengatur komputer agar melakukan suatu tindakan tertentu (Kadir:2012). Komputer memiliki bahasanya sendiri , promrogram harus menuliskan program dalam bahasa yang bisa dimengerti oleh komputer. Bahasa inilah yang disebut dengan bahasa



pemrograman atau bahasa komputer. Bahasa pemrograman terdiri dari atas sekumpulan instruksi yang ditujukan agar orang bisa menuangkan perintah yang nantinya dapat dijalankan oleh komputer.

Bahasa bahasa yang digunakan untuk mengatur perangkat keras komputer telah mengalami evolusi beberapa kali. Sampai sejauh ini, dikenal adanya lima generasi bahasa pemrograman . kecenderungan yang terjadi bahasa-bahasa pemrograman bergeser dari pendekatan yang berorientasi kepada mesin menuju ke pendekatan yang berorientasi pada manusia.



**Gambar 2.9. Generasi dari Bahasa Pemrograman**

(Sumber : jarot.hmi.web.id, 2016)

### **2.9.1. PHP (*Php Hypertext Preprocessor*)**

PHP adalah salah satu bahasa pemrograman script yang dirancang untuk membangun aplikasi web. Setiap permintaan yang dilakukan oleh user melalui web browser akan direspon oleh aplikasi web dan hasilnya akan dikembalikan lagi

kehadapan user. dengan Aplikasi web, halaman yang tampil di layar web browser dapat bersifat dinamis, tergantung dari nilai data atau parameter yang dikirimkan oleh user ke web server.

Program PHP memiliki kerangka yang sangat sederhana. Kita dapat menggunakannya atau menyisipkannya secara langsung didalam kode HTML. Kode PHP ditandai dengan tanda (tag) `<?php` dan `?>` Tanda `<?php` digunakan untuk mengawali kode PHP, sedangkan tanda `?>` untuk mengakhiri kode PHP (Raharjo:2011).

### **2.9.2. HTML (*Hyper Text Markup Language*)**

Menurut Adhi Prasetyo (2012), HTML (*Hyper Text Markup Language*) adalah sebuah bahasa pemrograman atau *file* teks yang berisi *tag-tag markup* yang berguna untuk memberitahukan *browser* bagaimana harus menampilkan sebuah halaman *web*. Sebuah *file* HTML harus memiliki ekstensi `htm` atau `html`. HTML merupakan bahasa standar yang digunakan oleh browser internet untuk membuat halaman dan dokumen pada sebuah *web* yang kemudian dapat diakses dan dibaca layaknya sebuah artikel. HTML juga dapat digunakan sebagai *link* antara *file-file* dalam situs atau dalam komputer dengan menggunakan `localhost`, atau link yang menghubungkan antar situs dalam dunia internet.

HTML (*Hypertext Markup Language*) merupakan standar bahasa yang digunakan untuk menampilkan dokumen *web*, yang bisa dilakukan dengan HTML yaitu:

- a. Mengontrol tampilan dari *web page* dan *content*.
- b. Mempublikasikan dokumen secara online sehingga bisa diakses dari seluruh dunia.
- c. Membuat *online form* yang bisa digunakan untuk menangani pendaftaran, transaksi secara online.
- d. Menambahkan objek-objek seperti *image*, *audio*, *video* dan juga *java applet* dalam dokumen HTML.

### **2.9.3. CSS (cascading style sheet)**

Menurut Jayan (2010), CSS berguna untuk mengatur tampilan dokumen html, contohnya seperti pengaturan jarak antara baris , teks, warna dan format border bahkan penampilan gambar. CSS dikembangkan oleh W3C, organisasi yang mengembangkan teknologi internet. Tujuannya untuk mempermudah penataan halaman web. CSS berupa kumpulan script yang tujuannya bukan untuk menggantikan HTML, melainkan sebagai pelengkap agar dokumen HTML bisa tampil lebih cantik dan dinamis.

### **2.9.4. Java Script**

Java SScript adalah bahasa pemrograman yang digunakan untuk membuat Web lebih dinamis dan interaktif. Javascript terintegrasi langsung oleh dengan HTML. Berbeda dengan PHP yang merupakan pemrograman server side, javascript merupakan bahasa pemrograman client side.kode PHP tidak akan ditemukan dalam kode sumber manapun, tetapi sangat besar kemungkinan ditemukannya kode javascript seperti layaknya kode HTML. Hal ini dikarenakan javascript dijalankan di client (browser), sedangkan PHP diajalankan oleh server. Java script akan

dikirimkan ke browser bersamaan dengan kode HTML dan dijalankan oleh browser (Adhi Prasetyo:2012).

Dasar-dasar javascript yaitu :

1) Pernyataan (statement)

Pernyataan adalah satu perintah utuh untuk melaksanakan suatu tindakan. Dalam bahasa alami pernyataan dianalogikan sebagai sebuah kalimat. Umumnya, sebuah pernyataan ditulis dalam satu baris. Namun seperti halnya sebuah kalimat, sebuah pernyataan boleh ditulis di beberapa baris.

a) Pernyataan majemuk

Pernyataan majemuk adalah kumpulan pernyataan yang diperlakukan sebagai satu pernyataan. Pernyataan ini ditandai dengan “{ }”. Tanda tersebut membuat yang ada di dalamnya dianggap sebagai satu pernyataan walaupun kenyataannya terdiri dari beberapa pernyataan. Pernyataan majemuk sering dijumpai pada pernyataan *if*, *while* dan *for*.

b) Struktur pernyataan

Berdasarkan strukturnya dibagi menjadi tiga jenis, yaitu struktur sekuensial, struktur keputusan dan struktur perulangan.

2) Fungsi bawaan

Javascript menyediakan banyak fungsi bawaan untuk mendukung kemudahan dalam membuat program. Sebagai contoh, *alert ()* dapat

digunakan untuk menampilkan pesan. Dengan adanya fungsi-fungsi tersebut pembuatan program menjadi lebih sederhana.

### 3) Tipe data

Tipe data adalah fondasi penting yang harus diketahui. Tipe data menyatakan cara memperlakukan data di dalam memori komputer. Javascript memiliki tiga tipe data sebagai berikut:

- a) Bilangan : bilangan bulat, bilangan desimal, bilangan oktal, dan heksadesimal
- b) String : berarti deretan karakter. Bisa ditulis dengan awalan dan akhiran petik tunggal atau petik ganda
- c) Boolean : tipe yang nilainya hanya berupa nilai benar [true] atau salah [false]. Tipe seperti ini sangat berguna untuk menangani struktur seleksi maupun struktur perulangan.

### 4) Variabel

Variabel dapat menampung suatu nilai dan nilainya dapat diubah kapan saja ketika program dieksekusi.

### 5) Komentar

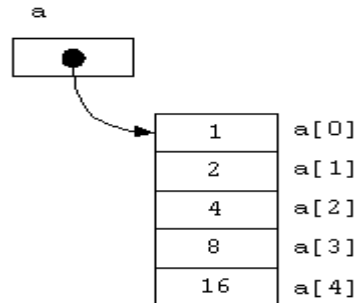
Komentar biasa disertakan di program untuk memberikan penjelasan tentang suatu perintah kepada pembaca program. Bagi pengguna aplikasi web, komentar tidak terlihat. Komentar dapat disisipkan dengan menyertakan tanda `//`. Setelah tanda tersebut hingga baris akhir akan dianggap sebagai komentar.

## 6) Kotak dialog

Javascript menyediakan tiga kotak dialog yang dapat digunakan untuk menampilkan pesan sesaat, meminta data dari pemakai dan konfirmasi pemakai. Kotak-kotak yang dimaksud dapat diperoleh melalui *alert ()*, *prompt ()*, dan *confirm ()*.

## 7) Array

Array atau larik adalah contoh wadah yang memungkinkan sejumlah data bisa disimpan dengan menggunakan satu nama. Contohnya dikehendaki untuk mencatat lima nilai, jika menggunakan variabel kita membutuhkan lima variabel. Jika menggunakan array kita dapat menggunakan dengan nama array nilai a.



**Gambar 2.10. Contoh Array**

(Sumber : javakomputer, 2016)

## 8) Objek

Suatu objek mengandung data dan fungsi. Fungsi biasanya terkait dengan data. Data yang terkandung di objek biasanya disebut properti, sedangkan fungsi yang melekat di objek dinamakan metode.

### 2.9.5. JQuery

Menurut Abdul kadir(2014) JQuery adalah pustaka yang dibangun dengan menggunakan JavaScript. Tujuan dari JQuery adalah agar pembuatan program JavaScript bisa dilakukan dengan cara yang lebih ringkas. Fitur-fitur yang ditawarkan oleh jQuery adalah sebagai berikut :

- a. Mempermudah akses dan manipulasi ke bagian page tertentu dengan sebuah selector yang robust dan efisien untuk mengambil bagian tertentu pada dokumen yang selanjutnya bisa dimanipulasi.
- b. Mempermudah perubahan tampilan dokumen, karena jQuery dapat merubah tampilan CSS dengan mudah.
- c. Dapat merespon interaksi user dengan webpage dengan cara yang elegan untuk memasukkan sebuah event ke dalam salah satu bagian dari webpage.
- d. Dapat menambah animasi pada webpage. JQuery dapat ditambahkan ke halaman web dengan menambahkan baris kode diantara tag <head>.

### 2.9.6. Python

Python merupakan suatu bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido Van Rossum. Python dapat berjalan di sistem operasi manapun dengan menggunakan *interpreter*. Python dibuat untuk meningkatkan tingkat keterbacaan suatu *source code*. Dalam pengaplikasiannya, Python sering digunakan oleh para ilmuwan yang membutuhkan suatu *scripting language* untuk memudahkan pekerjaan mereka. Selain ilmuwan, Python juga populer kalangan pelajar yang baru mempelajari bahasa pemrograman. Python mendukung semua konsep pemrograman baik itu prosedural maupun *object oriented*. Python berjalan

diatas *interpreter* yang kemudian menerjemahkan bahasa Python ke bahasa yang lebih rendah. Hal ini menyebabkan kecepatan kompilasi Python kalah cepat dengan bahasa tingkat rendah seperti C ataupun C++.

## 2.10. Basis Data

Menurut Abdul Kadir (2014) Basis data adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk meperoleh informasi. Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas.untuk mengelola basis data diperlukan perangkat lunak yang disebut Database Management System.

Komponen-komponen basis data yang mendukung lingkungan DBMS terdiri atas:

### a. Perangkat Keras

Perangkat keras digunakan untuk menjalankan DBMS beserta aplikasi-aplikasinya. Perangkat keras berupa komputer dan periperal pendukungnya. Komputer dapat berupa PC, mini komputer, *mainframe*, dll.

### b. Perangkat Lunak

Perangkat Lunak yang mencakup DBMS itu sendiri, program aplikasi serta perangkat lunak pendukung untuk komputer dan jaringan. Program aplikasi dapat dibangun dengan menggunakan bahasa pemrograman seperti C++, pascal, Delphi, atau Visual Basic.



### c. Data

Komponen terpenting dalam DBMS adalah data, Karena data inilah pemakai dapat memperoleh informasi yang sesuai dengan kebutuhan masing-masing.

### d. Prosedur

Prosedur adalah petunjuk tertulis yang berisi cara merancang hingga menggunakan basis data. Beberapa hal yang dimasukan kedalam prosedur antara lain :

- Cara masuk ke DBMS.
- Cara memakai fasilitas tertentu dalam DBMS.
- Cara mengaktifkan dan menghentikan DBMS.
- Cara membuat cadangan basis data dan mengembalikannya.

### e. Manusia

Manusia dikelompokkan menjadi tiga, yaitu :

#### 1. Pemakai Akhir (*end user*)

Pemakai akhir adalah orang yang menggunakan / mengoperasikan program aplikasi yang telah dibuat oleh pemrogram aplikasi. Pemakai ini tidak berhubungan secara langsung dengan DBMS.

#### 2. Pemrogram Aplikasi

Pemrogram aplikasi adalah orang yang membuat program aplikasi yang melibatkan basis data. Pemrogram aplikasi membuat program aplikasi berdasarkan kebutuhan pemakai.

### 3. Administrator Basis Data

DBA adalah orang yang bertanggung jawab terhadap manajemen basis data. Secara lebih detail, tugas DBA sebagai berikut :

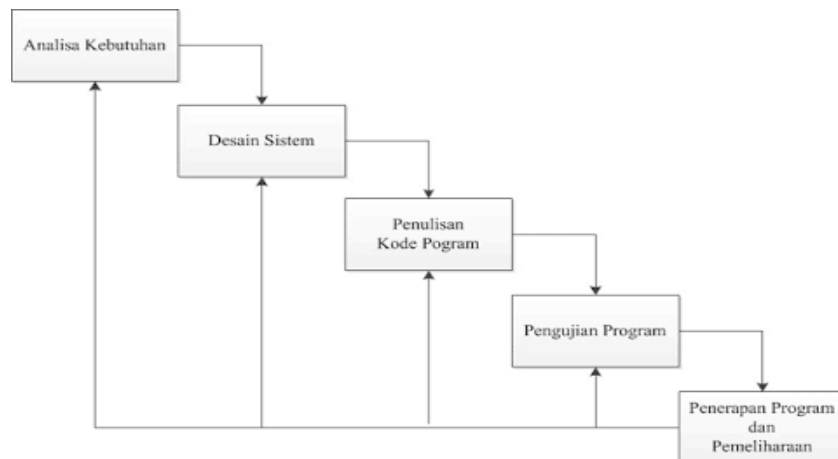
- Mendefinisikan basis data
- Mendefinisikan struktur dan metode akses penyimpanan
- Menentukan keamanan basis data
- Melakukan pemeliharaan basis data secara rutin

#### 2.10.1. MySQL

Menurut Adhi Prasetyo (2012), MySQL merupakan perangkat lunak *Database Management System* (DBMS) yang tergolong dalam *database server* dan bersifat *open source* yang mempunyai struktur sistem manajemen basis data relasional (RDBMS-*Relational Database Management System*). *Open Source* menyatakan bahwa software ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara mengunduh di Internet secara gratis.

## 2.11. Metode Waterfall

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak yang dilakukan secara berurutan atau secara linear.



**Gambar 2.11. Metode Waterfall**

(Sumber : Kadir , 2013)

Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut : Analisis, Desain, Penulisan, Pengujian dan Penerapan serta Pemeliharaan (Kadir, 2013)

### a. Analisis Kebutuhan

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Sistem analis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam

pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analis untuk menerjemahkan ke dalam bahasa pemrograman.

b. Desain Sistem

Tahapan dimana dilakukan penuangan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti diagram alir data (*data flow diagram*), diagram hubungan entitas (*entity relationship diagram*) serta struktur dan bahasan data.

c. Penulisan Kode Program

Penulisan kode program atau *coding* merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem tersebut dan kemudian bisa diperbaiki.

d. Pengujian Program

Tahapan akhir dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna.

e. Penerapan Program dan Pemeliharaan

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periperal atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

## 2.12. UML (*Unified Modeling Language*)

Menurut Whitten dan Bentley (2007: 371), *Unified Modelling Language (UML)* adalah sekumpulan aturan pemodelan yang digunakan untuk menspesifikasi dan menggambarkan suatu sistem software yang berkaitan dengan objek.

Diagram UML merupakan notasi pemodelan yang telah menjadi standar dalam bidang perancangan software. Diagram UML sama halnya seperti blueprint pada pembangunan rumah, dimana sekumpulan blueprint tersebut menyediakan sudut pandang gambaran yang berbeda-beda sesuai dengan kebutuhan pengelolanya. Diagram UML juga menyediakan gambaran dari masing-masing sudut pandang yang berbeda sesuai dengan tim pengembangnya. Standarisasi diagram UML telah diatur dan dibuat oleh Object Management Group (OMG).

