



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - IF184802

# Perancangan dan Implementasi Prototipe Sistem *Rumah Cerdas*

Mohammad Haekal Alawy  
05111640000141

Dosen Pembimbing I  
Ir. Muchammad Husni, M.Kom.

Dosen Pembimbing II  
Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D.

DEPARTEMEN INFORMATIKA  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2020

***[Halaman ini sengaja dikosongkan]***



**TUGAS AKHIR - IF184802**

# **Perancangan dan Implementasi Prototipe Sistem Rumah Cerdas**

**MOHAMMAD HAEKAL ALAWY**  
05111640000141

Dosen Pembimbing I  
Ir. Muchammad Husni, M.Kom.

Dosen Pembimbing II  
Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D.

**DEPARTEMEN INFORMATIKA**  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2020

***[Halaman ini sengaja dikosongkan]***



**FINAL PROJECT - IF184802**

# **DESIGN AND IMPLEMENTATION OF SMART HOME PROTOTYPE**

**MOHAMMAD HAEKAL ALAWY**  
05111640000141

**Supervisor I**  
Ir. Muchammad Husni, M.Kom.

**Supervisor II**  
Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D.

**DEPARTMENT OF INFORMATICS**  
Faculty of Intelligent Electrical and Informatics Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2020

***[Halaman ini sengaja dikosongkan]***

# LEMBAR PENGESAHAN

**Perancangan dan Implementasi Prototipe Sistem *Smart Home***

## TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Bidang Studi Arsitektur dan Jaringan  
Program Studi S-1 Informatika  
Departemen Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh:

**Mohammad Haekal Alawy**  
**NRP: 051116 40000 141**

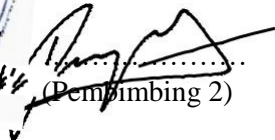
Disetujui oleh Dosen Pembimbing Tugas Akhir:

Ir. Muchammad Husni, M.Kom.  
(NIP. 196002211984031001)



.....  
(Pembimbing 1)

Royyana Muslim Ijtihadie, S.Kom., M.Kom.,  
Ph.D (NIP. 197708242006041001)



.....  
(Pembimbing 2)

**SURABAYA**  
**JUNI 2020**

***[Halaman ini sengaja dikosongkan]***



## **Perancangan dan Implementasi Prototype Sistem Smart Home**

Nama Mahasiswa : Mohammad Haekal Alawy  
NRP : 051116 40000 141  
Departemen : Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan  
Informatika Cerdas – ITS  
Dosen Pembimbing 1 : Ir. Muchammad Husni, M.Kom.  
Dosen Pembimbing 2 : Royyana Muslim Ijtihadie, S.Kom.,  
M.kom., Ph.D.

### ***Abstrak***

Permasalahan dalam tugas akhir ini diambil dari permasalahan yang terdapat di dunia nyata, persoalan yang akan diselesaikan pada penelitian tugas akhir ini berkaitan dengan keperluan pada rumah demi meningkatkan keamanan dan fungsionalitas tertentu. Metode yang digunakan untuk menyelesaikan persoalan pada tugas akhir ini adalah pemanfaatan teknologi IoT

Permasalahan yang akan diselesaikan yaitu permasalahan jemuran agar bisa otomatis tertutup jika sensor mendeteksi keadaan sekitar tidak mendukung untuk menjemur pakaian, Lampu dan kipas rumah yang menyesuaikan kondisi di sekitar rumah dan juga pintu yang dapat diakses tanpa menggunakan kunci. Teknologi IoT yang digunakan adalah sensor-sensor (*raindrop sensor*, sensor suhu dan kelembapan, sensor cahaya), *solenoid door lock* dan juga *motor servo* yang nantinya akan dirancang dalam bentuk prototipe Sistem Rumah Cerdas. Dilakukan uji coba pada hasil implementasi alat-alat tersebut dan alat-alat yang diuji meningkatkan keamanan dan fungsionalitas tertentu dari suatu rumah.

Prototipe Sistem Rumah Cerdas diimplementasikan dengan menggunakan bahasa pemrograman C yang di compile melalui Arduino IDE dan menggunakan sistem database MySQL, dari uji coba yang telah dilakukan, diperoleh kesimpulan bahwa

prototipe Sistem Rumah Cerdas telah berjalan dan terhubung dengan *WiFi* dan Website monitoring.

***Kata kunci: Teknologi IoT, MySQL, Bahasa Pemograman C, Arduino IDE***

## DESIGN AND IMPLEMENTATION OF SMART HOME

Student Name : Mohammad Haekal Alawy  
Registration Number : 051116 40000 141  
Department : Department of Informatics  
Faculty of Intelligent Electrical and  
Informatics Technology – ITS  
First Supervisor : Ir. Muchammad Husni, M.Kom.  
Second Supervisor : Royyana Muslim Ijtihadie, S.Kom.,  
M.Kom., Ph.D.

### *Abstract*

The problems in this final project are taken from problems that exist in the real world, problems that will be resolved in this final project research related to the needs of the home in order to improve certain security and functionality. The method used to solve the problem in this thesis is the use of IoT technology

The problem that will be solved is the problem of clothesline so that it can be automatically closed if the sensor detects the surrounding conditions does not support drying clothes, lights and home fans that adjust the conditions around the house and also doors that can be accessed without using a key. IoT technology used is sensors (raindrop sensors, temperature and humidity sensors, light sensors), door lock solenoids and servo motors which will be designed in the form of prototypes of the Smart Home System. Trials are carried out on the results of the implementation of these tools and the tools tested improve the security and certain functionality of a house.

The Smart Home System prototype is implemented using the C programming language compiled through the Arduino IDE and using the MySQL database system, from the trials that have been carried out, it is concluded that the prototype of the smart

home system is already running and connected to WiFi and Website monitoring.

**Keywords: IoT Technology, MySQL, C Programming Language, Arduino IDE**

## **KATA PENGANTAR**

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

### **PERANCANGAN DAN IMPLEMENTASI PROTOTYPE SISTEM RUMAH CERDAS**

Pengerjaan Tugas Akhir ini dilakukan untuk memenuhi salah satu syarat meraih gelar Sarjana di Departemen Informatika Fakultas Teknologi Informasi dan Komunikasi Institut Teknologi Sepuluh Nopember.

Dengan selesainya Tugas Akhir ini diharapkan apa yang telah dikerjakan penulis dapat memberikan manfaat bagi perkembangan ilmu pengetahuan terutama di bidang teknologi informasi serta bagi orang lain dan diri penulis sendiri.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung selama penulis mengerjakan Tugas Akhir maupun selama menempuh masa studi antara lain:

1. Terimakasih kepada Allah SWT, di mana penulis masih diberi kesempatan, kesehatan dan umur untuk menempuh kuliah disini dan menjalani hidup dengan baik sehingga mampu menyelesaikan Tugas Akhir ini.
2. Ibu Lili Wijayanti dan Bapak Lukman Nul Hakim selaku orang tua dari penulis yang selalu memberikan dukungan penuh kepada penulis sehingga dapat menyelesaikan tugas akhir ini
3. Bapak Ir. Muchammad Husni, M.Kom. selaku Dosen Pembimbing yang telah membimbing penulis selama masa-masa perkuliahan dan selama proses penyelesaian Tugas Akhir ini.
4. Bapak Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D. Selaku Dosen Pembimbing yang telah memberikan ilmu, dan masukan kepada penulis.

5. Teman Teman di grup discord Darari Garis Keras yang selalu membantu dan menemani penulis dalam pengerjaan tugas akhir dan penyusunan buku.
6. Anindya Gayatri Puspitasari, selaku teman dekat yang selalu menemani, membantu dan memberikan dukungan kepada penulis dalam penyusunan buku Tugas Akhir ini.
7. Teman-teman kontrakan penulis yang membantu menyelesaikan pengerjaan tugas akhir
8. Teman-teman angkatan 2016 jurusan Teknik Informatika ITS yang telah menemani perjuangan penulis selama masa perkuliahan.
9. Serta pihak-pihak lain yang tidak dapat disebutkan di sini yang telah banyak membantu penulis dalam penyusunan Tugas Akhir ini.

Penulis mohon maaf apabila masih ada kekurangan pada Tugas Akhir ini. Penulis juga mengharapkan kritik dan saran yang membangun untuk pembelajaran dan perbaikan di kemudian hari. Semoga melalui Tugas Akhir ini penulis dapat memberikan kontribusi dan manfaat yang sebaik-baiknya.

Surabaya, Juni 2020

Mohammad Haekal Alawy

# DAFTAR ISI

LEMBAR PENGESAHAN.....	v
<i>Abstrak</i> .....	vii
<i>Abstract</i> .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL.....	xix
<b>1 BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
<b>Sistem yang dibuat untuk Sistem Rumah Cerdas ini dibuat menggunakan <i>localhost</i> yang berarti menjadikan</b> .....	<b>2</b>
1.2 Rumusan Permasalahan .....	2
1.3 Batasan Permasalahan.....	2
1.4 Tujuan Pembuatan Tugas Akhir.....	3
1.5 Manfaat Tugas Akhir .....	3
1.6 Metodologi .....	3
1.6.1 Penyusunan Proposal Tugas Akhir.....	4
1.6.2 Studi Literatur.....	4
1.6.3 Implementasi Perangkat Lunak .....	4
1.6.4 Pengujian dan Evaluasi.....	4
1.6.5 Penyusunan Buku Tugas Akhir .....	5
1.7 Sistematika Penulisan .....	5
<b>2 BAB II TINJAUAN PUSTAKA</b> .....	<b>7</b>
2.1 Penelitian Terkait .....	7
2.1.1 Teknologi <i>Smart Home</i> Dan Manfaatnya.....	7
2.1.2 Perancangan <i>Smart Home</i> dengan Konsep <i>Internet of Things (IoT)</i> berbasis <i>Smartphone</i> .....	8
<b>Pada penelitian “Perancangan <i>Smart Home</i> dengan konsep <i>Internet of Things (IoT)</i> berbasis <i>Smartphone</i>” oleh Herdianto memiliki fitur untuk pengendalian dan pemantauan peralatan listrik dirumah apakah sudah dalam kondisi hidup (on) atau mati (off). Setelah diaktifkan melalui media handphone menggunakan jaringan internet. Untuk</b>	

**membangun *Smart Home* ini terdiri 2 bagian yaitu perangkat keras dan perangkat lunak. Perangkat keras digunakan sebagai media interface antara komputer (arduino) dengan peralatan listrik yang dikendalikan sedangkan perangkat lunaknya digunakan untuk mengaktifkan perangkat keras dan komunikasi antara arduino dan *smartphone*.....8**