Menurut Whitten dan Bently (2007: 381-382) versi diagram UML 2.0. memiliki 13 diagram, antara lain:

1. Use Case Diagram

Diagram yang menggambarkan interaksi antara sistem dengan sistem eksternal dan users.

## 2. Activity Diagram

Diagram yang menggambarkan urutan aliran aktivitas dari use case atau proses bisnis.

## 3. Class Diagram

Diagram yang menggambarkan objek-objek yang terdapat pada sistem dalam bentuk struktur.

## 4. Object Diagram

Diagram objek mirip dengan diagram class tetapi lebih menggambarkan objek yang terdapat dalam class.

## 5. State Machine Diagram

Diagram yang menggambarkan bagaimana suatu kejadian dapat mengubah status objek selama waktu hidupnya.

## 6. Sequence Diagram

Diagram yang menggambarkan bagaimana suatu objek berinteraksi melalui pesan dalam pelaksanaan use case atau operasi.

## 7. Communication Diagram

Dalam diagram UML versi 1.0. disebut diagram collaboration, menggambarkan interaksi dari objek melalui pesan.

## 8. Interaction Overview Diagram

Diagram yang mengkombinasikan fitur dari diagram activity dan diagram sequence untuk menunjukkan bagaimana objek berinteraksi pada setiap aktivitas dari use case.

#### 9. Timing Diagram

Diagram interaksi lainnya yang berfokus pada kendala waktu dalam perubahan status dari single objek atau sekumpulan objek.

#### 10. Component Diagram

Diagram yang menggambarkan pengaturan dari kode pemrograman yang dibagi menjadi komponen-komponen, dan bagaimana komponen tersebut berinteraksi.

#### 11. Deployment Diagram

Diagram yang menggambarkan konfigurasi dari komponen software dengan arsitektur fisik dari sistem hardware.

#### 12. Package Diagram

Diagram yang menggambarkan bagaimana class atau diagram UML lainnya membangun sebuah package yang terorganisir.

#### 13. Composite Structure Diagram

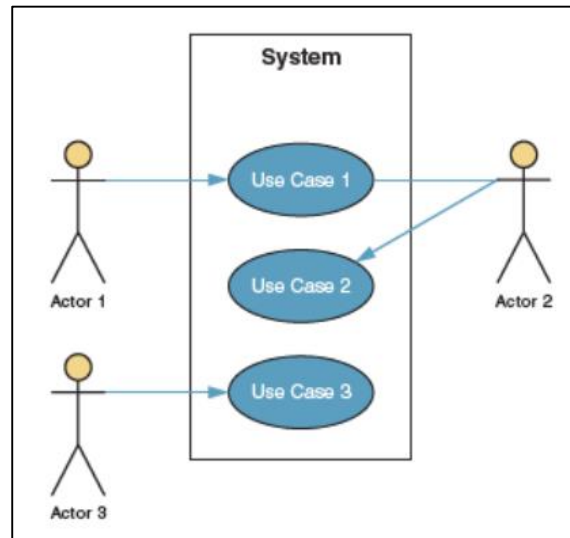
Diagram yang menguraikan struktur internal dari class, component, atau use case.

Meskipun UML memiliki begitu banyak diagram, tidak semuanya diperlukan dan harus digunakan dalam perancangan. Apabila sistem telah digambarkan dan dapat dipahami dengan baik menggunakan diagram yang ditentukan, maka tujuan dari UML telah tercapai.

##### **2.12.1. Use Case Diagram**

*Use case diagram* adalah diagram yang menggambarkan interaksi antara sistem dengan pengguna dan menggambarkan siapa yang akan menggunakan

sistem serta bagaimana pengguna berinteraksi dengan sistem (Whitten dan Bentley, 2007: 246-250).



**Gambar 2.12. Contoh Use Case Diagram**

(Sumber : Whitten & Bentley, 2007: 246)

*Use case diagram* memiliki komponen dasar yaitu *use case*, *actors* dan *relationships*. *Use case* merupakan urutan perilaku terkait langkah-langkah untuk menyelesaikan tugas bisnis tunggal. *Use case* menggambarkan fungsi sistem dengan cara dan terminologi yang dipahami oleh pengguna eksternal. *Use case* digambarkan dengan bentuk *ellipse* dengan nama use case yang dituliskan di atas, di bawah atau di dalam *ellipse*.

*Actors* yaitu pengguna eksternal yang berinteraksi dengan sistem untuk mencapai tujuan dari *use case*. Terdapat 4 (empat) tipe *actors*, yaitu:

1. *Primary business actor*: stakeholder yang paling diuntungkan dari terlaksananya *use case* dengan menerima sesuatu yang dapat diukur.

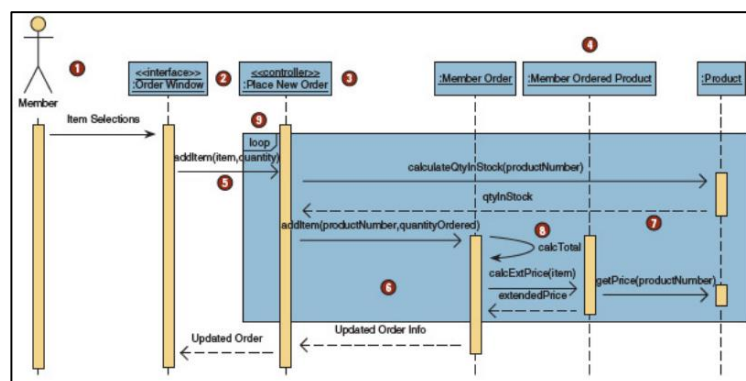


2. *Primary system actor*: stakeholder yang secara langsung berinteraksi dengan sistem untuk memulai kegiatan bisnis atau kegiatan sistem.
3. *External server actor*: stakeholder merespon permintaan dari *use case*.
4. *External receiver actor*: stakeholder yang bukan merupakan actor utama tetapi menerima sesuatu yang dapat bernilai dari *use case*.

Selain itu dalam *use case diagram* terdapat relationship yang digambarkan dengan garis yang menghubungkan antar *use case*. Arti dari *relationship* bergantung pada bagaimana *relationship* tersebut digambarkan. Terdapat beberapa *relationship* yang digunakan, yaitu asosiasi, *extends*, *uses/includes*, *depends on*, dan *Inheritance*.

### 2.12.2. Sequence Diagram

*Sequence diagram* adalah salah satu diagram UML yang menggambarkan model logika dari sebuah *use case* dengan menggambarkan interaksi pesan antar objek dalam suatu urutan waktu (Whitten dan Bentley, 2007: 659).



**Gambar 2.13. Contoh Sequence Diagram**

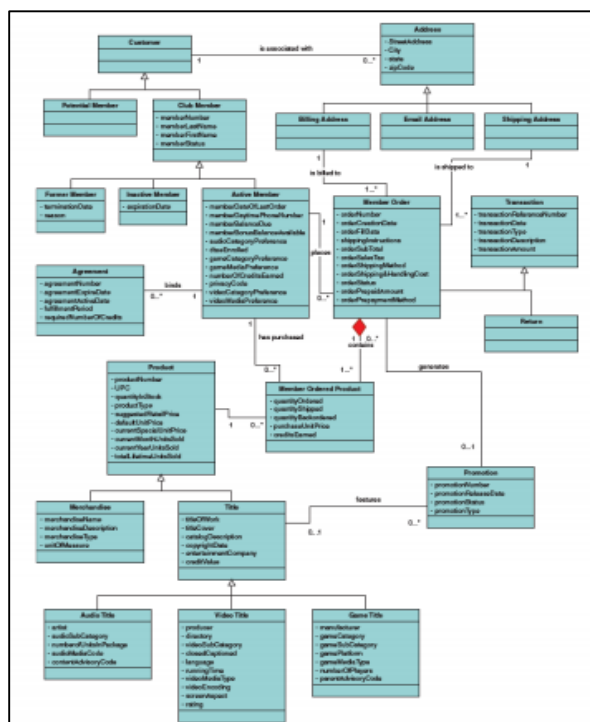
(Sumber : Whitten & Bentley, 2007: 659)

### 2.12.3. Activity Diagram

*Activity diagram* adalah suatu diagram yang digunakan untuk menggambarkan proses bisnis, langkah-langkah dari *use case* atau logika perilaku dari objek. Dalam *activity diagram*, lebih dari satu *use case* dapat digambarkan dengan satu *activity diagram*, tetapi satu *activity diagram* dapat digambarkan untuk setiap *use case* jika *use case* tersebut memiliki logika yang kompleks (Whitten dan Bentley, 2007: 390).

### 2.12.4. Class Diagram

*Class diagram* merupakan gambaran grafis dari struktur objek, yang menunjukkan kelas objek yang ada pada suatu sistem dan hubungan antara kelas-kelas objek tersebut (Whitten dan Bentley, 2007: 400).



**Gambar 2.14 Contoh Class Diagram**

(Sumber : Whitten & Bentley, 2007: 406)

Setiap kelas objek dibagi menjadi 3 (tiga) bagian, yaitu:

1. Nama kelas

Nama kelas harus unik atau dapat dibedakan antara kelas yang satu dengan kelas yang lain karena nama kelas merupakan identitas yang dimiliki oleh setiap kelas.

2. Atribut kelas

Atribut menunjukkan informasi yang dimiliki oleh suatu kelas atau informasi yang berhubungan dengan suatu kelas.

3. Operasi kelas

Operasi digunakan untuk menunjukkan fungsi yang dapat dilakukan oleh suatu kelas.

### **2.13. Penelitian Sebelumnya**

“Perancangan Smart Home Berbasis Arduino” yang ditulis oleh Zulhipni Reno Saputra (2015-2016). Tujuan dari penelitian tersebut adalah membangun sebuah alat pengamanan rumah dengan perancangan alarm menggunakan kamera untuk me monitoring keadaan rumah. Cara kerja sistem tersebut berupa mengintegrasikan semua alat dengan arduino dan output dari sistem yang bekerja dapat berupa sms dari modem yang sudah terhubung ataupun video hasil dari rekaman webcam yang terpasang. Metodologi yang dipakai berupa pengumpulan data primer dan data sekunder yang dikumpulkan secara bertahap.

“Aplikasi Rumah Pintar (Smart Home) Pengendali Peralatan Elektronik Rumah Tangga Berbasis Web” yang ditulis oleh Fiqiana Prasetyowati (2016). Tujuan dari penelitian tersebut adalah membangun system yang dapat melakukan

pengontrolan alat-alat elektronik didalam rumah. Cara kerja sistem tersebut yaitu dengan mengintegrasikan semua alat dengan *RaspberryPi* dengan fungsi kontrol ataupun output yang terdapat pada web yang juga sudah dihubungkan dengan sistem. Metodologi yang dipakai dalam penelitian tersebut berupa pengumpulan data dan wawancara.

“Perancangan Sistem Keamanan Ruangan Menggunakan *Raspberry Pi*” yang ditulis oleh Adhi Krisnawan (2015). Tujuan dari penelitian tersebut adalah membangun sistem yang dapat digunakan untuk melakukan pengamanan rumah atau kantor dengan menggunakan sensor gerak dan kamera sebagai pemantau yang dihubungkan ke smartphone menggunakan *mikrokontroller RaspberryPi*. Cara kerja sistem tersebut yaitu dengan mengintegrasikan semua alat dengan *RaspberryPi* yang juga terhubung dengan smartphone berbasis android berupa alert yang masuk kedalam smartphone dan dapat tersimpan dalam berupa bentuk video. Metodologi yang dipakai dalam penelitian ini berupa pengumpulan data.

“*Smart House Berbasis Web*” (2016-2017). Penelitian ini bertujuan untuk membantu pemilik rumah agar dapat memantau, mengontrol dan mengamankan rumah secara otomatis dan manual. Cara kerja sistem ini berupa monitoring otomatis yang akan menggunakan sensor suara, sensor suhu, sensor gas dan kelembaban, dan sensor gerak yang terhubung dengan *RaspberryPi* dengan hasil yang ditangkap sensor-sensor tersebut datanya dimasukkan ke database dan dari database dikeluarkan atau ditampilkan dalam web yang sudah terhubung dengan database juga. Sistem ini dibangun berbasis *web* dengan menggunakan *Laravel*

*Framework 5.4* dan MySQL sebagai basis datanya. Metodologi yang dipakai adalah pengumpulan data dari sumber yang berkaitan dengan sistem yang akan dibuat.

### **3. METODE PENELITIAN**

#### **3.1. Waktu dan Tempat Penelitian**

Penelitian ini dilaksanakan di Kampus Fakultas Teknik Universitas Negeri Jenderal Soedirman yang beralamat di Jl. Mayjend Sungkono KM.5 Blater, Kalimanah Purbalingga, Jawa Tengah 53316. Penelitian dimulai pada bulan Maret 2017 sampai dengan bulan Juni 2017.

#### **3.2. Alat dan Bahan Penelitian**

##### **3.2.1. Alat Penelitian**

Alat yang digunakan dalam penelitian ini berupa perangkat keras (*hardware*) dan perangkat lunak (*software*) yang terdiri dari:

**1. Laptop dengan spesifikasi :**

- *Processor* Intel(R) Core(TM) i3-4200U CPU @1.80 GHz
- RAM 6 GB
- Sistem Operasi Microsoft Windows 10 Enterprise x64

**2. Aplikasi yang digunakan, antara lain :**

- Microsoft Office Word 2016
- Gammu For Linux
- Balsamiq
- Laragon
- Sublime 64bit
- MySQL
- Adobe Photoshop CC 2017

### 3.2.2. Bahan Penelitian

Bahan yang digunakan dalam penelitian ini terdiri dari :

1. *Raspberry Pi*
2. Sensor suhu DHT22
3. Sensor Hujan
4. Sensor PIR/*Motion/InfraRed*
5. Sirine Mobil 12V
6. Lampu
7. Sensor Cahaya
8. Kabel
9. Relay
10. Papan
11. Kipas Angin.
12. Kabel *Jumper*
13. PCB
14. *Motor Stepper*
15. Servo 180
16. Switch

### 3.3. Prosedur Kerja

Penelitian ini dilakukan dengan menggunakan model *Waterfall*. Model proses *waterfall* terdiri dari lima tahap, yaitu sebagai berikut:

a. Analisis dan Definisi Persyaratan

Tahap pertama ini yang akan menjadi dasar dalam pengembangan sebuah sistem. Analisa kebutuhan dan definisi persyaratan ini akan menghasilkan kebutuhan pengguna atau yang sering disebut dengan *user requirement*. Tahap ini dilakukan untuk mengumpulkan data yang terkait dengan sistem yang akan dibangun. Pengumpulan data dilakukan dengan cara observasi dan studi pustaka.

- Observasi

Penulis akan mengobservasi sistem rumah pintar yang sudah ada, sehingga penulis mendapatkan gambaran yang lebih baik.

- Studi Pustaka

Studi pustaka merupakan teknik pengumpulan data dengan mengambil data-data dari buku, jurnal, artikel dan media yang lain.

Data yang dikumpulkan dari observasi dan studi pustakan akan diolah dan di analisis. Dari analisis tersebut akan diketahui kebutuhan pengguna, dan kebutuhan sistem. Hasil analisis dijadikan acuan dalam membuat perancangan dan pengembangan sistem.

b. Perancangan Sistem dan Perangkat Lunak

Perancangan sistem adalah perancangan basis data dan penggambaran sketsa serta pengaturan komponen yang akan ada pada



sistem. Pembuatan desain sistem dilakukan berdasarkan hasil analisa kebutuhan yang dilakukan pada tahap pertama. Tahap desain sistem akan menghasilkan *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, *collaboration diagram*, dan desain antarmuka yang akan digunakan sebagai acuan dalam penulisan kode program pada tahap selanjutnya.

#### c. Implementasi dan Pengujian Unit

Tahap ini adalah tahap penulisan kode program dengan menerjemahkan desain sistem yang telah dibuat ke dalam bentuk perintah-perintah yang dimengerti oleh komputer dengan menggunakan bahasa pemrograman. Bahasa pemrograman yang akan digunakan adalah Bahasa Pemrograman PHP dan Python. Basis data yang digunakan adalah MySQL, sedangkan tampilan akan menggunakan HTML, CSS, Java Script, dan JQuery. Bersamaan dengan penulisan kode program, dari tahap ini akan menghasilkan sistem yang diinginkan.

#### d. Integrasi dan Pengujian Sistem

Tahap integrasi dan pengujian sistem merupakan tahap pengujian terhadap sistem secara keseluruhan. Pengujian ini dilakukan untuk mengetahui kesalahan-kesalahan yang ada pada sistem yang tidak sesuai dengan desain sistem yang telah dibuat.

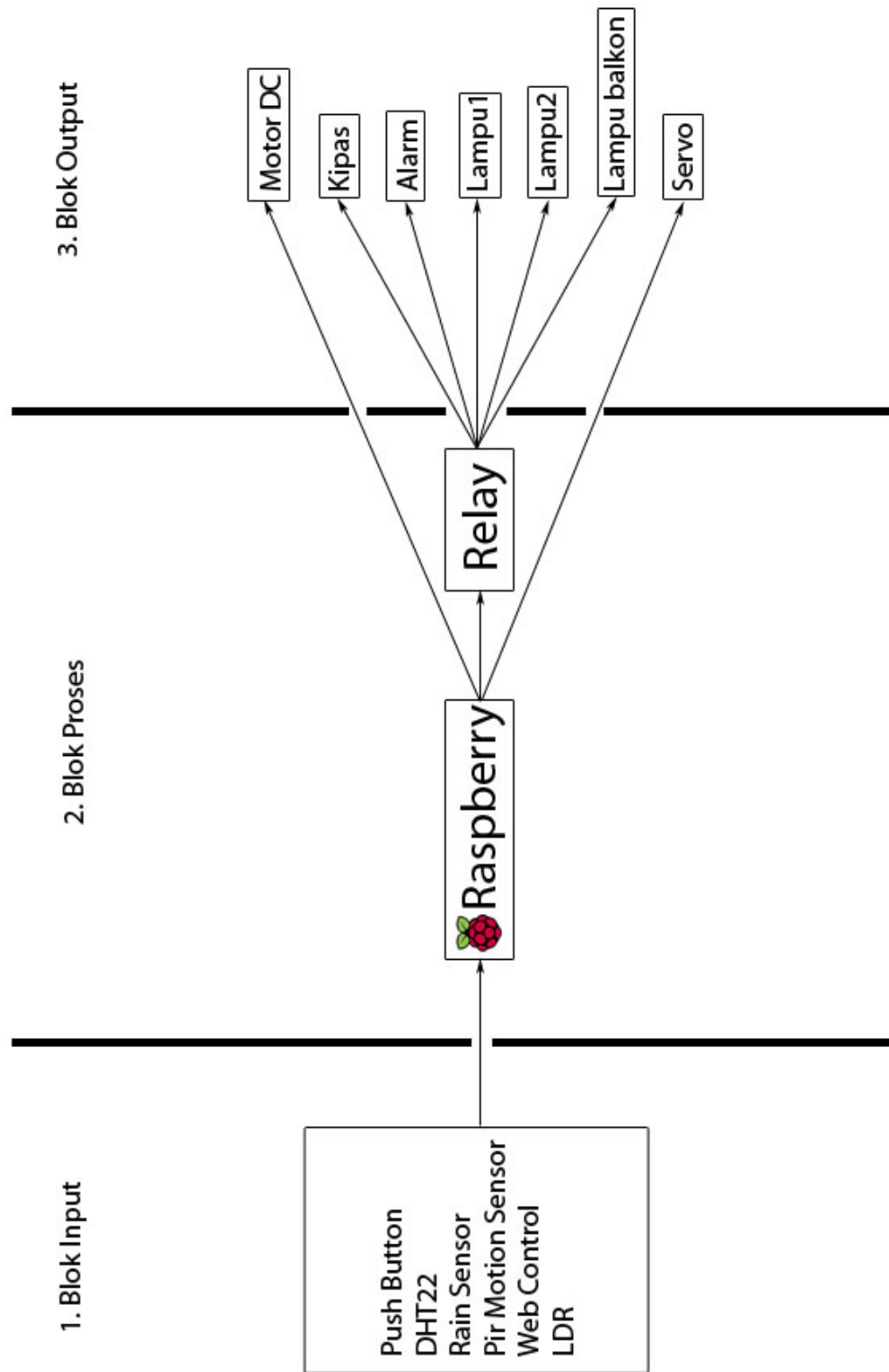
Ada dua jenis pengujian yang akan dilakukan pada penelitian ini yaitu pengujian perangkat keras dan pengujian perangkat lunak. Perangkat keras akan diuji dengan menguji semua respon sensor yang terhubung

dengan dalam menanggapi request dari Web oleh pengguna secara manual, dan respon sensor PIR yang terhubung dengan *buzzer* sebagai *alert* terhadap program yang telah dirancang secara otomatis akan hidup ketika sensor PIR menerima pergerakan tubuh manusia. Pengujian perangkat lunak dilakukan dengan menguji data yang diambil oleh sensor yang disimpan pada server dan ditampilkan ke Web.

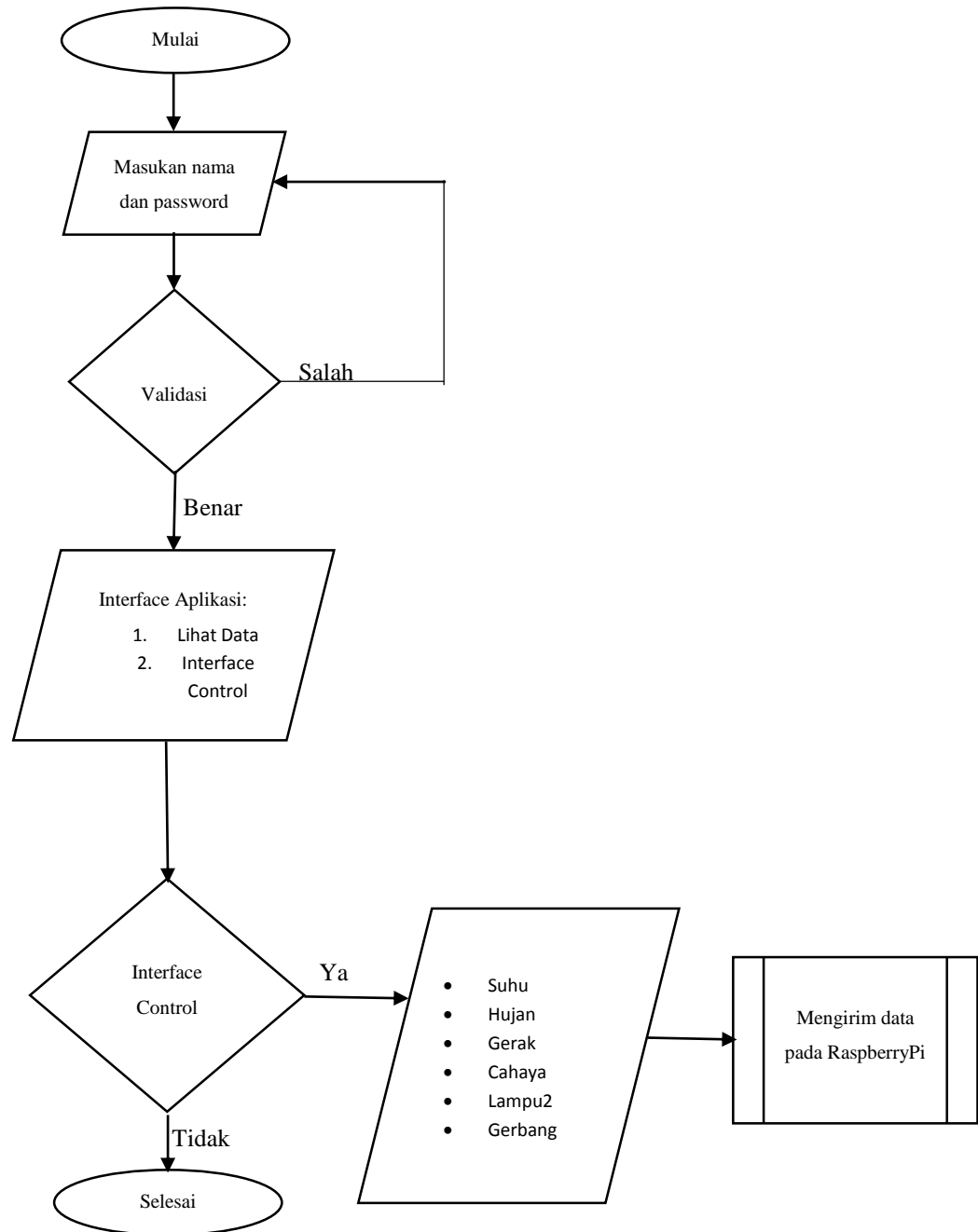
e. Operasi dan Pemeliharaan

Tahap operasi adalah tahap menerapkan perangkat lunak pada lingkungan instansi. Sedangkan tahap pemeliharaan dilakukan dengan membuat *user manual*. *User manual* dapat digunakan sebagai pedoman pengguna dalam menjalankan sistem sehingga sistem dapat dijalankan dengan tetap dan sistem terjaga dengan baik, serta sebagai pencegahan masalah yang mungkin akan timbul pada sistem saat dijalankan.

### 3.4. Diagram Blok



### 3.5. Flowchart Sistem



### 3.6. Jadwal Penelitian

Rincian jadwal penelitian Smart House Berbasis Web ditunjukkan pada

Tabel 3.1.

Tabel 3.1. Jadwal Penelitian

Prosedur Kerja	Durasi	Mulai	Selesai
<b>Analisis Kebutuhan</b>	<b>10 hari</b>	<b>Rabu, 01/03/2017</b>	<b>Jumat, 10/03/2017</b>
1. Studi Literatur	4 hari	Rabu, 01/03/2017	Sabtu, 04/03/2017
2. Analisis aplikasi sejenis	3 hari	Minggu, 05/03/2017	Selasa, 07/03/2017
3. Wawancara	3 hari	Rabu, 08/03/2017	Jumat, 10/03/2017
<b>Desain Aplikasi</b>	<b>30 hari</b>	<b>Sabtu, 11/03/2017</b>	<b>Minggu, 09/04/2017</b>
4. Membuat perancangan aplikasi	20 hari	Sabtu, 11/03/2017	Kamis, 30/03/2017
5. Membuat interface aplikasi	10 hari	Jumat, 31/03/2017	Minggu, 09/04/2017
<b>Development</b>	<b>40 hari</b>	<b>Senin, 10/03/2017</b>	<b>Jumat, 19/05/2017</b>
6. <i>Coding</i>	33 hari	Senin, 10/03/2017	Jumat, 12/05/2017
7. Pengujian fungsi kode	7 hari	Sabtu, 13/05/2017	Jumat, 19/05/2017
<b>Pengujian</b>	<b>14 hari</b>	<b>Sabtu, 20/05/2017</b>	<b>Jumat, 02/06/2017</b>
8. Blackbox testing	7 hari	Sabtu, 20/05/2017	Rabu, 26/05/2017
9. Debugging aplikasi	7 hari	Kamis, 27/05/2017	Jumat, 02/06/2017
<b>Penyusunan laporan</b>	<b>14 hari</b>	<b>Sabtu, 03/06/2017</b>	<b>Jumat, 16/06/2017</b>
10. Dokumentasi Perancangan	7 hari	Sabtu, 03/06/2017	Jumat, 09/06/2017
11. Dokumentasi implementasi	7 hari	Sabtu, 10/06/2017	Jumat, 16/06/2017
<b>Deployment</b>	<b>7 hari</b>	<b>Sabtu 17/06/2017</b>	<b>Jumat, 23/06/2017</b>
12. Membuat cron job	7 hari	Sabtu 17/06/2017	Jumat, 23/06/2017

## **4. HASIL DAN PEMBAHASAN**

### **4.1. Analisis Kebutuhan**

Tahap analisis kebutuhan dilakukan untuk mengetahui kebutuhan sistem yang akan dibangun. Cara memperoleh data dan informasi yang dibutuhkan untuk bahan analisis adalah dengan wawancara dan membandingkan aplikasi sejenis.

#### **4.1.1. Pengumpulan Data**

Adapun pengumpulan data yang dilakukan dalam pembuatan sistem ini dilakukann dengan dua cara yaitu wawancara dan analisis aplikasi sejenis. Berikut ini hasil pengumpulan data dengan cara wawancara :

##### **4.1.1.1. Wawancara**

Wawancara dilakukan kepada narasumber secara acak baik kapan dan apa pertanyaan wawancara tersebut. Wawancara tersebut dilakukan dengan menggunakan media sosial ataupun secara langsung.

##### **4.1.1.2. Analisis Aplikasi Sejenis**

Aplikasi yang berhubungan dengan kontrol alat elektronik di rumah sudh semakin banyak dan juga dikomersilkan. Oleh karena itu, aplikasi yang akan penulis buat akan dibandingkan dengan aplikasi yang sudah beredar ataupun aplikasi hasil penelitian peneliti lain dan mengadaptasi kelebihan dan kekurangannya. Berikut aplikasi sejenis yang penulis analisa:

## 1. Philips Hue

Philips Hue adalah suatu aplikasi *mobile* android yang dikeluarkan oleh philips yaitu salah satu perusahaan lampu terkemuka di dunia. Aplikasi tersebut digunakan sebagai remot pengendali lampu yang baru saja beberapa bulan lalu di *release* oleh perusahaan lampu tersebut. Jenis lampu atau nama lampu tersebut adalah Hue. Lampu Hue ini dapat diatur warna, intensitas brightness, menyalakan atau mematikan lampu yang semuanya itu di kendalikan oleh aplikasi *mobile* android yang bernama Philips Hue tersebut.

Berikut ini kelebihan dan kekurangan dari aplikasi Philips Hue hasil analisis:

### a. Kelebihan

1. Berdaya rendah.
2. Mudah untuk di kontrol.
3. Bisa dihubungkan ke *message application mobile phone* dan menjadi indicator ketika ada pesan yang masuk ke smartphone.

### b. Kekurangan

1. Koneksi wifi kadang terputus sendiri

## 2. Amazon Alexa

Suatu produk terbaru dari amazon, yaitu *app mobile* android asisten digital untuk mewujudkan rumah pintar. App tersebut tinggal atau tertanam pada perangkat amazon bernama echo, yaitu *speaker* yang terhubung dengan perangkat *wi-fi* dan dilengkapi mikrofon yang dapat diajak bicara.

Alexa adalah kecerdasan buatan yang dapat dimanfaatkan untuk mencari informasi dasar, kemampuan kecerdasan buatan ini juga masih dikembangkan agar lebih bisa menerima perintah.

Berikut ini kelebihan dan kekurangan dari aplikasi Amazon Alexa hasil analisis:

a. Kelebihan

1. Merupakan sistem cerdas.
2. Tidak harus berbicara tepat didepan echo.
3. Dapat memberikan berbagai macam informasi dan memainkan musik dan menyala-matikan lampu.
4. Selalu update

b. Kekurangan

1. Tidak mempunyai baterai yang sudah *built-in* sehingga harus disambungkan terus dengan arus listrik.
2. Belum tersedia untuk wilayah indonesia.

### 3. *Alfred Home Security Camera*

*Alfred Home Security Camera* adalah suatu app yang dapat mengatur pergerakan apapun pada cctv yang di pasang dirumah atau dimanapun selama cctv tersebut terhubung dengan koneksi internet.

Berikut ini kelebihan dan kekurangan dari aplikasi *Alfred Home Security Camera* hasil analisis:



a. Kelebihan

1. Hanya membutuhkan koneksi internet tanpa harus terhubung dengan kabel.
2. Dapat digunakan dalam 3 mode tampilan, yaitu *live streaming*, *motion detected*, *night vision and camera flash*.

b. Kekurangan

1. Butuh koneksi yang stabil.
2. Tampilan video menyesuaikan koneksi internet.
3. Baterai smartphone yang digunakan sebagai kamera akan cepat habis.