2.1.3 Rancang Bangun Sistem Jemuran Pakaian Otomatis Menggunakan Metode *Fuzzy Logic*.....8

2.1.4 *Door Lock* Berbasis *Internet of Things (IoT)*.....9

2.2 Hardware Tools .....9

2.2.1 WeMos D1 R1 .....9

2.2.2 Motor Servo SG90.....10

2.2.3 Sensor Kelembapan DHT11 .....10

2.2.4 *Jumper Cable*.....11

2.2.5 *Solenoid Door Lock* .....11

2.2.6 *Light Dependent Resistor* .....12

2.2.7 4 Channel Relay 5V.....13

2.2.8 Raindrop Sensor.....14

2.3 *Software Tools*.....14

2.3.1 Arduino IDE .....14

2.3.2 XAMPP .....15

2.3.3 CodeIgniter 3 .....15

2.3.4 Visual Studio Code .....15

2.3.5 SQLYog.....15

**3.....17**

**4 BAB III PERANCANGAN .....17**

4.1 Perancangan Sistem.....17

4.2 Perancangan *Database* .....24

4.3 Perancangan Alat.....28

4.4 Perancangan Sistem Monitor .....30

**5 BAB IV IMPLEMENTASI.....33**

5.1 Implementasi *Database* .....33

5.2 Implementasi Alat .....35

5.2.1 Rangkaian Alat Sistem Rumah Cerdas .....35

5.3 Implementasi Web.....53



5.3.1	Manage User Oleh Admin .....	54
5.3.2	<i>Monitoring</i> dan <i>Controlling</i> Perangkat.....	57
5.3.3	Deteksi Sensor .....	58
5.3.4	Kode Verifikasi Akses Pintu .....	59
<b>6</b>	<b>BAB V UJICOBA DAN EVALUASI.....</b>	<b>61</b>
6.1	Lingkup Uji Coba.....	61
6.2	Skenario Pengujian .....	61
6.2.1	Monitoring dan Controlling Sistem Pintu .....	62
6.2.2	Monitoring dan Controlling Sistem Lampu.....	66
6.2.3	Monitoring dan Controlling Sistem Kipas.....	70
6.2.4	Monitoring Sistem Jemuran.....	73
<b>7</b>	<b>BAB VI KESIMPULAN .....</b>	<b>83</b>
7.1	Kesimpulan .....	83
7.2	Saran .....	83
	<b>DAFTAR PUSTAKA .....</b>	<b>85</b>
1.	Kode Program SQL.....	87
2.	Kode Program Admin.php .....	88
3.	Kode Program Controll.php .....	91
4.	Kode Program Jemuran.php.....	94
5.	Kode Program Kipas.php.....	95
6.	Kode Program Lampu.php.....	97
7.	Kode Program Log_pintu.php.....	99
8.	Kode Program Login.php.....	100
9.	Kode Program Sensor.php .....	101
10.	Kode Program Arduino.....	103
	<b>BIODATA PENULIS.....</b>	<b>115</b>

***[Halaman ini sengaja dikosongkan]***

## DAFTAR GAMBAR

Gambar 2.1 Wemos D1 R1 .....	9
Gambar 2.2 Motor Servo G90.....	10
Gambar 2.3 Sensor DHT11 .....	11
Gambar 2.4 Jumper Cable.....	11
Gambar 2.5 Solenoid Door Lock .....	12
Gambar 2.6 Light Dependant Resistor.....	13
Gambar 2.7 4 Channel Relay 5 .....	13
Gambar 2.8 Raindrop Sensor .....	14
Gambar 3.1 Rancangan Jaringan Sistem Rumah Cerdas ..	18
Gambar 3.2 Flowchart Sistem Rumah Cerdas .....	19
Gambar 3.3 Flowchart Alur Mengakses Pintu .....	20
Gambar 3.4 Flowchart Alur Kerja Kipas .....	21
Gambar 3.5 Flowchart Alur Sistem Lampu .....	22
Gambar 3.6 Flowchart Alur Monitoring Jemuran.....	23
Gambar 3.7 Hasil Eksekusi Query Tabel Pintu.....	25
Gambar 3.8 Hasil Eksekusi Query Tabel Jemuran .....	26
Gambar 3.9 Hasil Eksekusi Query Tabel Log_pintu .....	27
Gambar 3.10 Hasil Eksekusi Query Tabel Lampu.....	27
Gambar 3.11 Hasil Eksekusi Query Tabel Kipas.....	28
Gambar 3.12 Use Case Diagram Admin.....	30
Gambar 3.13 Use Case Diagram Member .....	31
Gambar 4.1 Kode Sumber Tabel User .....	33
Gambar 4.2 Kode Sumber Tabel Pintu .....	34
Gambar 4.3 Kode Sumber Tabel Kipas .....	34
Gambar 4.4 Kode Sumber Tabel Jemuran .....	34
Gambar 4.5 Kode Sumber Tabel Lampu .....	35
Gambar 4.6 Kode Sumber Tabel Log_Pintu.....	35
Gambar 4.7 Include Libray Arduino IDE .....	36
Gambar 4.8 Fungsi Connect Wifi Pada Microcontroller ..	37
Gambar 4.9 Kode Sumber Integrasi Seluruh Sistem.....	38
Gambar 4.10 Rangkaian Sistem Pintu .....	39
Gambar 4.11 Fungsi Delete User .....	56
Gambar 5.1 Menu Monitoring Sistem Pintu .....	62
Gambar 5.2 Masukan Kode Verifikasi.....	63
Gambar 5.3 Pengiriman Kode Verifikasi Ke Email.....	64

Gambar 5.4 Status Pintu Terbuka .....	64
Gambar 5.5 Status Pintu Terbuka Pada Web .....	65
Gambar 5.6 Status Pintu Terkunci .....	65
Gambar 5.7 Menampilakn informasi Log Pintu.....	66
Gambar 5.8 Menampilkan Status Informasi Lampu .....	67
Gambar 5.9 Serial Monitor Status Lampu On.....	68
Gambar 5.10 Tampilan Informasi Status Sensor On.....	69
Gambar 5.11 Sensor On Lampu On .....	69
Gambar 5.12 Sensor On Lampu Off .....	69
Gambar 5.13 Menampilkan Informasi Status Kipas Off...	70
Gambar 5.14 Status Kipas Aktif .....	71
Gambar 5.15 Status Kipas On .....	71
Gambar 5.16 Status Sensor Kipas Aktif.....	72
Gambar 5.17 Sensor Aktif Kipas Aktif .....	73
Gambar 5.18 Menampilkan Status Jemuran.....	74
Gambar 5.19 Kondisi Gelap	<b>Error! Bookmark not</b>

**defined.**

Gambar 5.20 Kondisi Terang Tidak Hujan Dan Tidak Lembap.....	<b>Error! Bookmark not defined.</b>
Gambar 5.21 Kondisi Sedang Hujan	<b>Error! Bookmark not</b>

**defined.**

Gambar 5.22 Kondisi Jemuran Sedang Tidak Dijemur.....	76
Gambar 5.23 Prototipe Perangkat Aktif .....	82
Gambar 5.24 Prototipe Perangkat NonAktif .....	82

## DAFTAR TABEL

Tabel 3.1 Daftar Tabel .....	24
Tabel 3.2 Detail Tabel Pintu .....	24
Tabel 3.3 Detail Tabel Jemuran .....	25
Tabel 3.4 Detail Tabel Log_pintu .....	26
Tabel 3.5 Detail Tabel Lampu .....	27
Tabel 3.6 Detail Tabel Kipas .....	28
Tabel 4.1 Rangkaian Sistem Kipas .....	41
Tabel 4.2 Rangkaian Sistem Lampu .....	44
Tabel 4.3 Rangkaian Sistem Jemuran .....	47
Tabel 5.1 Tabel Spesifikasi Laptop.....	61
Tabel 5.2 Hasil Uji Skenario Monitoring Sistem Pintu ....	63
Tabel 5.3 Hasil Uji Skenario Controlling Pintu .....	65
Tabel 5.4 Hasil Uji Skenario Menampilkan Log Pintu .....	66
Tabel 5.5 Hasil Uji Skenario Monitoring Lampu .....	67
Tabel 5.6 Hasil Uji Skenario Controlling Lampu .....	68
Tabel 5.7 Hasil Uji Skenario Deteksi Sensor .....	70
Tabel 5.8 Hasil Uji Skenario Monitoring Kipas .....	71
Tabel 5.9 Hasil Uji Skenario Controlling Kipas .....	72
Tabel 5.10 Hasil Uji Skenario Deteksi Sensor Sistem Kipas .....	73
Tabel 5.11 Hasil Uji Skenario Monitoring Sistem Jemuran .....	74
Tabel 5.12 Hasil Uji Skenario Deteksi Sensor Sistem Jemuran .....	76
Tabel 5.13 Uji Respon Sistem Pintu .....	77
Tabel 5.14 Uji Respon Sistem Lampu .....	78
Tabel 5.15 Uji Respon Sistem Kipas .....	79
Tabel 5.16 Uji Respon Sistem Jemuran .....	80

***[Halaman ini sengaja dikosongkan]***

# **BAB I**

## **PENDAHULUAN**

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan masalah, batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan buku Tugas Akhir ini.

### **1.1 Latar Belakang**

*Smart Home System* atau Sistem Rumah Cerdas dapat diartikan sebagai rumah yang dilengkapi dengan perangkat teknologi canggih. Seluruh perangkat yang terpasang pada rumah dapat saling terhubung sehingga pemilik rumah dapat mengendalikan perangkat(perengkapan maupun peralatan) yang berada dirumah secara remote maupun secara otomatis.

Pada rumah biasa, tentunya penghuni rumah tidak dapat mengendalikan perlengkapan atau peralatan yang berada dirumah secara otomatis, misalnya saja jika penghuni rumah lupa untuk mengunci pintu rumah tentunya hal tersebut bisa menyebabkan masalah keamanan pada rumah tersebut. Atau jemuran rumah ketika hujan turun namun penghuni rumah sedang tidak berada dirumah maka hal tersebut juga bisa menjadi suatu masalah bagi penghuni rumah. Dan perlengkapan rumah sederhana misalnya saja lampu jika hari sudah gelap namun tidak ada penghuni rumah sehingga rumah terlihat gelap karena tidak ada penghuni yang menyalakan lampu.

Maka dari itu, diperlukan Sistem Rumah Cerdas yang dapat mengendalikan perlengkapan secara otomatis dan dapat dilakukan secara remote untuk membantu penghuni rumah mengatasi permasalahan-permasalahan yang telah disebutkan tadi.

## **Sistem yang dibuat untuk Sistem Rumah Cerdas ini dibuat menggunakan *localhost* yang berarti menjadikan**

komputer server lokal dari jaringan itu sendiri. Sebagai catatan yang dapat mengakses *localhost* hanyalah jaringan yang terhubung dengan *localhost*. Untuk pengimplementasian pada kehidupan nyata untuk menerapkan prototipe Sistem Rumah Cerdas ini dibuat menggunakan *cloud server* yang mampu berjalan 24 jam sehingga sistem ini mampu diakses kapanpun dan dimanapun.

### **1.2 Rumusan Permasalahan**

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun sistem pintu rumah *keyless* dan dapat diakses secara *remote*
2. Bagaimana merancang dan membangun sistem jemuran otomatis berdasarkan kondisi lingkungan sekitar rumah
3. Bagaimana merancang dan membangun sistem lampu dan kipas/AC berdasarkan suhu dan intensitas cahaya di sekitar rumah
4. Bagaimana merancang dan membangun sistem *monitoring* dan *controlling* untuk sistem smart home agar perangkat-perangkat yang bersangkutan saling terhubung?

### **1.3 Batasan Permasalahan**

Permasalahan pada Tugas Akhir ini, memiliki batasan masalah sebagai berikut:

1. Perangkat sensor yang digunakan masih memiliki keterbatasan tingkat akurasi untuk melakukan pendeteksian kondisi sekitar.



2. Sistem monitoring menggunakan *localhost* yang berarti menggunakan server lokal untuk menjalankan sistem web
3. Sistem Rumah Cerdas ini meliputi fitur jemuran otomatis, sistem penguncian pintu rumah melalui web dan *monitoring* dan *controlling* lampu dan kipas

#### **1.4 Tujuan Pembuatan Tugas Akhir**

1. Mampu mempermudah penghuni rumah dalam melakukan *monitoring* dan *controlling* perangkat yang terhubung dengan Sistem Rumah Cerdas
2. Mampu mempermudah penghuni rumah mengakses pintu ketika terjadi kelalaian penghuni rumah lupa membawa kunci.
3. Mampu mempermudah penghuni rumah melakukan *monitoring* dan *controlling* lampu dan kipas ketika penghuni sedang meninggalkan rumah
4. Mampu mempermudah penghuni rumah mengurus jemuran ketika penghuni rumah sedang tidak berada di rumah

#### **1.5 Manfaat Tugas Akhir**

Manfaat yang didapat dari penelitian ini, yaitu:

1. Sistem yang dibangun dapat membantu penghuni rumah meningkatkan keamanan rumah.
2. Sistem yang dibangun mampu membantu penghuni rumah dalam monitoring dan controlling perlengkapan dan peralatan rumah yang terkait dengan penelitian ini.

#### **1.6 Metodologi**

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

### **1.6.1 Penyusunan Proposal Tugas Akhir**

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang dibuat pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat batasan masalah untuk tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pemebuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan pengerjaan tugas akhir

### **1.6.2 Studi Literatur**

Pada studi literatur ini. Akan dipelajari beberapa jumlah referensi yang akan diperlukan untuk pengerjaan Tugas Akhir.

### **1.6.3 Implementasi Perangkat Lunak**

Membuat perangkat berupa prototype dan merancang sistem berdasarkan data yang sudah didapatkan dari sstudi literatur dan pengumpulan data yang sudah dilakukan. Tahap ini adalah bentuk awal dari perancangan yang nantinya akan diimplementasikan.

### **1.6.4 Pengujian dan Evaluasi**

Pengujian dilakukan melalui dua cara yaitu :

#### **1. Pengujian hardware**

Pengujian *hardware* adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat keras. Pengujian ini dilakukan untuk menguji apakah sistem Smart Home berjalan dengan baik atau tidak.

#### **2. Pengujian software**

Pengujian *software* adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak. Pengujian ini dilakukan untuk menguji apakah perangkat lunak sistem Smart Home berjalan dengan

baik atau tidak atau tidak.

### **1.6.5 Penyusunan Buku Tugas Akhir**

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain.

### **1.7 Sistematika Penulisan**

Buku Tugas Akhir ini merupakan laporan secara lengkap mengenai Tugas Akhir yang telah dikerjakan baik dari sisi teori, rancangan, maupun implementasi sehingga memudahkan bagi pembaca dan juga pihak yang ingin mengembangkannya lebih lanjut. Sistematika penulisan buku Tugas Akhir secara garis besar antara lain:

#### **Bab I Pendahuluan**

Bab ini berisi penjelasan latar belakang, rumusan masalah, batasan masalah dan tujuan pembuatan Tugas Akhir. Selain itu, metodologi pengerjaan dan sistematika penulisan laporan Tugas Akhir juga dijelaskan di dalamnya.

#### **Bab II Tinjauan Pustaka**

Pada bagian ini berisi kajian teori atau penjelasan dari metode, algoritma, *library*, dan *tools* yang digunakan dalam penyusunan Tugas Akhir ini.

#### **Bab III Perancangan Sistem**

Bab ini berisi penjelasan tentang rancangan dari sistem yang akan dibangun.

#### **Bab IV Implementasi**

Pada bagian ini menjelaskan implementasi yang berbentuk kode program dari proses modifikasi, proses pengujian, serta hardware dan prototipe.

**Bab V Pengujian dan Evaluasi**

Bab ini berisi penjelasan mengenai data hasil percobaan dan pembahasan mengenai hasil percobaan yang telah dilakukan.

**Bab VI Kesimpulan dan Saran**

Bab ini merupakan bab terakhir yang menjelaskan kesimpulan dari hasil ujicoba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini berisi penjelasan mengenai teori-teori dasar yang berkaitan dengan pengimplementasian perangkat lunak dan perangkat keras serta penunjangnya. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap *Internet of Things*, tools, serta definisi yang digunakan dalam pembuatan Tugas Akhir.

#### **2.1 Penelitian Terkait**

##### **2.1.1 Teknologi *Smarthome* Dan Manfaatnya**

*SmartHome* (rumah cerdas) adalah rumah yang menyediakan kenyamanan, keamanan, efisiensi energi bagi rumah setiap saat. Teknologi ini dapat bekerja saat orang ada di rumah maupun tidak ada di rumah. *Smarthome* atau rumah cerdas adalah istilah yang biasa digunakan untuk menentukan tempat tinggal yang memiliki peralatan, pencahayaan, pemanas, pendingin ruangan, TV, komputer, sistem audio dan video hiburan, keamanan, dan sistem kamera yang mampu berkomunikasi satu sama lain dan dapat menjadi dikendalikan jarak jauh dengan jadwal waktu, dari setiap ruangan di rumah, serta dari jarak jauh dari lokasi manapun melalui *smartphone* atau internet. Dengan adanya teknologi *smarthome* atau rumah cerdas ini memungkinkan pemilik rumah untuk mengendalikan kenyamanan dan keamanan rumahnya dengan menggunakan satu alat saja

### **2.1.2 Perancangan *Smart Home* dengan Konsep *Internet of Things (Iot)* berbasis *Smartphone***

Pada penelitian “Perancangan *Smart Home* dengan konsep *Internet of Things (IoT)* berbasis *Smartphone*” oleh Herdianto memiliki fitur untuk pengendalian dan pemantauan peralatan listrik dirumah apakah sudah dalam kondisi hidup (on) atau mati (off). Setelah diaktifkan melalui media handphone menggunakan jaringan internet. Untuk membangun *Smart Home* ini terdiri 2 bagian yaitu perangkat keras dan perangkat lunak. Perangkat keras digunakan sebagai media interface antara komputer (arduino) dengan peralatan listrik yang dikendalikan sedangkan perangkat lunaknya digunakan untuk mengaktifkan perangkat keras dan komunikasi antara arduino dan *smartphone*

Pada tugas akhir ini, penulis akan menggunakan ide dari penelitian diatas yaitu membuat sistem *monitoring* dan *controlling* di rumah tinggal dengan menggunakan sistem web untuk melakukan *monitoring* dan *controlling* meliputi alat-alat seperti kipas, lampu,, pintu dan jemuran.

### **2.1.3 Rancang Bangun Sistem Jemuran Pakaian Otomatis Menggunakan Metode *Fuzzy Logic***

Pada penelitian “Rancang Bangun Sistem Jemuran Pakaian Otomatis Menggunakan Metode *Fuzzy Logic*” oleh Adetiya Bagus Nusantara telah dibuat prototipe sistem jemuran pakaian otomatis menggunakan *microcontroller* arduino, modul wifi, *fuzzy logic* untuk menghasilkan *output* yang akurat. Rancang bangun prototipe ini ditujukan untuk memudahkan penghuni rumah ketika menjemur pakaian agar bila sewaktu-waktu hujan melanda, jemuran secara otomatis memindahkan pakaian menuju tempat yang teduh sehingga pakaian tidak basah terkena air hujan

Pada tugas akhir ini penulis akan menggunakan ide dari penelitian di atas untuk membangun sistem jemuran otomatis dan akan digabungkan dengan sistem perangkat lainnya.

#### 2.1.4 *Door Lock Berbasis Internet of Things (IoT)*

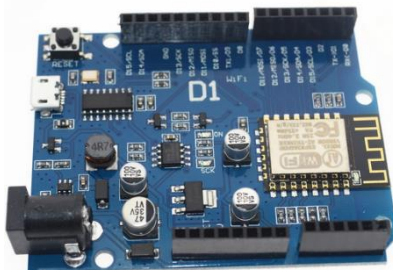
Pada penelitian “*Door Lock berbasis Internet of Things*” oleh Kamal Prihandani dan Agung Susilo Yuda Irawan yang membahas pengembangan dari *Internet of Things* yaitu *Door Lock*. Dikarenakan permasalahan dalam penggunaan kunci konvensional yang dapat diduplikat jika dipinjamkan ke orang lain. Tidak dapat diketahui langsung jika pintu dibuka paksa dan tidak dapat mengakses pintu dari jarak jauh pada penelitian ini sistem *Door Lock* yang dibangun dapat meminjamkan key akses untuk *unlock/lock* kunci, memberikan pemberitahuan jika pintu dibuka dengan paksa dan dapat melakukan *unlock/lock* kunci dari jarak jauh

Pada tugas akhir ini, penulis akan menggunakan ide dari penelitian diatas untuk diterapkan pada sistem pintu. Pada tugas akhir ini akan digunakan kode verifikasi untuk membuka pintu atau mengunci pintu melalui sistem web dan sistem akan mencatat log akses pintu.