### Kesimpulan

Berikut ini perbandingan ketiga aplikasi diatas yang ditunjukkan pada tabel 4.1.

**Tabel 4.1.** Perbandingan Aplikasi Sejenis

	<b>Philips Hue</b>	<b>Amazon Alexa</b>	<b><i>Alfred Home Security Camera</i></b>
Membutuhkan Koneksi Internet	Ya	Ya	Ya
Mobile Apps	Ya	Ya	Ya
Baterai Built-In	No	No	No
Interaktif	No	Yes	No

#### 4.1.2. Analisis Kebutuhan Pengguna

Berdasarkan pada pengumpulan data, diperoleh pengguna yang berperan dalam aplikasi yaitu hanya *end user* atau pengguna terakhir. Pengguna terakhir ini

adalah pemilik dari app dan sistem tersebut yang sudah dibuat oleh seorang programmer.

#### **4.1.3. Analisis Kebutuhan Aplikasi**

Analisis kebutuhan aplikasi merujuk dari kebutuhan pengguna. Kebutuhan pengguna aplikasi terbagi menjadi dua, yaitu kebutuhan fungsional dan non-fungsional. Berikut definisinya.

##### **4.1.3.1. Kebutuhan Fungsional**

Kebutuhan fungsional merupakan kebutuhan fungsi atau proses yang harus ada dalam aplikasi dan mempengaruhi kinerja aplikasi. Jika fungsi atau proses ini tidak terdapat dalam sistem, maka sistem tidak dapat bekerja secara maksimal dan sebagai mana mestinya. Kebutuhan fungsional dari Smart House sebagai berikut:

1. Aplikasi menyediakan fungsi untuk login.
2. Aplikasi menyediakan fungsi untuk reset password.
3. Aplikasi memiliki fitur mengubah data pada database
4. Aplikasi memiliki fitur ubah tema/tampilan sistem.

##### **4.1.3.2. Kebutuhan Non-Fungsional**

Kebutuhan non-fungsional merupakan kebutuhan yang tidak terkait dengan perancangan dan pembangunan aplikasi, namun mempengaruhi jalannya sistem. Kebutuhan non-fungsional dari Smart House sebagai berikut:

1. Aplikasi diinstal pada server Raspberry.
2. Aplikasi diakses menggunakan aplikasi browser.
3. Aplikasi dapat diakses ketika berada dalam satu jaringan.

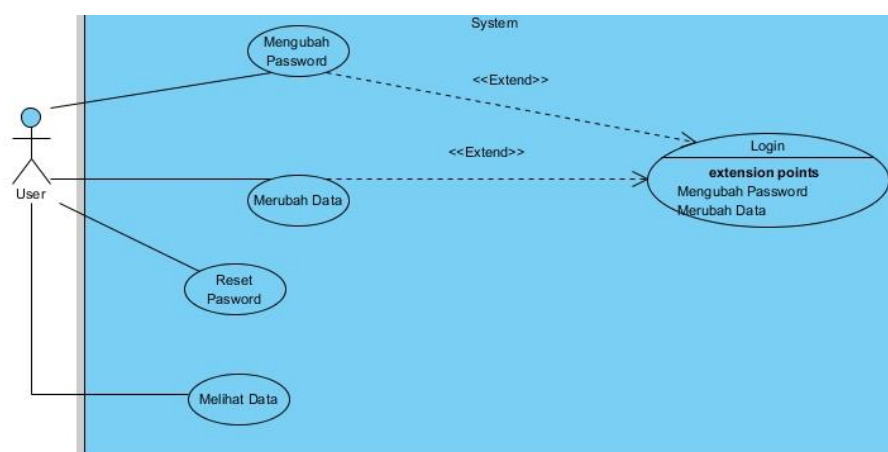
## 4.2. Perancangan Sistem dan Perangkat Lunak

Perancangan sistem dibuat berdasarkan hasil analisis yang telah dilakukan dan dijelaskan pada analisis kebutuhan sistem. Perancangan sistem menggunakan konsep *Object Oriented Programming* (OOP) dengan bahasa *Unified Modeling Language* (UML). Perancangan ini akan menghasilkan tiga diagram, yaitu *use-case* diagram, *sequence* diagram dan *class* diagram. Pembuatan UML sendiri menggunakan perangkat lunak Visual Paradigm.

Tahap desain juga mencakup perancangan tampilan antarmuka pengguna sistem. Perancangan ini menggunakan tipe *high-fidelity* sehingga memiliki tingkat kedetailan yang cukup tinggi pada setiap halaman item tertentu. Pembuatan gambar antarmuka menggunakan perangkat lunak Balsamiq.

### 4.2.1. Use Case Diagram

*Use case diagram* menggambarkan fungsionalitas dari aplikasi, serta mempresentasikan interaksi antar aktor/pengguna dengan sistem. *Use case diagram* dari aplikasi ini dapat ditunjukkan pada Gambar 4.2.1.



**Gambar 4.1. Use Case Diagram Aplikasi**

Gambar 4.1 menunjukkan bahwa pada aplikasi terdapat satu aktor yang berinteraksi dengan aplikasi, yaitu hanya *User*. *Use case* disesuaikan dengan kebutuhan fungsionalitas aplikasi dan kebutuhan pengguna yang telah didapatkan sebelumnya. Penjelasan *use case* terhadap kebutuhan fungsionalitas aplikasi dan pengguna dapat dilihat pada tabel 4.1.

**Tabel 4.2.** Tabel *Use Case* Kebutuhan Fungsionalitas Aplikasi dan Pengguna.

<i>Use Case</i>	<i>Actor</i>
<i>Login</i>	<i>User</i>
<i>Reset Password</i>	<i>User</i>
Mengubah Data	<i>User</i>
Melihat Data	<i>User</i>
Mengubah Password	<i>User</i>

*Use Case* diagram pada Gambar 4.1 memiliki 5 *case* yang masing-masing dijelaskan dengan *use case narrative* berikut ini:

#### 1. *Use Case Narrative Login*

*Use case narrative login* dapat ditunjukan pada tabel 4.2.

**Tabel 4.3.** Tabel *Use Case Narrative Login*

Nama Case	<i>Login</i>	
Aktor	<i>User</i>	
Deksripsi	Proses melakukan <i>login</i>	
Kondisi Sebelum	Pengguna berada pada menu <i>login</i>	
Pemicu	Pengguna menekan tombol <i>login</i>	
Urutan Kejadian	Aksi Aktor	Respon Sistem
	Pengguna memasukan e-mail dan Password pada <i>form login</i>	Sistem melakukan verifikasi data dan menampilkan halaman sesuai dengan hak akses
Kondisi Akhir	Pengguna berada pada halaman sesuai dengan hak akses	

## 2. Use Case Narrative Reset Password

Use case narrative Reset Password dapat ditunjukkan pada tabel 4.3.

**Tabel 4.4.** Tabel Use Case Narrative Reset Password

Nama Case	Reset Password	
Aktor	User	
Deksripsi	Proses <i>reset password</i> ketika lupa <i>password</i> untuk login	
Kondisi Sebelum	Pengguna berada pada menu <i>login</i>	
Pemicu	Pengguna menekan tombol <i>reset password</i>	
Urutan Kejadian	Aksi Aktor	Respon Sistem
	1. Pengguna menekan tombol <i>reset password</i> .	Sistem menampilkan form kirim e-mail <i>reset password</i>
	2. Pengguna memasukkan e-mail reset password yang sudah terdaftar	Sistem mengirim e-mail ke e-mail <i>reset password</i> . Programmer merubah <i>password</i> default
	3. Pengguna membuka e-mail dan menerima <i>password</i> yang telah di <i>reset</i> .	
Kondisi Akhir	Pengguna sukses login dengan <i>password default</i>	

## 3. Use Case Narrative Mengubah Data

Use case narrative mengubah data dapat ditunjukkan pada tabel 4.4.

**Tabel 4.5.** Tabel Use Case Narrative Mengubah Data

Nama Case	Mengubah Data	
Aktor	User	
Deksripsi	Proses mengubah data yang disimpan dalam <i>database</i>	
Kondisi Sebelum	Pengguna berada pada halaman lihat data setelah <i>login</i>	
Pemicu	Pengguna melakukan <i>login</i>	
Urutan Kejadian	Aksi Aktor	Respon Sistem

	1. Pengguna melakukan <i>login</i>	Sistem menampilkan halaman lihat data yang dapat diubah
	2. Pengguna menekan tombol kontrol pada menu data	Sistem memunculkan <i>popup form ubah data</i>
	3. Pengguna mengubah data dan menekan tombol simpan	Sistem menyimpan data yang telah diubah kedalam <i>database</i>
Kondisi Akhir	Pengguna berada pada halaman lihat data setelah <i>login</i>	

#### 4. Use Case Narrative Melihat Data

*Use case narrative* melihat data dapat ditujukan pada tabel 4.5.

**Tabel 4.6.** Tabel *Use Case Narrative* Melihat Data

Nama Case	Melihat Data	
Aktor	<i>User</i>	
Deksripsi	Proses menampilkan data dari <i>database</i>	
Kondisi Sebelum	Pengguna belum membuka sistem	
Pemicu	Pengguna membuka sistem	
Urutan Kejadian	Aksi Aktor	Respon Sistem
	1. Pengguna memasukkan ip <i>address</i> ke dalam browser	Sistem menampilkan halaman lihat data
Kondisi Akhir	Pengguna berada pada halaman lihat data setelah <i>login</i>	

#### 5. Use Case Narrative Mengubah Password

*Use case narrative* mengubah *Password* dapat ditujukan pada tabel 4.6.

**Tabel 4.7.** Tabel *Use Case Narrative* Mengubah Password

Nama Case	Mengubah <i>Password</i>
Aktor	<i>User</i>
Deksripsi	Proses mengubah <i>password</i> untuk login ke halaman kontrol/ubah data

Kondisi Sebelum	Pengguna berada pada halaman lihat data sebelum <i>login</i>	
Pemicu	Pengguna melakukan <i>login</i>	
Urutan Kejadian	Aksi Aktor	Respon Sistem
	1. Pengguna melakukan <i>login</i>	Sistem menampilkan halaman lihat data dengan menu ubah <i>password</i>
	2. Pengguna menekan tombol ubah <i>password</i>	Sistem memunculkan <i>form</i> ubah <i>password</i>
	3. Pengguna mengubah dan mengkonfirmasi <i>password</i> dan menekan tombol ubah	Sistem menyimpan <i>password</i> baru yang telah diubah kedalam <i>database</i>
Kondisi Akhir	Pengguna berada pada halaman lihat data setelah <i>login</i>	

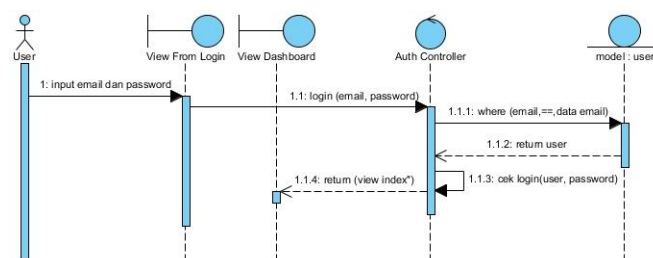
#### 4.2.2. Sequence Diagram

*Sequence diagram* menggambarkan skenario atau rangkaian langkah-langkah setiap *case* yang ada pada *use case diagram*. *Sequence diagram* terdiri atas dimensi vertikal yang menggambarkan waktu dan dimensi horizontal yang menggambarkan objek yang terkait dengan *case* tersebut.

##### 1. Sequence Diagram Login

*Sequence diagram login* dari *case login* dapat dilihat pada gambar

4.2 berikut ini:

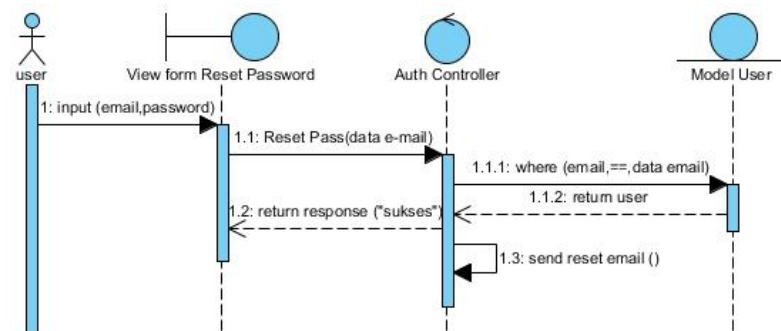


Gambar 4.2. Sequence Diagram Login

*Sequence Diagram Login* ini menerangkan proses pengguna *user* untuk masuk kedalam halaman kontrol aplikasi. Pengguna pada halaman aplikasi menekan tombol login lalu akan keluar pop-up *form* login dan di isi dengan e-mail dan *password* kemudian menekan tombol masuk. Data e-mail dan *password* dikirim ke *auth Controller* dan mengeksekusi fungsi *login*. Pada fungsi *login*, data e-mail dan *password* dibandingkan dengan data *user* yang ada di model *user*. Ketika data *user* yang ada di *database* sama dengan e-mail dan *password* yang di *input*-kan, penngguna akan dirujuk kehalaman dashboard aplikasi berupa kontrol aplikasi.

## 2. *Sequence Diagram Reset Password*

*Sequence diagram reset password* dari *case reset password* dapat dilihat pada gambar 4.3 berikut ini:



**Gambar 4.3. *Sequence Diagram Reset Password***

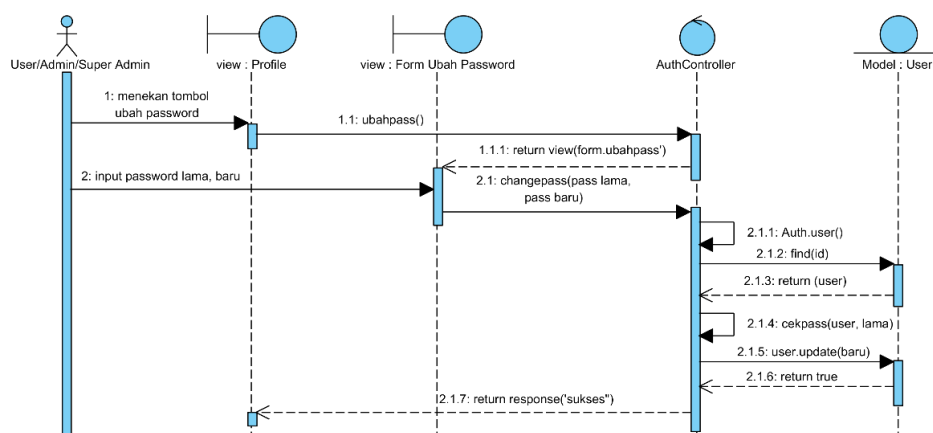
*Sequence diagram reset password* menerangkan proses mereset *password* menjadi *default* ketika pengguna lupa dengan *password*-nya. Pengguna dapat menekan tombol *reset password* yang ada pada menu diatas dan akan keluar *pop-up form reset password*. Pada *form* tersebut,



pengguna mengisi e-mail dan mengklik tombol *reset* yang akan memanggil fungsi *reset password*. Yang kemudian aplikasi akan mengubah *password* menjadi *default*.