## 2.2 **Hardware Tools**

### 2.2.1 **WeMos D1 R1**

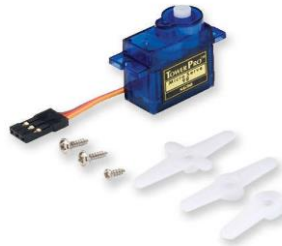
*WeMos* D1 merupakan module development board



yang berbasis *WiFi* dari ESP8266 yang dimana dapat diprogram menggunakan *software* IDE Arduino, perangkat inilah yang nantinya akan menghubungkan perangkat dengan *internet* melalui *WiFi*.

### **2.2.2 Motor Servo SG90**

Motor Servo adalah sebuah motor dengan sistem umpan balik tertutup dimana posisi dari motor akan diinformasikan kembali ke rangkaian control yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer dan rangkaian control, potensiometer berfungsi untuk menentukan batas sudut dari putaran servo sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor



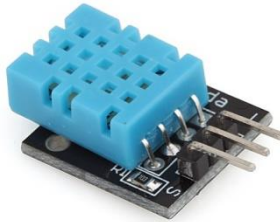
**Gambar 0.2 Motor Servo G90**

### **2.2.3 Sensor Kelembapan DHT11**

Merupakan sensor untuk mensensi objek suhu dan kelembapan pada 1 module yang dimana memiliki output sinyal digital yang sudah terkalibrasi. Module sensor ini tergolong kedalam elemen resistif seperti perangkat pengukur suhu seperti contohnya yaitu NTC. Keunggulan dari sensor DHT11 dibanding dengan yang lainnya antara



lain memiliki kualitas embacaan data sensing yang sangat baik, responsif serta tidak mudah terinterferensi



**Gambar 0.3 Sensor DHT11**

#### **2.2.4 Jumper Cable**

Kabel jumper adalah kabel yang digunakan untuk menghubungkan Arduino dengan sensor atau modul lainnya, terdapat tiga jenis kabel jumper, *Female to Male*, *Female to Female*, *Male to Male*, dimana masing masing memiliki fungsi yang sama hanya berbeda pada ujung kabelnya saja



**Gambar 0.4 Jumper Cable**

#### **2.2.5 Solenoid Door Lock**

*Solenoid Door Lock* ini adalah salah satu solenoid pengunci otomatis yang difungsikan khusus sebagai solenoid untuk pengunci pintu. *Solenoid Door Lock* ini membutuhkan tegangan supply 12V. sistem kerja solenoid

pengunci pintu ini adalah NC( *Normally Close*). Katup solenoid akan tertarik jika ada tegangan dan sebaliknya katup solenoid akan memanjang jika tidak ada tegangan.



**Gambar 0.5 Solenoid Door Lock**

### **2.2.6 *Light Dependent Resistor***

*Light Dependent Resistor* merupakan salah satu komponen resistor yang nilai resistansi nya akan berubah-ubah sesuai dengan intensitas cahaya yang mengenai sensor ini. *LDR* juga dapat digunakan sebagai sensor cahaya. Perlu diketahui bahwa nilai resistensi dari sensor ini sangat bergantung pada intensitas cahaya. Semakin banyak cahaya yang mengenainya maka akan semakin menurun nilai resistensinya



**Gambar 0.6 Light Dependant Resistor**

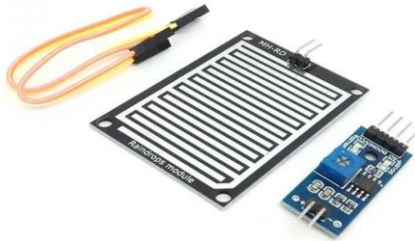
### **2.2.7 4 Channel Relay 5V**

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat Kontak Saklar/Switch). *Relay* menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi.



**Gambar 0.7 4 Channel Relay 5**

## 2.2.8 Raindrop Sensor



**Gambar 0.8 Raindrop Sensor**

Sensor hujan adalah jenis sensor yang berfungsi untuk mendeteksi terjadinya hujan atau tidak, yang dapat difungsikan dalam segala macam aplikasi dalam kehidupan sehari – hari. Dipasaran sensor ini dijual dalam bentuk module sehingga hanya perlu menyediakan kabel jumper untuk dihubungkan ke mikrokontroler atau Arduino. Prinsip kerja dari module sensor ini yaitu pada saat ada air hujan turun dan mengenai panel sensor maka akan terjadi proses elektrolisis oleh air hujan. Dan karena air hujan termasuk dalam golongan cairan elektrolit yang dimana cairan tersebut akan menghantarkan arus listrik.

## 2.3 *Software Tools*

### 2.3.1 **Arduino IDE**

Arduino IDE merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman

Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.

### **2.3.2 XAMPP**

XAMPP adalah sebuah paket perangkat lunak (software) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris. XAMPP pada Tugas Akhir ini akan digunakan sebagai *WebServer* serta penggunaan *database* MySQL yang telah tersedia pada XAMPP.

### **2.3.3 CodeIgniter 3**

CodeIgniter 3 adalah sebuah *Framework* berbasis PHP yang digunakan pada pembuatan *website* untuk melakukan *Monitoring* dan *Controlling* Sistem Rumah Cerdas.

### **2.3.4 Visual Studio Code**

Visual Code adalah sebuah teks editor yang digunakan penulis dalam melakukan pemrograman untuk membangun *Website Monitoring* dan *Controlling* Sistem Rumah Cerdas

### **2.3.5 SQLYog**

Sebuah software untuk membuat database yang dibutuhkan oleh penulis untuk membuat Sistem Rumah Cerdas

*[Halaman ini sengaja dikosongkan]*

## **BAB III PERANCANGAN**

Pada bagian ini akan dijelaskan rancangan *hardware* serta sistem yang akan dibangun oleh penulis pada Tugas Akhir ini.

### **4.1 Perancangan Sistem**

Perancangan dan implementasi Prototipe Sistem Rumah Cerdas ini merupakan sebuah prototipe yang dirancang untuk membantu penghuni suatu rumah menghindari kelalaian seperti jemuran yang lupa dimasukan ketika penghuni keluar rumah, kunci pintu yang tertinggal atau pintu rumah yang lupa dikunci, dan juga penerangan rumah ketika penghuni rumah sedang tidak berada di rumah. Sistem ini akan menggunakan beberapa sensor yang terhubung ke *microcontroller* WeMos D1R1 dan akan mendeteksi keadaan cuaca (untuk sistem jemuran) ketika cuaca terdeteksi akan turun hujan maka jemuran akan masuk secara otomatis. Begitu juga dengan sistem lampu dan kipas yang nantinya akan mendeteksi keadaan sekitar rumah jika sudah gelap maka lampu akan otomatis menyala sedangkan kipas akan mendeteksi suhu sehingga jika ada penghuni rumah yang masuk maka suhu ruangan akan lebih nyaman disbanding suhu di luar. Untuk sistem pintu nantinya penghuni dapat mengontrol untuk membuka atau mengunci pintu menggunakan web dan akan menerima kode verifikasi melalui email untuk membuka pintu. Untuk dapat berfungsi alat ini perlu terhubung ke Wi-fi. Sistem juga akan mencatat log siapa saja pengguna yang mengakses pintu. Fitur-fitur yang disediakan adalah sebagai berikut

#### **1. Sistem otomatis**

Jemuran akan otomatis masuk jika sensor nanti mendeteksi kondisi di sekitar rumah akan turun hujan atau tidak kondusif untuk menjemur pakaian. Lampu akan menyala ketika keadaan sekitar rumah sudah mulai gelap.

Kipas akan menyala secara otomatis jika suhu di luar cukup panas.

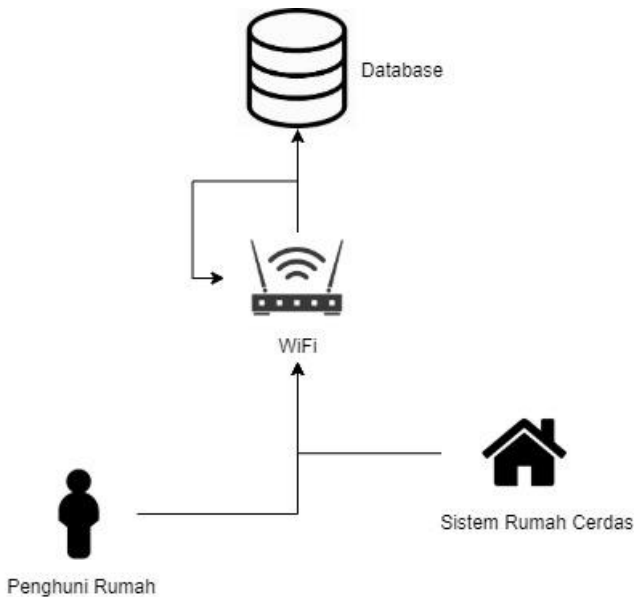
## 2. Akses kunci pintu secara *mobile*

Penghuni rumah dapat mengakses kunci pintu melalui web, sistem akan mengirim kode verifikasi melalui email.

## 3. *Web Monitoring*

Penghuni rumah dapat melihat status jemuran apakah jemuran sedang dijemur apa tidak, dan dapat melihat status pintu apakah sudah terkunci atau belum terkunci. Penghuni rumah dapat melihat juga log pengguna yang mengakses pintu.