### 3. *Sequence Diagram Ubah Password*

*Sequence diagram* ubah *password* dari *case* ubah *password* dapat dilihat pada gambar 4.4 berikut ini:



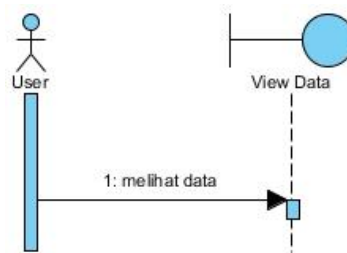
**Gambar 4.4. *Sequence Diagram Ubah Password***

*Sequence diagram* ubah *password* menjelaskan proses pengguna mengubah *password* lama dari *user*. Pada menu, pengguna menekan tombol ubah *password* dan akan keluar *pop-up* berisi *form ubah password*. Pada *form* ini pengguna harus mengisi *input-an password* lama, *password* baru dan konfirmasi *password*. Pada fungsi ini, data *password* lama akan dibandingkan dengan *password* pengguna yang ada di *database*. Jika data *password* lama benar, maka data *password* baru akan disimpan kedalam *database* menggantikan data *password* lama. Setelah berhasil mengubah,

pengguna akan kembali di hadapkan dengan halaman aplikasi sebelum *login*.

#### 4. *Sequence Diagram Melihat Data*

*Sequence diagram* melihat data dari *case* melihat data dapat dilihat pada gambar 4.5 berikut ini:

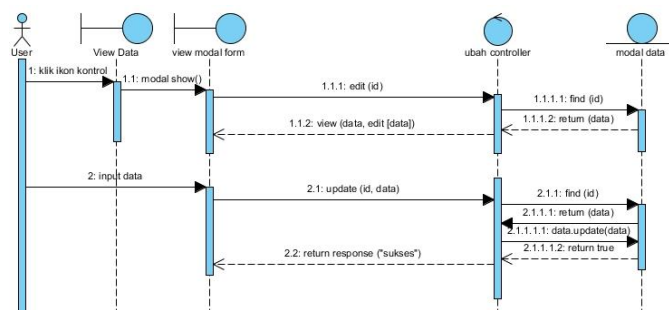


**Gambar 4.5. *Sequence Diagram Melihat Data***

*Sequence diagram* melihat data ini menjelaskan pengguna dalam melihat data. Pada saat pengguna mengunjungi alamat aplikasi, pengguna langsung dapat melihat data dalam aplikasi.

#### 5. *Sequence Diagram Mengubah Data*

*Sequence diagram* mengubah data dari *case* mengubah data dapat dilihat pada gambar 4.6 berikut ini:



**Gambar 4.6. *Sequence Diagram Mengubah Data***

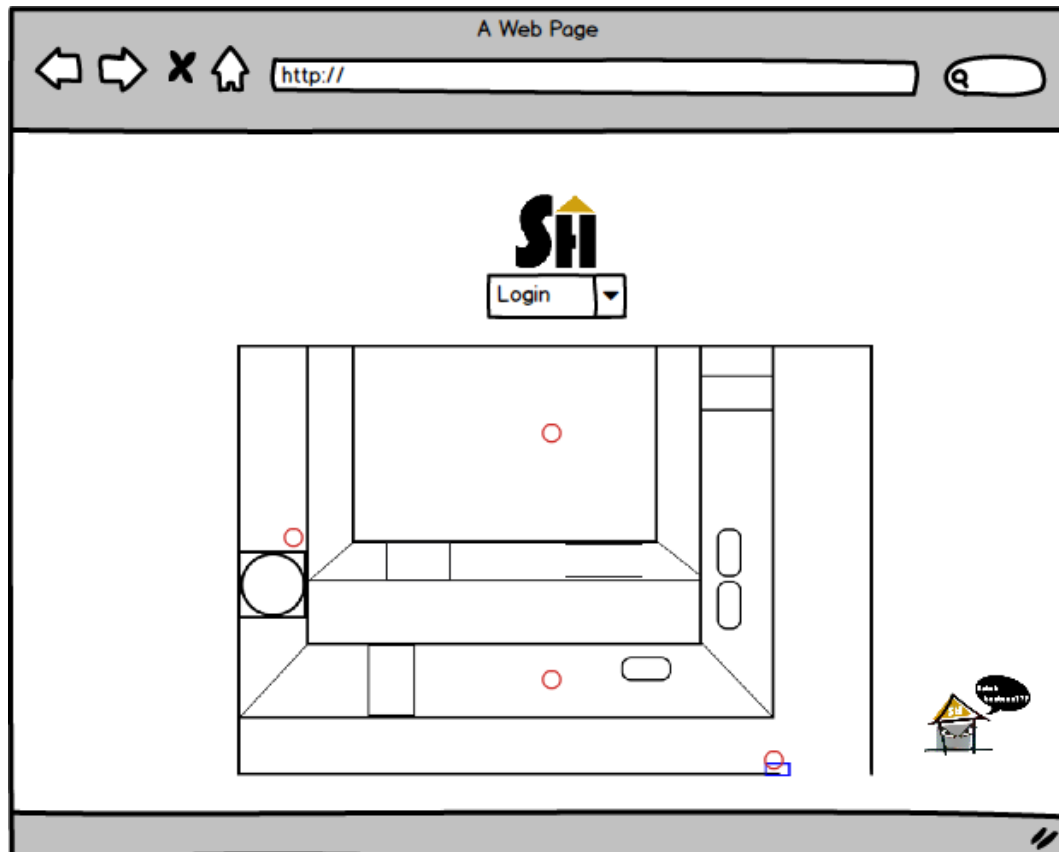
*Sequence diagram* mengubah data ini menggambarkan proses saat pengguna mengubah suatu data. *User* yang memiliki hak akses yang berada pada halaman lihat data menekan ikon kontrol pada data yang ingin diubah. Kemudian sebuah modal muncul dan melakukan fungsi *edit()* pada *UbahController* dengan membawa id data yang didapatkan pada ikon kontrol. Pengguna melakukan perubahan data dan disimpan dengan menekan tombol simpan. Setelah itu data akan dikirim ke fungsi *update()* yang dimiliki oleh *UbahController*. Fungsi *update()* ini melakukan pencarian data berdasarkan id data dengan fungsi *find()*. Fungsi *find()* mengembalikan sebuah *object* data yang digunakan untuk memanggil fungsi *update* yang dimiliki modal data. Data yang sudah di ubah digunakan untuk masukan fungsi *update()* tersebut. Ketika perubahan berhasil disimpan, fungsi *update()* mengembalikan nilai *true* dan mengembalikan pesan sukses ke halaman lihat data.

#### **4.2.3. Desain Antar Muka Pengguna Aplikasi**

Pada tahap desain antar muka pengguna, dirancang tampilan antar muka untuk pengguna sistem dengan menyesuaikan *use case* yang ada beserta proses bisnis yang dibutuhkan. Desain antar muka didokumentasikan dalam bentuk *mockup* dengan konsep *highly-fidelity wireframes*. *Mockup* dari desain antar muka pengguna Aplikasi Smart House yaitu sebagai berikut:

#### 4.2.3.1. Rancangan Halaman Lihat Data

Rancangan halaman lihat data aplikasi dapat dilihat pada gambar 4.7 berikut ini:

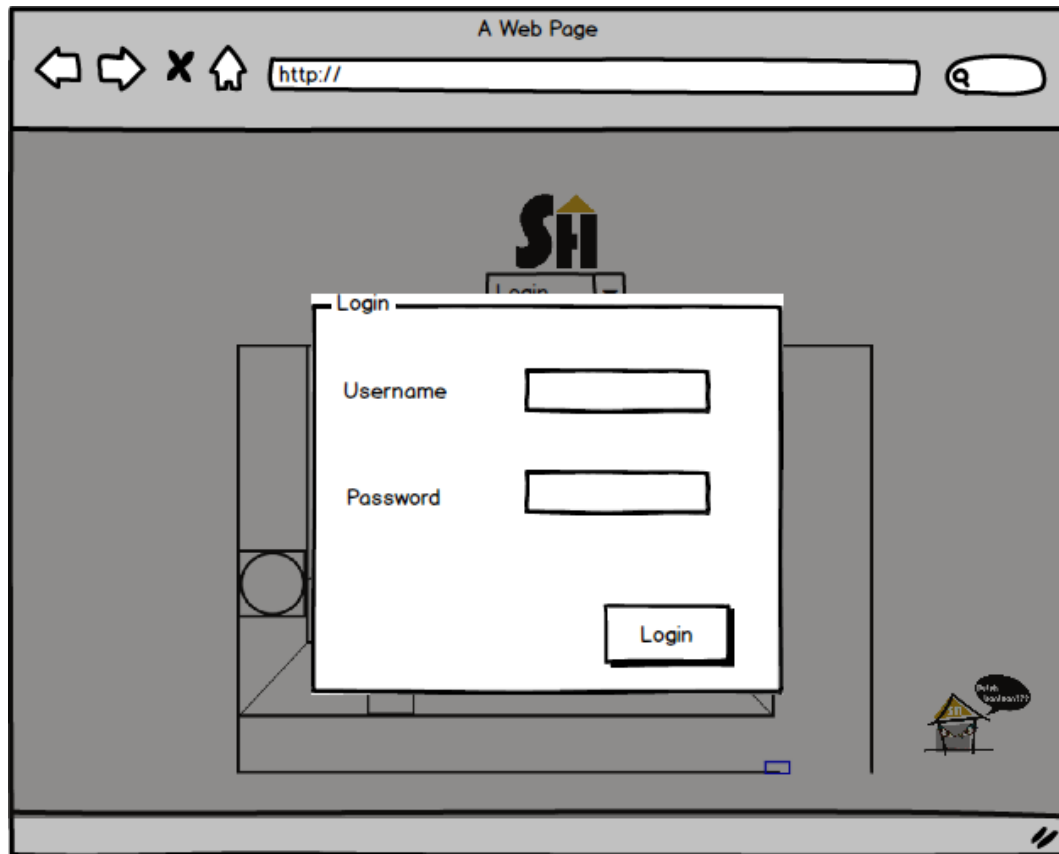


**Gambar 4.7. Rancangan Halaman Lihat Data**

Rancangan halaman lihat data memiliki tampilan utama, yaitu terdapat animasi yang membentuk denah rumah yang menampilkan data aplikasi. Dimana dibagian-bagian tertentu sudah diberikan switch untuk mengontrol atau mengubah nilai data tersebut (namun harus melakukan proses login terlebih dahulu) “ditandai dengan simbol berwarna merah”. Di halaman ini juga terdapat tombol login dan bantuan.

#### 4.2.3.2. Rancangan Halaman Lihat Data *Pop-up Login*

Rancangan halaman data lihat *pop-up login* aplikasi dapat dilihat pada gambar 4.8 berikut ini:

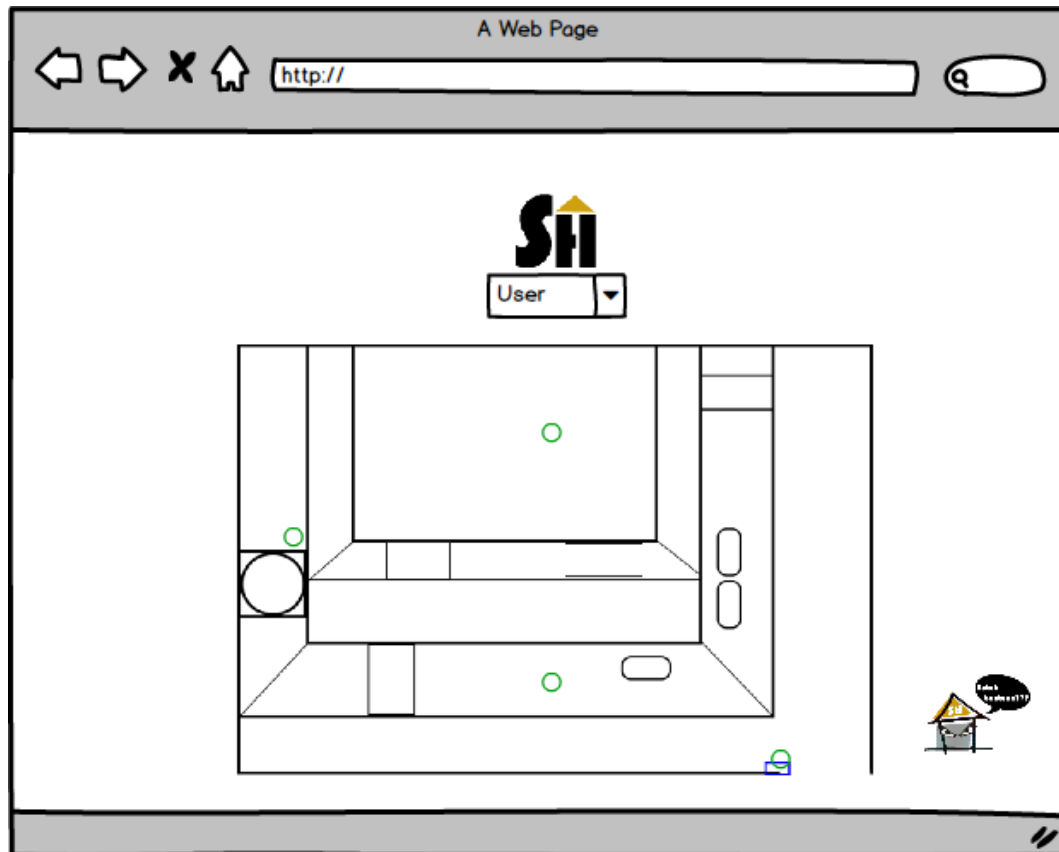


**Gambar 4.8. Rancangan Halaman Lihat Data *Pop-up Login***

Rancangan halaman lihat data *pop-up login* merupakan halaman lihat data awal yang memunculkan *pop-up login* ketika pengguna menekan tombol/menu *login*. Dimana dalam *pop-up login* tersebut berisi *form login* berupa input e-mail dan *password*.

#### 4.2.3.3. Rancangan Halaman Lihat Data dan Ubah Data

Rancangan halaman data lihat dan ubah data aplikasi dapat dilihat pada gambar 4.9 berikut ini:

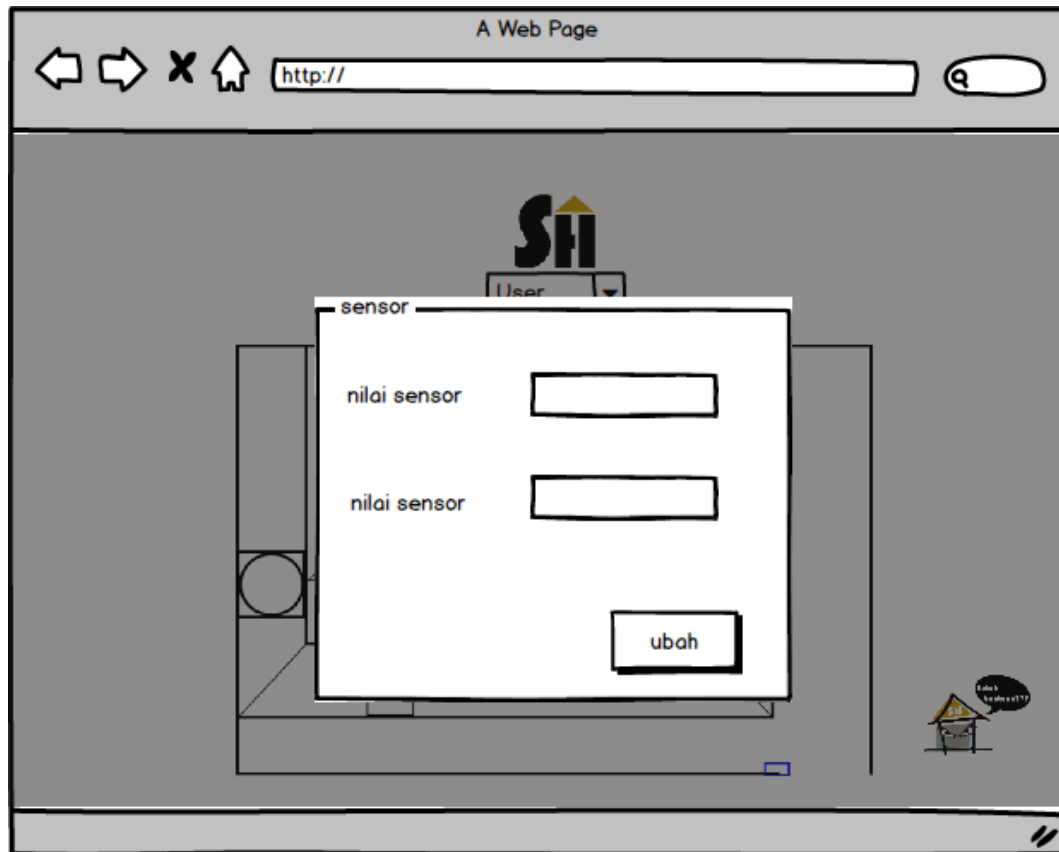


**Gambar 4.9. Rancangan Halaman Lihat Data dan Ubah Data**

Rancangan halaman lihat data dan ubah data merupakan halaman lihat data setelah pengguna melakukan proses login. Dimana setelah proses login berhasil, maka pengguna akan membuka hak akses ubah data yang ditandai dengan berubahnya warna swtch menjadi hijau, tanda switch dapat ditekan dan mengeluarkan form ubah nilai data.