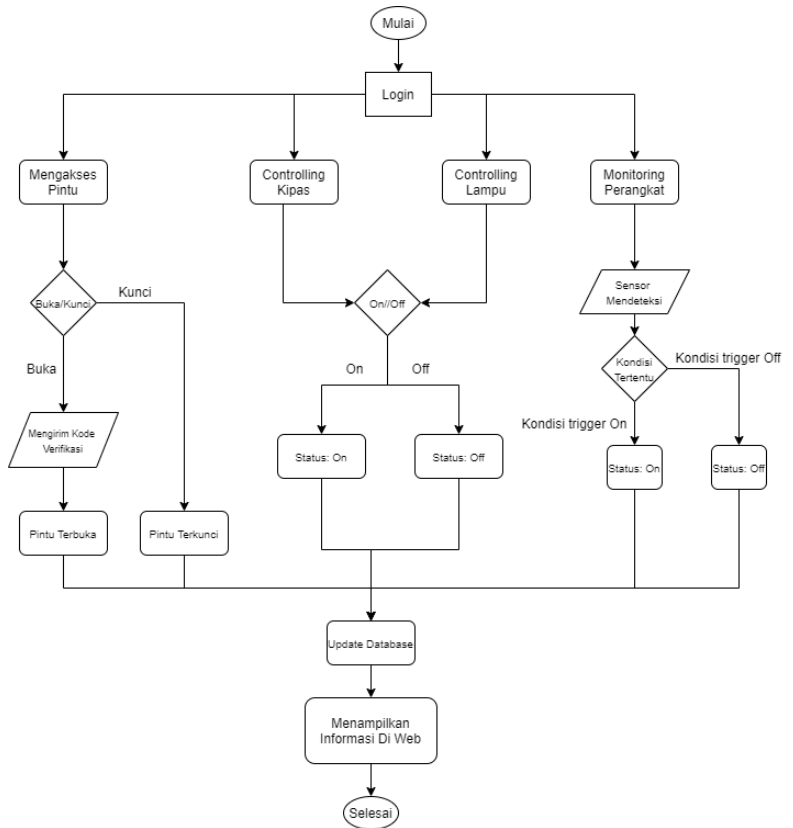
Sistem ini memiliki Alur interaksi device sebagai berikut



**Gambar 4.1 Rancangan Jaringan Sistem Rumah Cerdas**



Alur kerja Sistem Rumah Cerdas dapat dilihat dalam flowchart berikut:

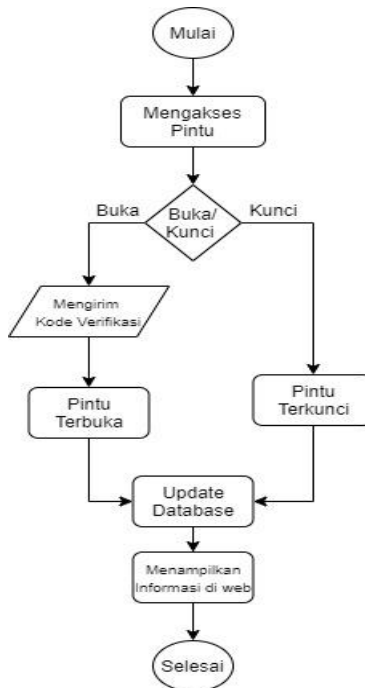


**Gambar 4.2 Flowchart Sistem Rumah Cerdas**

Pada flowchart Sistem Rumah Cerdas memiliki beberapa alur yaitu:

### 1. *Monitoring dan Controlling Pintu*

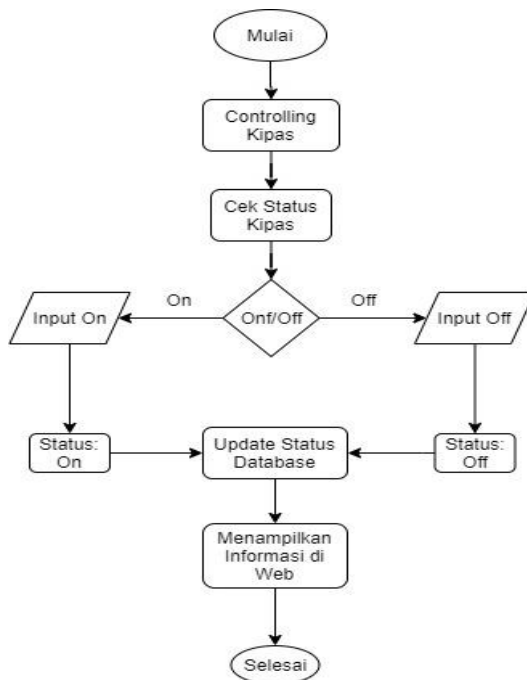
Pada proses ini penghuni dapat mengakses pintu melalui sistem. Dengan asumsi penghuni rumah sudah teregistrasi di web Sistem Rumah Cerdas, maka penghuni sekaligus pengguna sistem ini akan mendapatkan kode verifikasi melalui web agar bisa membuka kunci. Dalam sistem ini akan digunakan *Solenoid Door Lock* sebagai perangkat untuk mengunci pintu.



**Gambar 4.3 Flowchart Alur Mengakses Pintu**

## 2. *Monitoring dan Controlling Kipas*

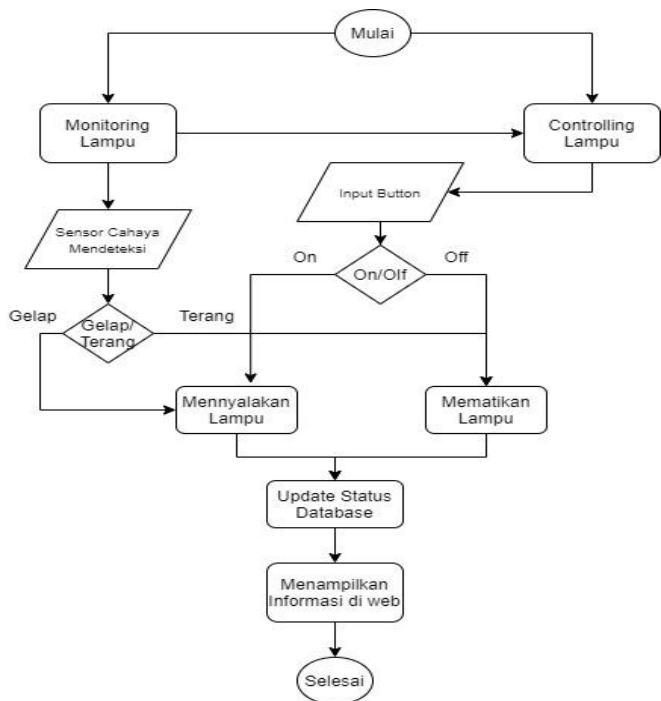
Pada proses ini penghuni rumah dapat melakukan controlling melalui web Sistem Rumah Cerdas, dengan asumsi penghuni rumah sudah terdaftar di web Sistem Rumah Cerdas. Pada sistem ini akan digunakan sensor DHT11 untuk mendeteksi suhu sekitar untuk mengaktifkan kipas secara otomatis jika sistem sensor diaktifkan. Alur Controlling Kipas dapat dilihat dalam flowchart berikut:



**Gambar 4.4** Flowchart Alur Kerja Kipas

### 3. Monitoring dan Controlling Lampu

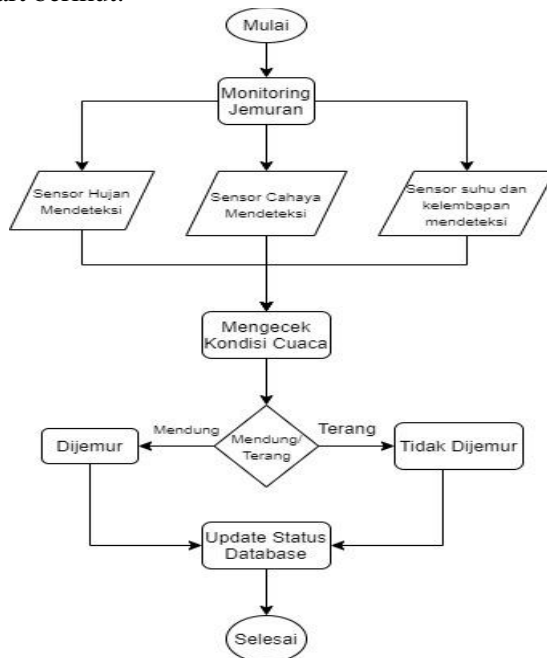
Pada proses ini penghuni rumah dapat melakukan monitoring dan controlling lampu. Sensor cahaya akan mendeteksi keadaan sekitar, jika sudah gelap maka lampu akan otomatis menyala dan sebaliknya. Penghuni rumah juga dapat melakukan controlling melalui web Sistem Rumah. Pada sistem lampu ini digunakan sensor *Light Dependant Resisto* untuk mendeteksi kondisi cahaya sekitar. Alur kerja lampu dapat dilihat melalui flowchart berikut:



**Gambar 4.5 Flowchart Alur Sistem Lampu**

#### 4. Monitoring Jemuran

Pada proses ini penghuni rumah dapat memonitoring jemuran. Sistem jemuran ini menggunakan 3 sensor yaitu sensor suhu dan kelembapan(DHT11),sensor cahaya (Light Dependant Resistor) dan sensor hujan (Raindrop Sensor) nantinya 3 sensor tersebut akan mendeteksi jika cuaca di sekitar rumah efektif untuk menjemur pakaian maka jemuran akan secara otomatis dijemur dan juga sebaliknya. Alur kerja sistem jemuran otomatis dapat dilihat pada flowchart berikut:



**Gambar 4.6 Flowchart Alur Monitoring Jemuran**

## 4.2 Perancangan Database

Perancangan Database ini berfungsi untuk penyimpanan data oleh *Web Service* yang nantinya data akan berubah sesuai data yang dikirim melalui *Web Service* dan akan ditampilkan pada web sistem rumah cerdas ini. Database ini berisi 6 tabel yaitu:

**Tabel 4.1 Daftar Tabel**

Nama Tabel	Deskripsi
pintu	Tabel yang berisi data pintu yang terhubung dengan sistem
jemuran	Tabel yang berisi data jemuran yang terhubung dengan sistem
Log_pintu	Tabel yang berisi data log interaksi pada pintu
User	Tabel yang berisi data penghuni rumah yang dapat mengakses sistem
lampu	Tabel yang berisi data lampu yang terhubung dengan sistem
kipas	Tabel yang berisi data kipas yang terhubung dengan sistem

Penjelasan mengenai rincian tabel sebagai berikut

**Tabel 4.2 Detail Tabel Pintu**

pintu		
Atribut	Data Type	Deskripsi
Id_device	Integer (11)	Atribut <i>primary key</i> pada table ini yang berfungsi sebagai ID pintu yang terhubung ke sistem.
qr_code	Varchar(25)	Atribut yang berisi kode verifikasi untuk mengakses pintu
	Int (1)	Atribut yang berisi

Status		status dari pintu yang terhubung ke sistem
--------	--	--

Status	id_device	id_micro	qr_code
1	51001	1	314258
0	51002	1	192708

**Gambar 4.7 Hasil Eksekusi Query Tabel Pintu**

**Tabel 4.3 Detail Tabel Jemuran**

jemuran		
Atribut	Data Type	Deskripsi
Id_jemuran	Integer (11)	Atribut <i>primary key</i> pada table ini yang berfungsi sebagai ID jemuran yang terhubung pada sistem .
Id_micro	Integer(1)	Atribut yang berisi id <i>microcontroller</i> yang digunakan
Status_jemuran	Int(1)	Atribut yang berisi status jemuran yang terhubung ke sistem

id_jemuran	id_micro	status_jemuran
1	2	1
41001	2	1

**Gambar 4.8 Hasil Eksekusi Query Tabel Jemuran**

**Tabel 4.4 Detail Tabel Log\_pintu**

Log_pintu		
Atribut	DataType	Deskripsi
Id_log	Integer (11)	Atribut <i>primary key</i> pada table ini yang berfungsi sebagai ID <i>primary</i> pada tabel.
tanggal_log	Timestamp	Atribut untuk mencatat
Id_device	Int(11)	Atribut <i>foreign key</i> pada tabel yang didapat dari table pintu.
Id_user	Int(11)	Atribut <i>foreign key</i> pada tabel yang didapat dari tabel user.
status	varchar (50)	Atribut yang digunakan untuk menyatakan status pintu di log.



id_log	tanggal_log	id_device	id_user	status
1	2020-05-11 20:31:23	51001	NULL	terbuka
2	2020-05-11 20:31:26	51001	NULL	terkunci
3	2020-05-11 20:34:12	51001	2	terbuka
4	2020-05-11 20:34:14	51001	2	terkunci
5	2020-05-11 20:34:55	51002	1	terkunci
6	2020-05-11 20:56:21	51001	1	terbuka

**Gambar 4.9 Hasil Eksekusi Query Tabel Log\_pintu**

**Tabel 4.5 Detail Tabel Lampu**

lampu		
Atribut	Data Type	Deskripsi
Id_lampu	Integer (11)	Atribut <i>primary key</i> pada table ini yang berfungsi sebagai ID <i>primary</i> pada tabel.
Id_micro	Integer(1)	Atribut yang berisi id <i>microcontroller</i> yang digunakan
Status_lampu	Int(1)	Atribut yang berisi status lampu yang terhubung dengan sistem

id_lampu	id_micro	status_lampu	sensor_lampu
1	4	1	0

**Gambar 4.10 Hasil Eksekusi Query Tabel Lampu**

**Tabel 4.6 Detail Tabel Kipas**

Kipas		
Atribut	Data Type	Deskripsi
Id_kipas	Integer (11)	Atribut <i>primary key</i> pada table ini yang berfungsi sebagai ID <i>primary</i> pada tabel.
Id_micro	Integer (1)	Atribut yang berisi id <i>microcontroller</i> yang digunakan
Status_lampu	Int(1)	Atribut yang berisi status kipas yang terhubung dengan sistem

id_kipas	id_micro	status_kipas	sensor_kipas
1	3	1	1

**Gambar 4.11 Hasil Eksekusi Query Tabel Kipas**

### 4.3 Perancangan Alat

Perancangan prototipe Sistem Rumah Cerdas akan menggunakan beberapa komponen *hardware*, dengan WeMos D1 R1 sebagai *microcontroller*, diantaranya:

1. 4-Channel Relay 5V, untuk menghubungkan perangkat-perangkat seperti kipas, lampu dan *solenoid door lock* dengan *microcontroller*.
2. Solenoid Lock Door 12V, untuk sistem pintu berfungsi sebagai perangkat untuk mengunci pintu.
3. Kipas
4. Kabel Jumper, untuk menghubungkan perangkat agar saling terhubung satu sama lain.
5. Adaptor 12V, untuk menghubungkan perangkat seperti kipas dan *solenoid door lock* dengan sumber listrik.

6. Lampu LED
7. Light Dependant Resistor(Sensor Cahaya), untuk mendeteksi kondisi cahaya sekitar.
8. Sensor Suhu dan Kelembapan (DHT11), untuk mendeteksi tingkat kelembapan dan suhu keadaan sekitar.
9. Raindrop Sensor, untuk mendeteksi hujan.
10. Motor Servo SG90, sebagai perangkat untuk menggerakkan sistem jemuran.

*Microcontroller* akan dihubungkan dengan masing masing komponen melalui 4-Channel Relay 5V dengan mengirimkan arus listrik kepada relay, dimana *microcontroller* akan mengirimkan sinyal *High* dan *Low* kepada relay.

Sensor yang terhubung nantinya akan mengirimkan hasil input kepada *microcontroller*. Setelah itu *microcontroller* yang terhubung ke wifi akan mengupdate status data yang diperoleh dari input sensor yang telah dikirim ke database agar bisa ditampilkan di sistem.

*Microcontroller* yang terhubung pada 4-Channel Relay 5V memiliki fungsi sebagai berikut:

1. Membuka pintu dengan mengirimkan sinyal *Low* saat status pintu = 0 pada tabel pintu dan menutup pintu dengan mengirimkan sinyal *High* saat status pintu = 1 pada tabel pintu kepada relay.
2. Menghidupkan lampu dengan mengirimkan sinyal *Low* saat status\_lampu = 1 pada tabel lampu dan mematikan lampu dengan mengirimkan sinyal *High* saat status\_lampu = 0 pada tabel lampu kepada relay.
3. Menghidupkan kipas dengan mengirimkan sinyal *Low* saat status\_kipas = 1 pada tabel status\_kipas dan mematikan kipas dengan mengirimkan sinyal *High* saat status\_kipas = 0 pada tabel kipas kepada relay.
4. Mengaktifkan Jemuran dengan mengirimkan sinyal *Low* saat status\_jemuran = 1 pada tabel jemuran dan menonaktifkan jemuran dengan mengirimkan sinyal

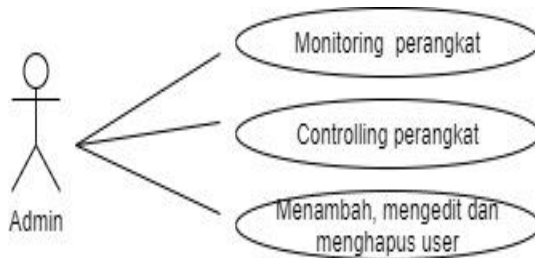
High saat status\_jemuran = 0 pada tabel jemuran kepada relay.

#### 4.4 Perancangan Sistem Monitor

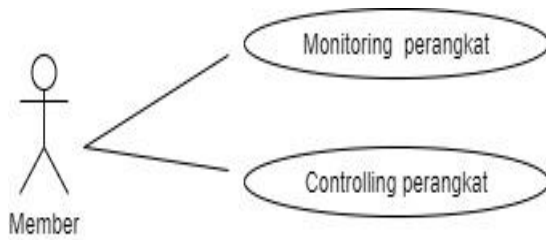
Sistem Monitor dibangun menggunakan framework CodeIgniter 3. Sistem monitor digunakan untuk memantau status pintu rumah, status jemuran, status kipas dan juga status lampu. Dari sistem monitor ini penghuni rumah dapat mengakses pintu rumah sehingga dapat membuka dan mengunci pintu rumah secara *mobile*. Penghuni rumah juga dapat melakukan controlling kipas dan lampu dan memantau kondisi jemuran.

Dalam sistem monitor ini terdapat dua role user yaitu admin dan member. Admin dapat melihat daftar user yang terdaftar di sistem, menambah user baru, mengubah data user dan menghapus user yang telah terdaftar

Perancangan sistem monitoring dan controlling dapat dilihat melalui use case diagram berikut:



Gambar 4.12 Use Case Diagram Admin



**Gambar 4.13 Use Case Diagram Member**

***[Halaman ini sengaja dikosongkan]***

## BAB IV IMPLEMENTASI

Pada bab ini akan dijelaskan tentang implementasi yang dilakukan berdasarkan perancangan pada bab sebelumnya

### 5.1 Implementasi Database

Pada implementasi database ini, penulis menggunakan tool SQLyog untuk membuat *query database*, XAMP dan PHPmyadmin sebagai penyimpanan *database*, *database* akan dibuat sebanyak 6 tabel seperti yang sudah dijelaskan di perancangan database

#### 1. Tabel user

Pada pengimplementasian tabel user akan menjalankan query seperti gambar dibawah

```
CREATE TABLE `user` (
  `id_user` int(20) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `role` varchar(20) NOT NULL DEFAULT
'Member',
  PRIMARY KEY (`id_user`);
```

**Gambar 5.1 Kode Sumber Tabel User**

#### 2. Tabel Pintu

Pada pengimplementasian tabel pintu akan menjalankan *query* seperti gambar dibawah

```

CREATE TABLE `pintu` (
  `status` int(1) NOT NULL,
  `id_device` int(11) NOT NULL
  AUTO_INCREMENT,
  `id_micro` varchar(11) NOT NULL,
  `qr_code` varchar(25) NOT NULL,
  PRIMARY KEY (`id_device`);

```

**Gambar 5.2 Kode Sumber Tabel Pintu**

### 3. Tabel kipas

Pada pengimplementasian tabel kipas akan menjalankan *query* seperti gambar dibawah

```

CREATE TABLE `kipas` (
  `id_kipas` INT(11) NOT NULL AUTO_INCREMENT,
  `id_micro` VARCHAR(11) NOT NULL,
  `status_kipas` INT(1) NOT NULL,
  PRIMARY KEY (`id_kipas`);

```

**Gambar 5.3 Kode Sumber Tabel Kipas**

### 4. Tabel Jemuran

Pada pengimplementasian tabel jemuran akan menjalankan *query* seperti gambar dibawah

```

CREATE TABLE `jemuran` (
  `id_jemuran` int(11) NOT NULL,
  `id_micro` varchar(11) NOT NULL,
  `status_jemuran` int(1) NOT NULL,
  PRIMARY KEY (`id_jemuran`);

```

**Gambar 5.4 Kode Sumber Tabel Jemuran**

### 5. Tabel Lampu

Pada pengimplementasian tabel lampu akan menjalankan *query* seperti gambar dibawah



```

CREATE TABLE `lampu` (
  `id_lampu` int(11) NOT NULL AUTO_INCREMENT,
  `id_micro` varchar(11) NOT NULL,
  `status_lampu` int(1) NOT NULL,
  PRIMARY KEY (`id_lampu`);

```

**Gambar 5.5 Kode Sumber Tabel Lampu**

## 6. Tabel Log\_Pintu

Pada pengimplementasian tabel log\_pintu akan menjalankan *query* seperti gambar dibawah

```

CREATE TABLE `log_pintu` (
  `id_log` int(11) NOT NULL AUTO_INCREMENT,
  `tanggal_log` timestamp NOT NULL DEFAULT
  CURRENT_TIMESTAMP ON UPDATE
  CURRENT_TIMESTAMP,
  `id_device` int(11) DEFAULT NULL,
  `id_user` int(20) DEFAULT NULL,
  `status` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id_log`),
  KEY `fkpintu` (`id_device`),
  KEY `fkuser` (`id_user`),
  CONSTRAINT `fkpintu` FOREIGN KEY
  (`id_device`) REFERENCES `pintu`
  (`id_device`) ON DELETE CASCADE ON UPDATE
  CASCADE,
  CONSTRAINT `fkuser` FOREIGN KEY (`id_user`)
  REFERENCES `user` (`id_user`) ON DELETE
  CASCADE ON UPDATE CASCADE;