#### 4.2.3.4. Rancangan Halaman Lihat Data dan Ubah Data *Pop-up* Ubah Data

Rancangan halaman data lihat dan ubah data *pop-up* ubah data aplikasi dapat dilihat pada gambar 4.10 berikut ini:

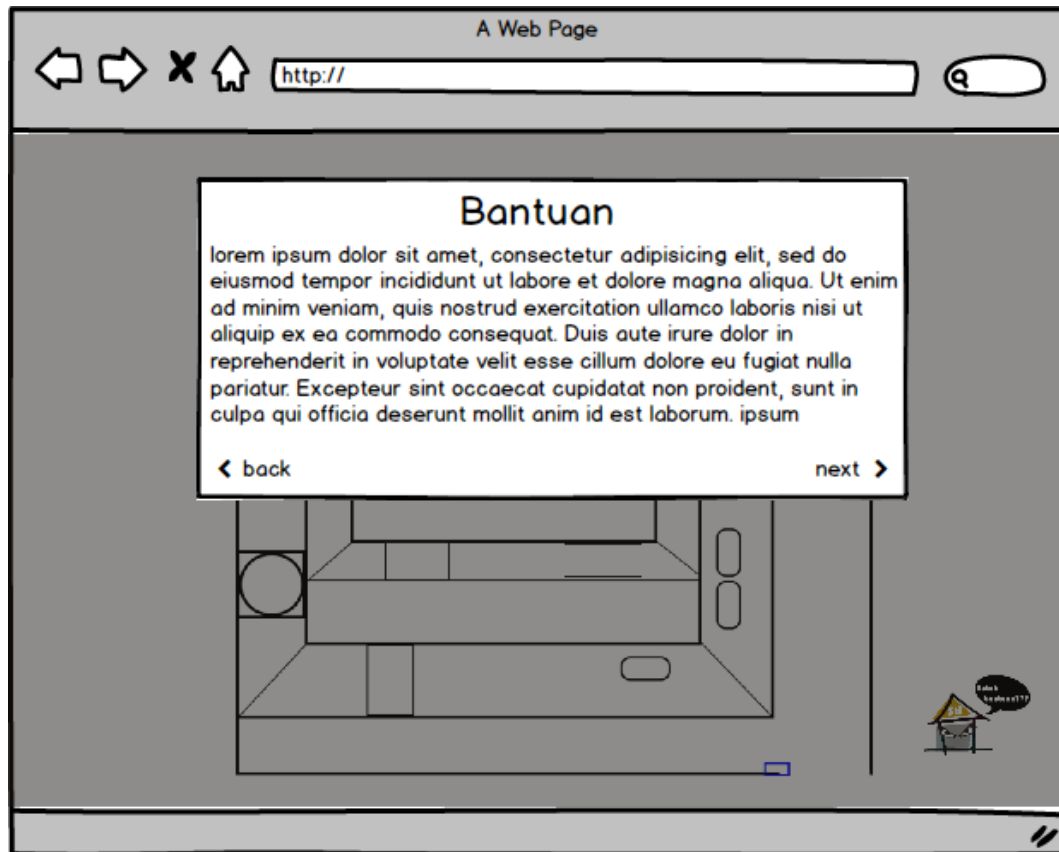


**Gambar 4.10. Rancangan Halaman Lihat Data dan Ubah Data *Pop-up* Ubah Data**

Rancangan halaman lihat data dan ubah data *pop-up* ubah data merupakan halaman lihat data setelah pengguna menekan pilihan ubah data yang kemudian di proses dan memunculkan *pop-up form* ubah data. Dimana isi *form* tersebut berupa data-data dalam tabel database yang dapat pengguna rubah nilainya setelah menekan pilihan ubah.

#### 4.2.3.5. Rancangan Halaman *Pop-Up* Bantuan

Rancangan halaman data lihat dan ubah data *pup-up* ubah data aplikasi dapat dilihat pada gambar 4.11 berikut ini:



**Gambar 4.11. Rancangan Halaman *Pop-Up* Bantuan**

Rancangan halaman *pou-up* bantuan merupakan halaman setelah *login* yang memunculkan suatu *pop-up* yang berisi informasi mengenai tiap halaman beserta fungsi *tools*-nya dan terdapat juga tombol *reset password* ke bentuk *default system* ketika terjadi masalah lupa password.



### 4.3. Implementasi

Pada tahap implementasi ini penulis akan memaparkan sistem yang dibuat berdasarkan rancangan yang telah dibuat sebelumnya. Pertama miniatur *SmartHouse* beserta rangkaian elektroniknya akan dijelaskan terlebih dahulu. Kemudian *koSmartde* pada *server* dan diakhiri dengan kode *python* pada *Raspberry*.

#### 4.3.1. Miniatur *SmartHouse* Beserta Rangkaian

Rumah yang digunakan dalam penelitian ini hanyalah sebuah miniatur. Dimana beberapa dari alat dan rangkaian yang ada dapat diterapkan pada rumah yang sebenarnya. Berikut merupakan gambar dari miniatur *SmartHouse* yang telah dibuat.

##### 1. Miniatur *SmartHouse* Tampak Depan

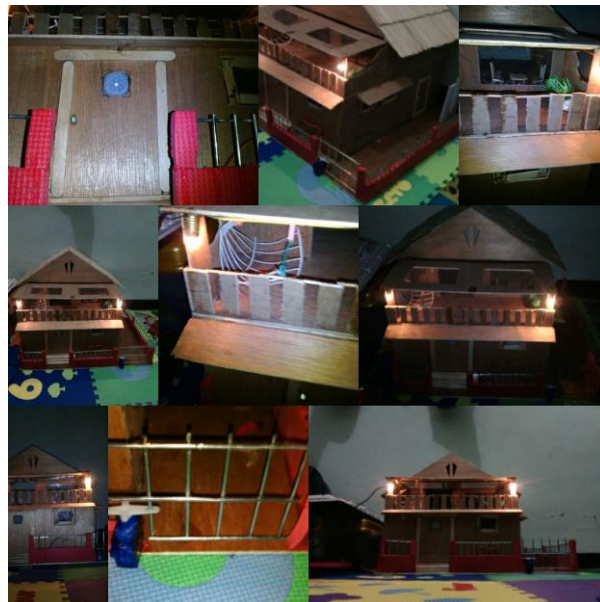


**Gambar 4.12. Miniatur *SmartHouse* Tampak Depan**

Gambar diatas merupakan miniatur rumah tampak depan yang digunakan dalam peniltian ini. Dari depan kita dapat melihat pagar yang berada pada sebelah

kanan. Pada pagar tersebut dipasang suatu penggerak (servo 180°) yang berguna untuk membuka pagar secara otomatis tanpa menggunakan tenaga secara manual dengan cara merubah kondisi nilai pagar yang ada dalam database melalui website. Kemudian di pintu utama terdapat suatu alat pendeteksi gerakan lewat panas tubuh manusia. Dimana alat tersebut dihubungkan dengan sirine yang akan berbunyi jika sensor pendeteksi gerak tersebut mendeteksi suatu gerakan manusia disekitarnya. Dilantai dua dari depan dapat dilihat dua lampu kecil yang dipasing di masing-masing sisi. Kedua lampu tersebut terhubung dengan sensor cahaya (LDR). Jadi ketika sensor tersebut mendeteksi cahaya, maka lampu akan mati. Dan sebaliknya ketika sensor tersebut tidak mendapatkan cahaya, maka lampu akan menyala.

## 2. Miniatur *SmartHouse* Tampak Samping Kanan

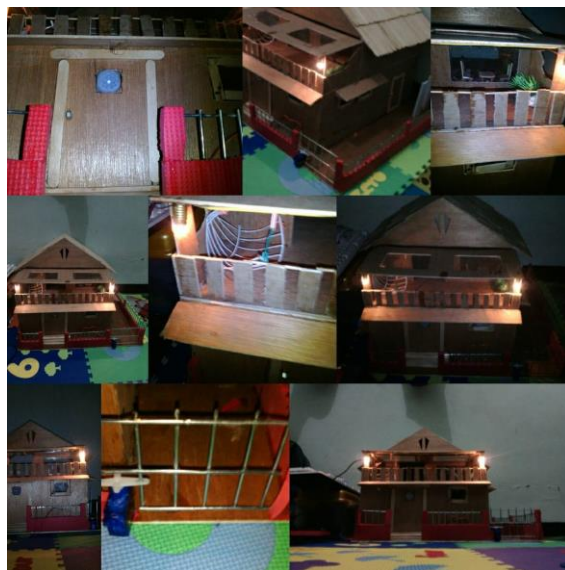


**Gambar 4.13. Miniatur *SmartHouse* Tampak Samping Kanan**

Gambar diatas merupakan miniatur rumah tampak samping sebelah kanan yang digunakan dalam penelitian ini. Disitu dapat dilihat ada bagian yang tertutup

atap dan tidak tertutup atap. Bagian tersebut merupakan daerah untuk menjemur pakaian, dimana tiang jemuran tersebut dipasang suatu moter yang dapat mengerek tali jemuran yang terpasang secara otomatis. Kondisi tersebut dapat terjadi jika sesnsor hujan yang sudah terpasang diatas atap yang menutupi sebagian wilayah jemuran tersebut terkena hujan.

### 3. Miniatur *SmartHouse* Tampak Samping Kiri



**Gambar 4.14. Miniatur SmartHouse Tampak Samping Kiri**

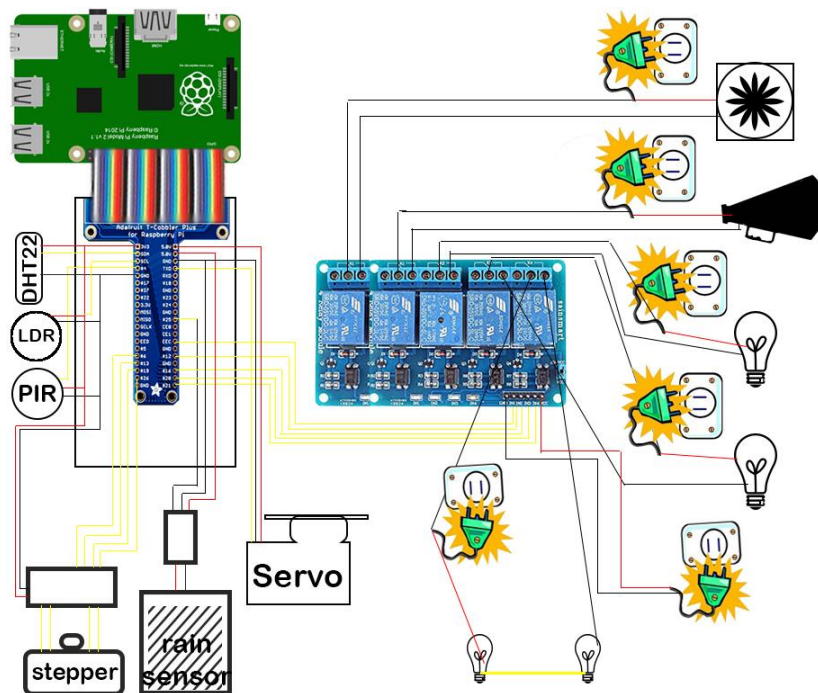
Gambar diatas merupakan miniatur rumah tampak samping sebelah kiri yang digunakan dalam penelitian ini. Dalam gambar tersebut hanya dapat terlihat suatu kipas yang berada dibalik dinding miniatur rumah. Kipas tersebut terhubung dengan sensor suhu (DHT22). Dimana ketika suhu yang ditangkap oleh sesnsor tersebut lebih besar oleh suhu standar yang ada pada database. Maka baling-baling kipas tersebut akan berputar. Dan sebaliknya jika suhu yang ditangkap oleh sensor

tersebut lebih kecil dari suhu standar yang ada pada database, maka baling-baling kipas tidak akan berputar.

#### 4. Isi Bagian Dalam *SmartHouse*

Bagian dalam miniatur rumah tersebut terdapat dua wilayah, yaitu lantai atas dan lantai bawah. Untuk lantai bawah terdapat rangkaian kabel yang saling terhubung antara *Raspberry*, *relay*, lampu, kipas, sirine, sensor suhu, *switch sirine*, dan *servo*. Sedangkan dibagian atas terdapat rangkaian kabel yang saling terhubung antara lampu, lampu balkon, sensor cahaya, pcb, motor, dan sensor hujan.

Berikut *wiring diagram* dari sistem ini:



**Gambar 4.15. Wiring Diagram**

Gambar 4.15 diatas menunjukkan *wiring diagram* pada sistem ini. Dapat dilihat terdapat 4 jenis sensor, yaitu LDR, DHT22, PIR Motion, dan Sensor Hujan.

Dalam gambar tersebut juga terdapat 8 aktuator berupa, kipas, motor *stepper*, 4 lampu, servo, dan sirine.

Raspberry Pi model 3 B ini memiliki 40 pin GPIO, dimana dalam penelitian ini digunakan 14 pin GPIO. Untuk penggunaan daya 3V digunakan oleh servo, pir, dan DHT22. Sementara untuk penggunaan daya 5V digunakan oleh sensor hujan dan motor *stepper*.

#### 4.3.2. Sistem *SmartHouse*

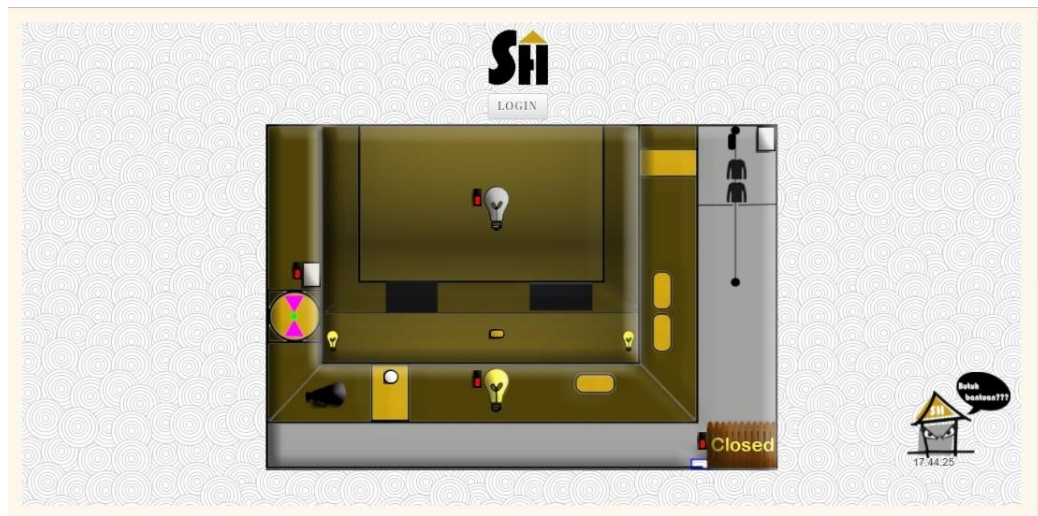
Server terletak pada *Raspberry pi*. Berikut merupakan penjelasan dari beberapa fungsi yang ada pada *server*.