```

**Gambar 5.6 Kode Sumber Tabel Log\_Pintu**

## 5.2 Implementasi Alat

### 5.2.1 Rangkaian Alat Sistem Rumah Cerdas

Dalam Mengimplementasi Sistem Rumah Cerdas diperlukan beberapa alat yang dihubungkan ke

*microcontroller* seperti yang sudah dijelaskan pada perancangan alat. Perangkat seperti lampu, kipas dan *Solenoid Door Lock* akan disambungkan dengan adaptor dan dihubungkan ke Relay. Sedangkan perangkat seperti sensor *DHT11*, *Light Dependant Resistant* dan *Raindrop Sensor* akan dihubungkan ke *microcontroller* melalui *breadboard*. *Microcontroller* akan menghidupkan dan mematikan relay dengan mengirimkan sinyal *Low* dan *High*. Ketika relay hidup maka saklar akan tertutup dan arus akan dialirkan ke terminal *NO* (*Normally Open*). Alat-alat yang terhubung dengan relay akan menggunakan terminal *NO*. Setelah alat-alat nya sudah saling terhubung maka kode program bisa dimasukkan kedalam *microcontroller* menggunakan *Arduino IDE* Implementasi kode program adalah sebagai berikut :

```
#include "DHT.h"
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
#include <Servo.h>
#define INPUT_SIZE 15
#define DHTPIN D2 // Data di pin A0

#define DHTTYPE DHT11 // yang kita
gunakan adalah DHT 11
```

**Gambar 5.7 Include Libray Arduino IDE**

Pada Kode Program diatas berisi library yang digunakan pada Sistem Rumah Cerdas serta mendeklarasikan pin yang digunakan dan juga Jenis Sensor yang digunakan

```

const char *ssid = "Rumah yang Yahud";
//ENTER YOUR WIFI SETTINGS
const char *password = "haikalkucing";
String
URL_ADDRESS="http://192.168.0.16/smarthome/sen
sor/";

void connectWiFi () {
    WiFi.mode(WIFI_OFF);           //Prevents
reconnection issue (taking too long to
connect)
    delay(1000);
    WiFi.mode(WIFI_STA);           //This line
hides the viewing of ESP as wifi hotspot

    WiFi.begin(ssid, password);     //Connect to
your WiFi router
    Serial.println("");

    Serial.print("Connecting");
    // Wait for connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
}

```

**Gambar 5.8 Fungsi Connect Wifi Pada Microcontroller**

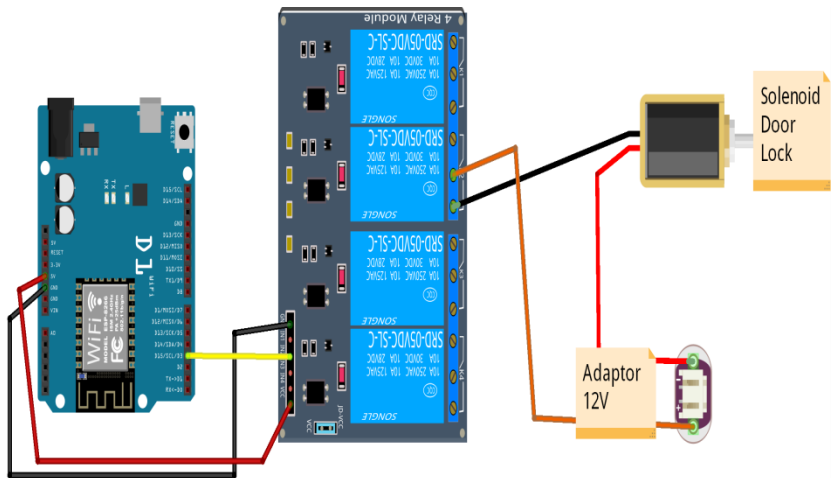
Pada Kode Program diatas berisi fungsi *Microcontroller* untuk terhubung dengan *WiFi* dan mendeklarasikan SSID dari WiFi yang akan digunakan dan juga Password dari WiFi. Menggunakan Fungsi `connectWiFi` untuk melakukan konfigurasi antara microcontroller dengan WiFi.

```
void loop() {  
  getCahaya();  
  getDHT();  
  getData("getdata");  
  packData();  
  kondisijemuran(jemuran, hujan, cahaya,  
humidity);  
  Serial.print("Jemuran: ");  
  Serial.println(jemuran);  
  printSensor();  
  setservo(jemuran);  
  setPerangkat(pintu, kipas, lampu,  
sensorlp, sensorkp);  
  Serial.println("");  
  delay(200); // delay 2 detik
```

**Gambar 5.9 Kode Sumber Integrasi Seluruh Sistem**

Pada kode sumber diatas *Microcontroller* akan menjalankan fungsi void loop diatas secara terus menerus selama *Microcontroller* terhubung dengan arus listrik. Fungsi ini digunakan untuk menjalankan fungsi-fungsi dari masing-masing sistem yang ada agar berjalan menjadi Sistem Rumah Cerdas

### 5.2.1.1 Rangkaian Sistem Pintu



Gambar 5.10 Rangkaian Sistem Pintu

fritzing

Tabel 5.1 Rangkaian Sistem Pintu

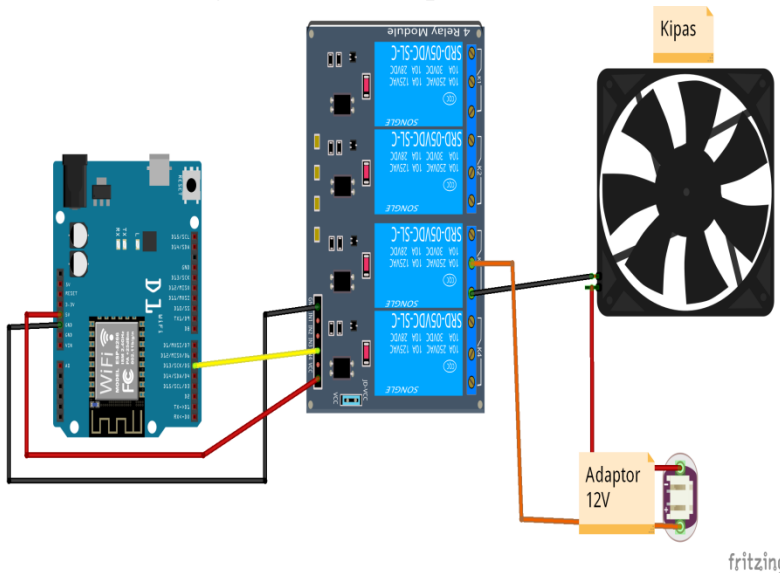
	PIN	
Wemos D1 R1	Relay	Terminal(Pintu)
5V	VCC	-
GND	GND	-
D3	IN-2	NO
-	-	-
-	-	-
-	-	-

```
void setPerangkat(int pnt, int kps, int  
lmp, int srlp, int srkp){  
    Serial.print("Pintu: ");  
    if(pnt==0){  
        Serial.println("BUKA");  
        digitalWrite(in2, LOW);  
    }else{  
        Serial.println("TUTUP");  
        digitalWrite(in2, HIGH);  
    }  
}
```

**Gambar 5.11 Kode Sumber Sistem Pintu**

Pada Kode program diatas berfungsi untuk mengatur *Solenoid Door Lock* untuk menentukan apakah pintu dalam kondisi terkunci atau terbuka. Status pintu diperoleh dari database yang nantinya akan disimpan di variabel pnt. Jika variabel pnt = 0 maka *microcontroller* akan menjalankan *command* digitalWrite pada *Channel Input 2* dan mengirimkan arus *Low* kepada Relay sehingga *Solenoid* berada di kondisi terbuka. Jika variabel pnt = 1 maka *microcontroller* akan menjalankan *command* digitalWrite pada *Channel Input 2* dan mengirimkan arus *High* kepada Relay sehingga *Solenoid* berada di kondisi terkunci

### 5.2.1.2 Rangkaian Sistem Kipas



Gambar 5.12 Rangkaian Sistem Kipas

Tabel 5.1 Rangkaian Sistem Kipas

PIN		
Wemos D1 R1	Relay	Terminal(Kipas)
5V	VCC	-
GND	GND	-
-	-	-
-	-	-
D5	IN-3	NO
-	-	-

Pada Sistem kipas, terdapat dua cara untuk mengaktifkan kipas. Yang pertama adalah dengan menekan button *On* dan *Off* melalui Web. Yang kedua

dengan mengaktifkan sistem sensor dalam mendeteksi suhu sekitar. Kode program untuk sistem ini dapat dilihat pada gambar berikut:

```

void setPerangkat(int pnt, int kps, int lmp,
int srlp, int srkp){

    if(srkp==0){
        Serial.println("Kipas: SENSOR MATI");
        Serial.print("Kipas: ");
        if(kps==1){
            Serial.println("ON");
            digitalWrite(in3, LOW);
        }
        else{
            Serial.println("OFF");
            digitalWrite(in3, HIGH);
        }
    }
    else if(srkp==1){
        Serial.println("Kipas: SENSOR HIDUP");
        Serial.print("Kipas: ");
        if(shkp==0){
            Serial.println("OFF");
            Serial.print("digin");
            digitalWrite(in3, HIGH);
            getData("updatekipas/1?status=0");
        }
        else if(shkp==1){
            Serial.println("ON");
            Serial.print("panas");
            digitalWrite(in3, LOW);
            getData("updatekipas/1?status=1");
        }
    }
}