##### 1. Lihat Data

```
<!-- suhu -->
<?php
if (($crud->suhu)>($crud->suhustandar)) {
    ?><div style="margin-left:-45.5%; margin-top:-17.9%;">
    
    </div><?php
}
else{
    ?><div style="margin-left:-45.5%; margin-top:-17.9%;">
    
    </div><?php
}
?>
<!-- end suhu -->
```

**Gambar 4.16. Potongan Kode Liha Data**

Kode diatas berjalan langsung disaat ketika membuka website. Perintah if else dalam kode tersebut dibuat untuk menampilkan animasi bagian-bagian dari miniatur rumah yang telah dibuat. Animasi akan menyesuaikan ketika variabel \$crud->'nilai' sesuai dengan yang kita inginkan atau yang sudah kita atur dalam database.



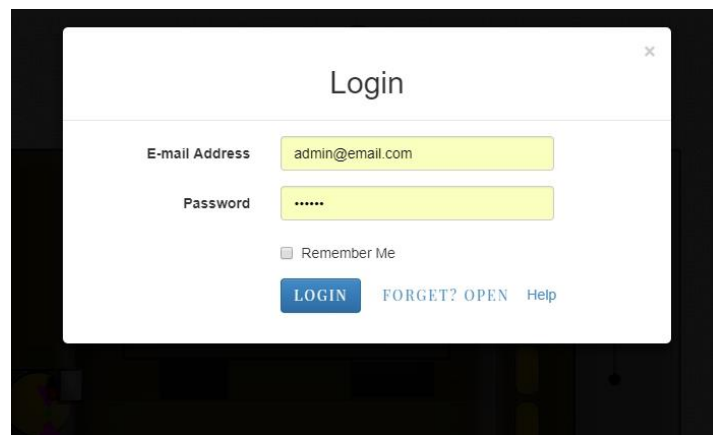
**Gambar 4.17. Tampilan Halaman Web Lihat Data**

## 2. Login

```
<div class="modal fade" id="modallogin" tabindex="-1" role="dialog"
aria-labelledby="ModalLabel">
    <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
    <h2 style="text-align:center;" class="modal-title"
id="ModalLabel">Login</h2>
</div>
<div class="modal-body">
```

**Gambar 4.18. Potongan Kode *Login***

Kode diatas berisi form tampilan *login* berupa sebuah modal yang telah dipanggil dalam fungsi *button login*. Fungsi login tersebut akan berjalan dengan *method POST* yang dikirim ke *route web* dan akan di proses dalam *login controller*.



**Gambar 4.19. Tampilan Halaman Modal *Login***

### 3. Ubah Data

```
<script>
$(document).ready(function() {
  $('.modal-footer').on('click', '.edit', function() {
    $.ajax({
      type: 'post',
      url: 'crud.edit',
      data: {
        '_token': $('input[name=_token]').val(),
        'lampu2': $('#12').val()
      },
      success: function(data) {
        $('.item' + data.id).replaceWith("<tr class='item"
```

**Gambar 4. 20. Potongan Kode Ubah Data**

Kode diatas merupakan kode *javascript* yang berfungsi untuk menampilkan data dari *database* ke dalam sebuah *form* ubah data yang dipanggil melalui fungsi modal. Pada *javascript* tersebut pun mempunyai fungsi update data pada fungsi 'post' yang dikirim melalui route 'crud.edit' ke *CrudController* dan disimpan kedalam *database*



**Gambar 4. 21. Tampilan Halaman Modal Ubah Data**

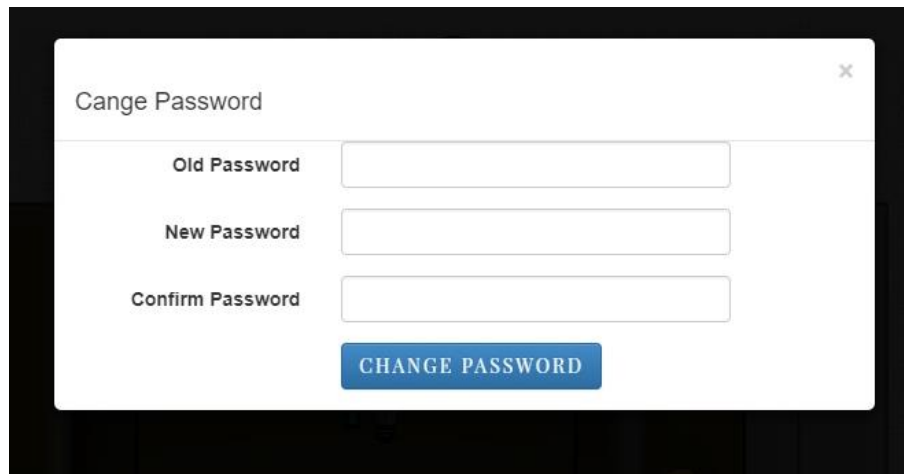
#### 4. Ubah Password

```
<?php
// use App\Session;
use Illuminate\Support\Facades\Input as input;
use App\User;
Route::get('/changepassword', function(){
    Session::flash('flash_message','Password has been changed');
    return view('auth.edit'); });
Route::Post('edit',function(){
    $User = User::find(1);
```

**Gambar 4.22. Potongan Kode Ubah Password**



Kode diatas terdapat pada *web routing*, dimana fungsinya disatukan dalam *web routing* tanpa membuat fungsi didalam file *controller*.



**Gambar 4.23. Tampilan Halaman Modal Ubah Password**

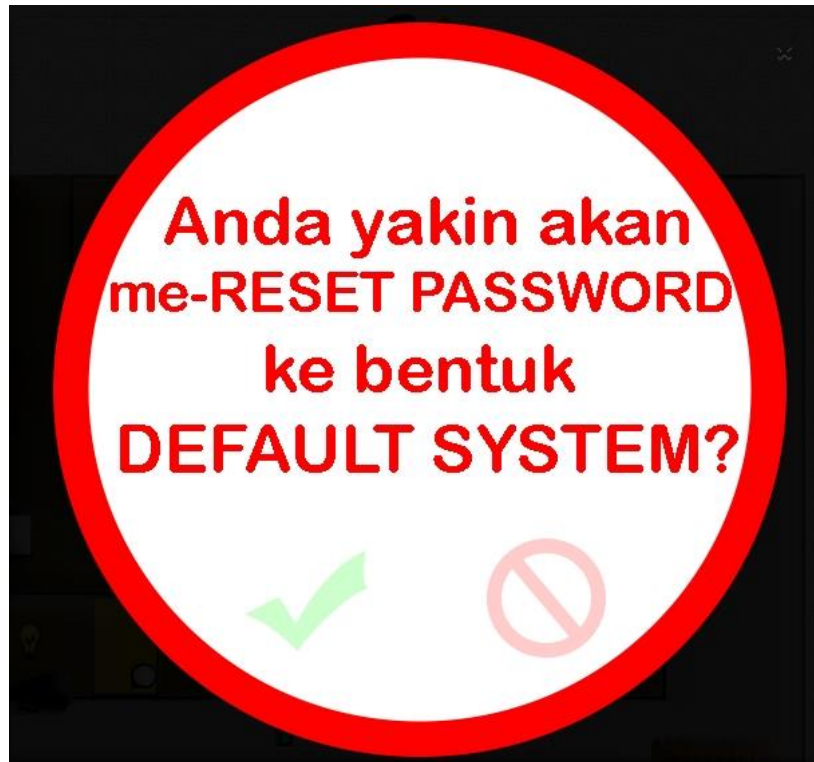
#### 5. *Reset Password*

```
public function reset(Request $request)
{
    $cruds=User::find(1);
    $ubah2=bcrypt('admin1');
    $cruds->password=$ubah2;
    $cruds->update();
    Session::flash('flash_message','Password Has Been Reset');
    return redirect('crud');
}
```

**Gambar 4.24. Potongan Kode *Reset Password***

Kode diatas merupakan kode dari *controller reset password* yang terletak pada file *CrudController*. Dimana fungsi reset tersebut dipanggil melalui *web*

*routing*. Dalam kode diatas terdapat nilai 'bcrypt' yaitu merubah suatu variabel nilai yang sudah ditentukan kedalam suatu bentuk enkripsi.



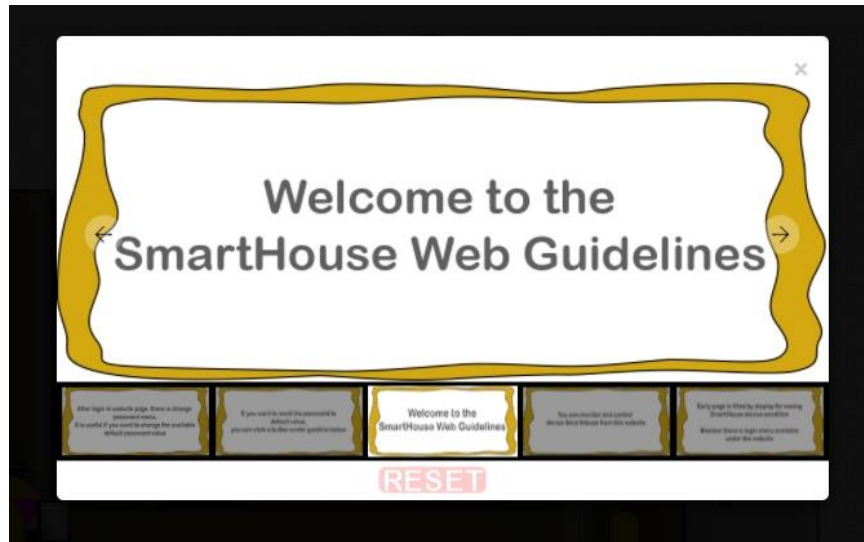
**Gambar 4.25. Tampilan Halaman Modal *Reset Password***

## 6. Bantuan

```
<a href="" type="button" data-toggle="modal" data-
target="#modalbantuan"></a>
<div id="clockDisplay" class="clockStyle" style="position:fixed;
margin-top: 36%; margin-left:86.5%;">
</div>
```

**Gambar 4.26. Potongan Kode Bantuan**

Kode diatas merupakan sebuah kode berupa tombol pemanggil modal yang berisi *slide-slide* yang sudah terdapat bantuan pemakaian website.



**Gambar 4.27. Tampilan Halaman Modal Bantuan**

#### 4.3.3. Sistem *SmartHouse* (Raspberry)

##### 1. Membaca Sensor Gerak (PIR *Motion*)

```
print " Ready"

# Loop until users quits with CTRL-C
while True :

    # Read PIR state
    Current_State = GPIO.input(GPIO_PIR)

    if Current_State==1 and Previous_State==0:
        # PIR is triggered
        print " Motion detected!"
        GPIO.output(21,GPIO.HIGH)
        # Record previous state
        Previous_State=1
    elif Current_State==0 and Previous_State==1:
        # PIR has returned to ready state
        print " Ready"
        GPIO.output(21,GPIO.LOW)
        Previous_State=0
```

**Gambar 4.28. Potongan Kode Membaca Sensor Gerak (PIR *Motion*)**

Fungsi pada kode diatas mempunyai 2 buah parameter, yaitu sensor dan pin. Keduanya digunakan jika jenis sensor atau pin nya kita ingin rubah menyesuaikan sensor dan pin yang dipakai pada penelitian. Jalannya fungsi tersebut adalah ketika sistem mengeluarkan tanda *Ready*, maka sensor siap membaca gerakan. Kemudian ketika sistem mengeluarkan tanda *Motion detected*, artinya sensor menerima atau menangkap suatu gerakan dan dilanjutkan dengan memberikan nilai *HIGH* pada pin yang sudah ditentukan untuk menyalakan sirine.

## 2. Membaca Sensor Suhu (DHT 22)

```
humidity, temperature = dht.read_retry(dht.DHT22, 6)
print('Temp={0:0.1f}*Humidity={1:0.1f}%'.format(temperature, humidity))

if temperature >= 30:
    print('panas')
    GPIO.output(19, GPIO.HIGH)
elif temperature < 30:
    print('dingin')
    GPIO.output(19, GPIO.LOW)
    time.sleep(1)
```

**Gambar 4.29. Potongan Kode Membaca Sensor Suhu (DHT22)**

Potongan kode diatas merupakan bagian dari kode sensor pembaca suhu, dimana pada sensor tersebut terbaca suhu dan kelembapan. Kondisi dalam kode tersebut adalah jika suhu lebih besar dari nilai yang sudah ditentukan, maka nilai pada perangkat kipas akan bernilai *HIGH*. Sebaliknya jika nilai suhu lebih kecil dari nilai yang sudah ditentukan, maka pada perangkat kipas akan bernilai *LOW*.

### 3. Membaca sensor hujan

```

if GPIO.input(18):
    active_time = time.time()
    print "hujan tidak terdeteksi"
    forward(int(5) / 1000.0, int(512))
else:
    active_time = time.time()
    print 'hujan'
    print (active_time)
    backwards(int(5) / 1000.0, int(512))

```

**Gambar 4.30. Potongan Kode Membaca Sensor Hujan**

Potongan kode diatas merupakan bagian dari kode sensor pembaca hujan, dimana sensor tersebut meneripa input sesuai pin yang di berikan. Kondisi yang terjadi pada kode tersebut adalah jika pin GPIO yang sudah ditentukan mendapatkan input dari sensor hujan, maka motor penggerak akan bergerak maju. Dan sebaliknya jika sensor hujan tidak menerima input, maka moteor penggerak akan bergerak mundur.

### 4. Membaca Penggerak Servo

```

if ('database'=On):
    p.ChangeDutyCycle(0.25) # turn towards 90 degree
    print 'Buka'
    time.sleep(1) # sleep 1 second
else
    p.ChangeDutyCycle(5.5) # turn towards 0 degree
    print 'Tutup'
    time.sleep(1) # sleep 1 second

```

**Gambar 4.31. Potongan Kode Penggerak Servo**

Pada potongan kode diatas terdapat nilai database yang menjadi acuan untuk penggerak servo. Dimana kondisi servo bergantung pada nilai database yang ada.

Jika databse bernilai *on*, maka servo berputar  $-90^{\circ}$ . Dan sebaliknya jika nilai database bernilai *off*, maka servo berputar  $90^{\circ}$ .

#### **4.4. Pengujian**

##### **4.4.1. Pengujian Data**

##### **4.4.1.1. Pengujian Sensor Suhu (DHT22)**

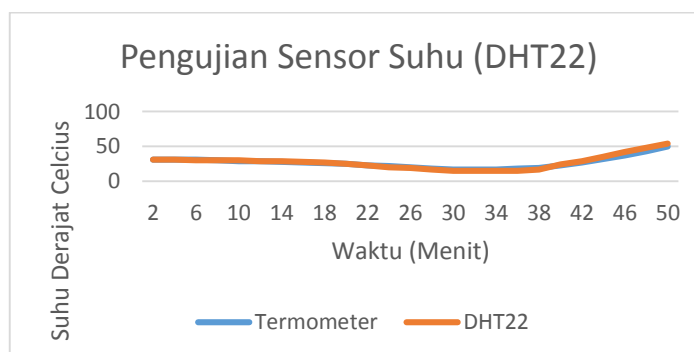


**Gambar 4.32. Pengujian Sensor Suhu (DHT22)**

Pengujian sensor suhu yang dilakukan seperti pada gambar 4.32 merupakan pengujian kepekaan sensor tersebut dengan suhu ruangan yang berbeda yang disimulasikan dengan membuat ruang berukuran kecil yang tinggi rendahnya suhu ruangan tersebut diatur dengan es batu sebagai penurun suhu dan solder sebagai peninggi suhu ruangan yang telah dibuat. Berikut merupakan hasil pengujian yang dilakukan.

**Tabel 4.8.** Hasil Pengujian Sensor Suhu (DHT22)

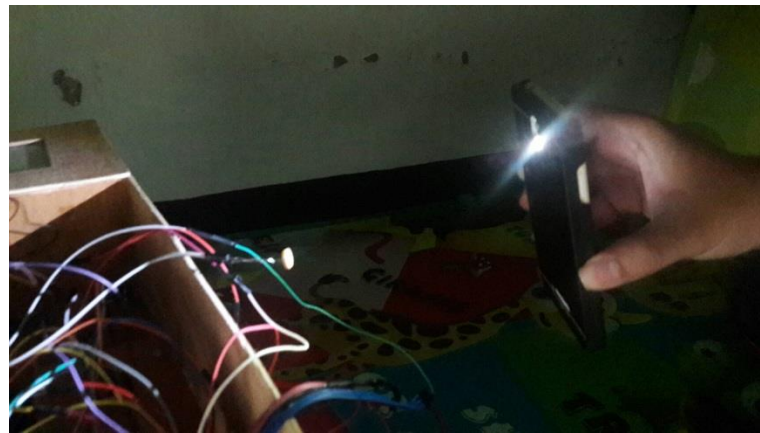
No	Kondisi Pengujian yang Dilakukan	Menit Ke-	Nilai Termometer	Nilai Sensor DHT22	Prosentase Keberhasilan
1	Memasukan es batu kedalam ruang tempat pengujian (Magic Com dalam keadaan power mati)	2	31°	31°	100%
		4	31°	31°	100%
		6	31°	30°	96.7%
		8	30°	30°	100%
		10	29°	30°	96.5%
		12	29°	29°	100%
		14	28°	29°	96.4%
		16	27°	28°	96.2%
		18	26°	27°	96.1%
		20	25°	25°	100%
		22	23°	23°	100%
		24	22°	20°	90.9%
		26	20°	19°	95%
		28	18°	17°	94.4%
		30	17°	15°	88.2%
2	Memanaskan ruang tempat pengujian (Magic Com dalam keadaan power menyala dengan kondisi memasak)	32	17°	15°	88.2%
		34	17°	15°	88.2%
		36	18°	15°	88.2%
		38	19°	17 °	89.4%
		40	23 °	24 °	95.6%
		42	27 °	29 °	92.5%
		44	32 °	35 °	90.6%
		46	37 °	42 °	86.4%
		48	43 °	48 °	88.3%
		50	50 °	54 °	92%
Rata-rata Prosentase Keberhasilan					93.992%

**Gambar 4.33.** Grafik Pengujian Sensor Suhu

Kesimpulan yang didapat :

Nilai sensor suhu yang dihasilkan dari pengujian tersebut jika dibandingkan dengan nilai termometer yang dihasilkan dari pengujian pada ruang dan waktu yang sama dapat ditarik kesimpulan jika nilainya sudah cukup akurat. Hal tersebut dapat dilihat dari sangat dekatnya nilai suhu per dua menit dan rata-rata prosentase keberhasilan yang didapat.

#### **4.4.1.2. Pengujian Sensor Cahaya (LDR)**



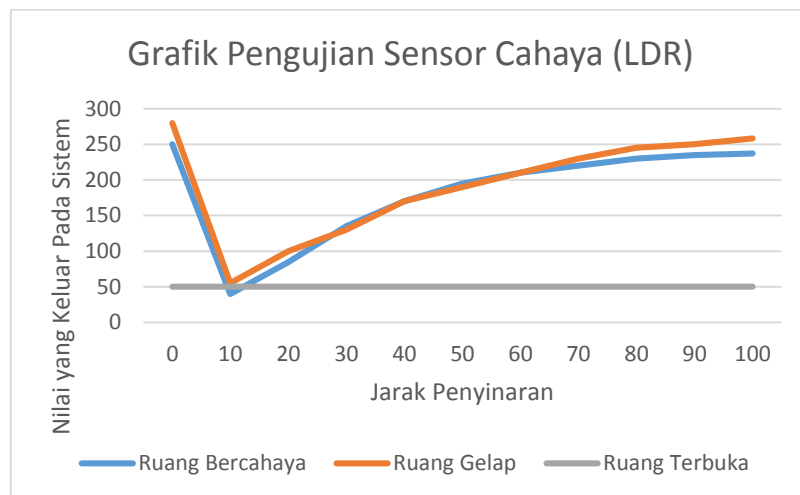
**Gambar 4.34. Pengujian Sensor Cahaya (LDR)**

Pengujian sensor cahaya yang dilakukan seperti pada gambar 4.33 merupakan pengujian kepekaan sensor terhadap cahaya di masing-masing tempat pengujian beserta dengan penyinaran terhadap sensor dengan jarak yang berbeda-beda. Dimana ketika nilai dari sensor cahaya tersebut didapatkan lebih kecil dari nilai standar cahaya pada sistem, maka akan terhubung ke aktuator untuk menyalakan sebuah lampu. Dan jika nilai yang didapatkan lebih besar dari nilai standar cahaya pada sistem, maka akan terhubung ke aktuator untuk tidak menyalakan atau mematikan lampu yang sedang dalam kondisi menyala. Namun karena nilai standar untuk diinputkan pada sistem belum diketahui, maka dalam pengujian ini nilai tersebut akan didapatkan. Berikut merupakan hasil pengujian yang dilakukan.



**Tabel 4.9.** Hasil Pengujian Sensor Cahaya (LDR)

No	Jarak Pemberian Cahaya Senter Handphone	Nilai yang Muncul Pada Sistem Ketika LDR Ditempatkan Diruangan Bercahaya	Nilai yang Muncul Pada Sistem Ketika LDR Ditempatkan Diruangan Gelap	Nilai yang Muncul Pada Sistem Ketika LDR Ditempatkan Diruang Terbuka Dibawah Sinar Matahari
1	-	250	280	50
2	10cm	40	55	50
3	20cm	85	100	50
4	30cm	135	130	50
5	40cm	170	170	50
6	50cm	195	190	50
7	60cm	210	210	50
8	70cm	220	230	50
9	80cm	230	245	50
10	90cm	235	250	50
11	100cm	237	258	50

**Gamba4 4.35. Grafik Pengujian Sensor Cahaya (LDR)**

Kesimpulan yang didapat :

Nilai cahaya yang didapatkan dari sensor LDR berdasarkan pengujian yang telah dilakukan di dua tempat dimana satu tempat dilakukan dengan dua kondisi

(gelap dan terang) dengan masing-masing diberi sebelas jenis kondisi (jarak penyinaran) yang berbeda dapat ditarik kesimpulan bahwa semakin rendah nilai yang keluar pada hasil pengujian, maka semakin terang ruangan atau cahaya yang ditangkap oleh sensor LDR. Dan berdasarkan dari pengujian yang dilakukan, dapat diasumsikan untuk memberikan nilai standar yang akan disimpan dalam database sistem adalah 200. Dimana jika nilai yang keluar dari sistem berdasarkan cahaya yang diterima oleh LDR lebih besar dari nilai standar pada sistem yang sudah diasumsikan, maka kondisi aktuator (lampu) akan menyala. Dan jika nilai yang keluar dari sistem berdasarkan cahaya yang diterima oleh LDR lebih kecil dari nilai standar yang sudah diasumsikan, maka kondisi aktuator (lampu) akan mati jika sebelumnya dalam kondisi menyala.

#### **4.4.1.3. Pengujian Sensor Hujan**



**Gambar 4.36. Pengujian Sensor Hujan**

Pengujian sensor hujan yang dilakukan seperti pada gambar 4.34 merupakan pengujian kepekaan sensor terhadap air dengan berbagai kondisi yang akan diujikan. Hasil yang didapat ketika sensor mendeteksi sebuah kondisi tersebut adalah hujan, maka motor penggerak akan bergara sesuai arah yang ditentukan dalam sistem. Berikut merupakan hasil pengujian yang dilakukan.

**Tabel 4.10.** Pengujian Sensor Hujan

No	Kondisi Pengujian yang Dilakukan	Nilai Sensor Hujan	Kondisi Aktuator (Motor)
1	Menyemprotkan air dengan botol semprot ke seluruh bidang sensor	Tidak terdeteksi hujan	Mati
2	Mengoleskan air menggunakan kain keseluruhan bidang sensor	Terdeteksi hujan	Menyala
3	Meneteskan air menggunakan selang kecil ke bidang sensor	Tidak terdeteksi hujan	Mati
4	Menumpahkan air keseluruhan bidang sensor dengan titik jatuh 3cm	Tidak terdeteksi hujan	Mati
5	Menumpahkan air keseluruhan bidang sensor dengan titik jatuh 15cm	Terdeteksi hujan	Menyala

Kesimpulan yang didapat :

Nilai yang didapat dari pengujian sensor hujan yang telah dilakukan, hanya mendapatkan hasil hujan terdeteksi atau tidak terdeteksi. Dimana terdeteksinya sensor tersebut jika seluruh bidang sensor terkena air dan air tersebut mempunyai massa yang menekan bidang air tersebut.

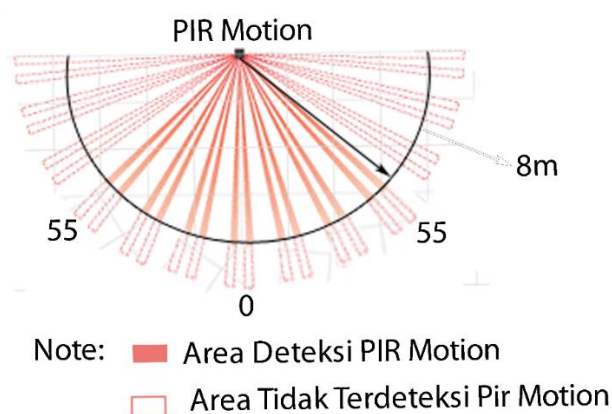
#### 4.4.1.4. Pengujian Sensor Gerak (PIR Motion)

**Gambar 4.37.** Pengujian Sensor Gerak (PIR Motion)

Pengujian sensor gerak yang dilakukan seperti pada gambar 4.35 merupakan pengujian sensor terhadap kepekaan sensor terhadap gerakan manusia didalam area deteksi sensor. Pengujian yang dilakukan adalah dengan melakukan pendataan nilai sensor berdasarkan sudut kemiringan dari titik sensor berada. Jika sensor menerima atau mendeteksi sebuah gerakan, maka alarm atau sirine yang bertugas sebagai aktuator akan menyala. Berikut merupakan hasil pengujian yang dilakukan.

**Tabel 4.11.** Pengujian Sensor Gerak (PIR Motion)

No	Sudut Pengujian	Jarak Pengujian	Delay	Nilai Sensor Gerak	Kondisi Aktuator (Alarm)
1	75°	10m	-	Tidak terdeteksi	Mati
2	75°	9m	-	Tidak terdeteksi	Mati
3	75°	8m	-	Tidak terdeteksi	Mati
4	75°	7m	-	Tidak terdeteksi	Mati
5	75°	4m	-	Tidak terdeteksi	Mati
6	75°	2m	-	Tidak terdeteksi	Mati
7	55°	10m	-	Tidak terdeteksi	Mati
8	55°	9m	-	Tidak terdeteksi	Mati
9	55°	8m	2s	Terdeteksi	Menyala
10	55°	7m	2s	Terdeteksi	Menyala
11	55°	4m	2s	Terdeteksi	Menyala
12	55°	2m	2s	Terdeteksi	Menyala



**Gambar 4.38.** Data Hasil Pengujian Sensor Gerak (PIR Motion)

Kesimpulan yang didapat :

Nilai yang didapat dari pengujian sensor gerak yang telah dilakukan, menghasilkan kesimpulan bahwa jarak efektif yang dapat dijangkau oleh sensor gerak tersebut adalah 8m, dimana dalam cakupan sudut  $55^\circ$  adalah jangkauan efektif dari sensor gerak.

#### 4.4.2. Pengujian *BlackBox*

Pengujian pada sistem yang dibuat dilakukan dengan *blackbox testing*.

Berikut merupakan hasil pengujian yang dilakukan.

**Tabel 4.12.** Hasil Pengujian *BlackBox*

Kode Uji	Kode SR	Aksi	Keluaran Yang Diinginkan	Keluaran	Kesimpulan
U-1	SR-SH-1	Menyalakan Raspberry	Data muncul dari database dan sensor	Data muncul dari database dan sensor	Memenuhi
U-2	SR-SH-2	Menyalakan Raspberry dengan program otomatis	Sensor dan aktuator berjalan semestinya	Sensor dan aktuator berjalan semestinya	Memenuhi
U-3	SR-SH-3	Memasukkan alamat web	Keluar halaman website	Keluar halaman website	Memenuhi
U-4	SR-SH-4	Menekan tombol login	Muncul modal form login	Muncul form modal login	Memenuhi
U-5	SR-SH-5	Mengisi form login dan melakukan login	Muncul halaman lihat dan ubah data	Muncul halaman lihat dan ubah data	Memenuhi
U-6	SR-SH-6	Menekan ikon/gambar switch	Muncul modal form ubah data	Muncul modal form ubah data	Memenuhi
U-7	SR-SH-7	Mengisi form ubah data dan	Kondisi data berubah	Kondisi data berubah	Memenuhi

		melakukan proses ubah data	sesuai nilai yang dimasukkan	sesuai nilai yang dimasukkan	
U-8	SR-SH-8	Mengisi form ubah password dan melakukan proses ubah password	Nilai password berubah sesuai nilai yang diinputkan	Nilai password berubah sesuai nilai yang diinputkan	Memenuhi
U-9	SR-SH-9	Menekan ikon animasi bantuan	Muncul modal berisi panduan website	Muncul modal berisi panduan website	Memenuhi
U-10	SR-SH-10	Menekan tombol reset password yang ada pada modal bantuan	Muncul notifikasi lanjutan atau batalkan reset password	Muncul notifikasi lanjutan atau batalkan reset password	Memenuhi
U-11	SR-SH-11	Menekan tombol logout	Berpindah halaman menuju halaman lihat data	Berpindah halaman menuju halaman lihat data	Memenuhi

#### 4.5. Pemeliharaan dan Pengembangan

Tahap pemeliharaan dan pengembangan sistem ini akan diperbanyak alat yang akan di pantau dan dikontrol, serta menambah sistem keamanan dengan adanya kamera cctv.

## 5. Penutup

### 5.1. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan, didapatkan beberapa hasil kesimpulan sebagai berikut.

1. Sistem *SmartHouse* dengan fitur memonitor suhu, cahaya, hujan dan mengaturnya dengan *interface* web dan dapat di operasionalkan dari jauh telah berhasil dirancang dan dibuat.
2. *Interface* yang digunakan adalah sebuah web, dimana web tersebut dibangun menggunakan *framework Laravel*.
3. Berdasarkan data yang diperoleh dari tahap pengujian yang ada pada bab pengujian, yang dihasilkan:
  1. nilai yang dihasilkan untuk rata-rata prosentase keberhasilan pengujian sensor suhu adalah 93.992%
  2. pengujian nilai ldr yaitu semakin rendah nilai yang didapat sensor, maka semakin terang kondisi ruangan tersebut. Dan semakin tinggi nilai yang didapat sensor, maka semakin gelap kondisi ruangan tersebut.
  3. Sensitifitas sensor hujan dipengaruhi oleh massa air yang jatuh ke penampang sensor hujan.
  4. Jarak atau wilayah cakupan sensor gerak yaitu 8m dengan tingkat derajat 55 dari titik 0.

## 5.2. Saran

Berikut ini beberapa saran untuk pengembangan sistem selanjutnya adalah sebagai berikut.

1. Mengembangkan *interface* agar lebih interaktif.
2. Membuat remot pada kendaraan untuk membuka gerbang.
3. Menambahkan cctv dan menambahkan kamera *real time* yang bisa dipantau melalui web.