```

**Gambar 5.13 Kode Program Sistem Kipas**



Pada kode program diatas berfungsi untuk mendeteksi status sensor pada kipas. Jika srkp = 0 maka status sensor kipas nonaktif dan pengguna bisa melakukan *controlling* melalui web. Selanjutnya sistem akan mendeteksi status kipas yang diperoleh dari database. Jika kps = 1 maka *microcontroller* akan menjalankan perintah digitalWrite pada *channel input 3* yaitu yang terhubung dengan sistem kipas dan akan mengirimkan arus *Low* pada relay agar kipas aktif. Jika kps = 0 maka *microcontroller* akan menjalankan perintah digitalWrite pada chanel input 3 dan mengirimkan arus High untuk menonaktifkan kipas. Jika srkp = 1 maka *Microcontroller* akan mengecek nilai variabel shkp pada fungsi getDHT.

```
void getDHT() {
  // Baca kelembapan udara dalam satuan persen
  (%)
  h = dht.readHumidity();
  // Baca suhu udara dalam satuan Celsius
  (default)
  t = dht.readTemperature();

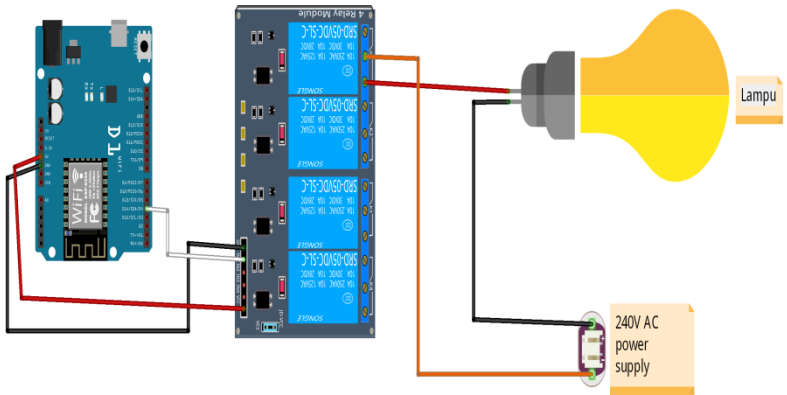
  if(h>80){humidity = "lembab";}
  else{humidity = "kering";}

  if(t<28){
    temperature = "dingin";
    shkp=0;
  }
  else{
    temperature = "panas";
    shkp=1;
  }
}
```

**Gambar 5.14 Kode Sumber Mendeteksi Suhu**

Sensor *DHT11* akan mendeteksi suhu sekitar dan menyimpannya di variabel *t*, dan akan melakukan pengecekan jika suhu dibawa 28 derajat celcius maka akan menyimpan nilai pada variabel *temperature* = “dingin” dan akan melakukan set value pada variabel *shkp*=1. jika diatas 28 derajat celcius akan menyimpan nilai pada variabel *temperature* = “panas” dan akan melakukan set value pada variabel *shkp*=1.

### 5.2.1.3 Rangkaian Sistem Lampu



**Gambar 5.15 Rangkaian Sistem Lampu**

**Tabel 5.2 Rangkaian Sistem Lampu**

PIN		
Wemos D1 R1	Relay	Terminal(Lampu)
5V	VCC	-
GND	GND	-
-	-	-
-	-	-
-	-	-
D4	IN-1	NO

Pada Sistem Lampu terdapat dua cara untuk mengaktifkan Lampu. Yang pertama adalah dengan menekan button On dan Off melalui Web. Yang kedua dengan mengaktifkan sistem sensor dalam mendeteksi kondisi cahaya sekitar. Kode program untuk sistem ini

```

if(srlp==0){
  Serial.println("Lampu: SENSOR MATI");
  Serial.print("Lampu: ");
  if(lmp==1){
    Serial.println("ON");
    digitalWrite(in1, LOW);
  }
  else{
    Serial.println("OFF");
    digitalWrite(in1, HIGH);
  }
}
else if(srlp==1){
  Serial.println("Lampu: SENSOR HIDUP");
  Serial.print("Lampu: ");
  if(chylp==0){
    Serial.println("OFF");
    Serial.print("terang");
    digitalWrite(in1, HIGH);
    getData("updatelampu/1?status=0");
  }
  else if(chylp==1){
    Serial.println("ON");
    Serial.print("gelap");
    digitalWrite(in1, LOW);
    getData("updatelampu/1?status=1");
  }
}
}

```

**Gambar 5.16** Kode Program Sistem Lampu

Pada kode program diatas berfungsi untuk mendeteksi status sensor pada lampu. Jika srlp = 0 maka status sensor lampu nonaktif dan pengguna bisa melakukan controlling melalui web. Selanjutnya sistem akan mendeteksi status lampu yang diperoleh dari database. Jika Imp = 1 maka microcontroller akan menjalankan perintah digitalWrite pada channel input 1 yaitu yang terhubung dengan sistem lampu dan akan mengirimkan arus Low pada relay agar lampu aktif. Jika Imp = 0 maka microcontroller akan menjalankan perintah digitalWrite pada chanel input 1 dan mengirimkan arus High untuk menonaktifkan lampu. Jika srlp = 1 maka Microcontroller akan mengecek nilai variabel chylp pada fungsi getCahaya.

```

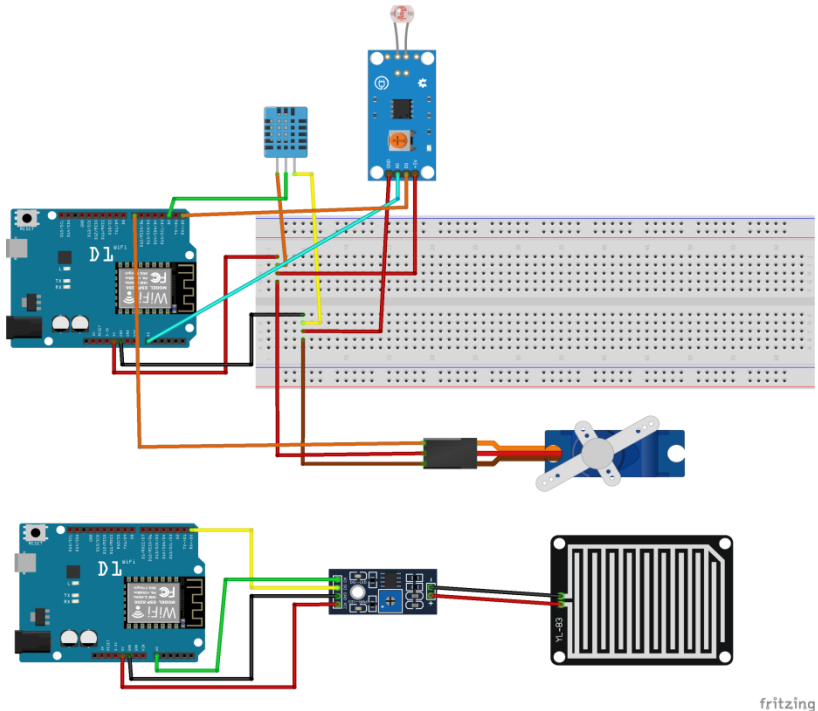
void getCahaya () {
    unsigned int AnalogValue;
    AnalogValue = analogRead(A0);
    if (AnalogValue<300) {
        cahaya = "terang";
        chylp=0;
    }else{
        cahaya = "gelap";
        chylp=1;
    }
}

```

**Gambar 5.17 Kode Program Mendeteksi Cahaya**

*Light Dependant Resistor* akan mendeteksi cahaya sekitar dengan mendeteksi AnalogValue melalui pin A0. Jika nilai dari AnalogValue < 300 maka *microcontroller* akan mengubah nilai variabel cahaya = “terang” dan mengubah nilai variabel chylp menjadi 0. Jika nilai dari AnalogValue > 300 maka *microcontroller* akan mengubah nilai variabel cahaya = “gelap” dan mengubah nilai variabel chylp menjadi 1.

### 5.2.1.4 Rangkaian Sistem Jemuran



Gambar 5.18 Rangkaian Sistem Jemuran

fritzing

Tabel 5.3 Rangkaian Sistem Jemuran

PIN				
Wemos D1 R1	DHT11	Light Dependant Resistor	Raindrop Sensor	Motor Servo SG90
5V	VCC	VCC	VCC	VCC
GND	GND	GND	GND	GND
D2	Out	-	-	-

A0	-	A0	A0	-
D0	-	D0	D0	-
D7	-	-	-	Out

Pada Sistem Jemuran ada 3 sensor yang digunakan yaitu Sensor *DHT11*, *Light Dependant*

```

void getDHT() {
    // Baca kelembapan udara dalam satuan
    persen (%)
    h = dht.readHumidity();
    // Baca suhu udara dalam satuan Celsius
    (default)
    t = dht.readTemperature();

    if(h>80){humidity = "lembab";}
    else{humidity = "kering";}
}

```

**Gambar 5.19 Kode Program Mendeteksi Kelembapan**

Dengan menggunakan fungsi `getDHT` sensor *DHT11* akan mendeteksi kelembapan di sekitar sensor. Jika nilai dari variable `h > 80` maka sensor akan mengubah nilai variabel Kelembapan = “lembap” jika nilai `h < 80` maka nilai variable kelembapan = “kering”.

```
void loop() {
    val_analogique=analogRead(capteur_A);
    if(val_analogique>840){
        Serial.println("Digital value : DRY");
        data = "DRY";
        delay(10);
    }
    else{
        Serial.println("Digital value : WET");
        data = "WET";
        delay(10);
    }
    Serial.print("Analog value : ");
    Serial.println(val_analogique);
    Serial.print("Data: ");
    Serial.println(data);
    postData();
    delay(1000);
}
```

**Gambar 5.20 Kode Sumber Mendeteksi Hujan**

Pada fungsi diatas digunakan untuk mendeteksi hujan. Sensor akan disambungkan dengan pin A0. Variabel `val_analogique` menyimpan nilai analog yang dideteksi oleh sensor dan jika bernilai  $> 840$  maka *microcontroller* akan mengubah nilai variabel `data` = "DRY". Jika nilai analog yang dideteksi bernilai  $< 840$  maka nilai variabel `data` = "WET".

Pada gambar 4.12 fungsi `getCahaya` akan digunakan kembali untuk sistem jemuran ini. Hasil input yang diperoleh dari 3 sensor akan digunakan untuk menentukan status jemuran apakah akan dijemur atau tidak.

```
void kondisijemuran(String jmr, String hjn,
String chy, String hmd){
    if(hjn=="WET"){
        delay(200);
        jemuran="0";

getData("setjemuran?jemuran="+jemuran);
        return;
    }

    if((chy=="gelap") &&
(humidity=="lembab")){
        delay(200);
        jemuran="0";

getData("setjemuran?jemuran="+jemuran);
        return;
    }

    if((chy=="gelap") &&
(humidity=="kering")){
        delay(200);
        jemuran="1";

getData("setjemuran?jemuran="+jemuran);
        return;
    }

    if((chy=="terang") && (hjn=="DRY")){
        delay(200);
        jemuran="1";
    }
}
```

**Gambar 5.21 Kode Sumber Set Status Jemuran**



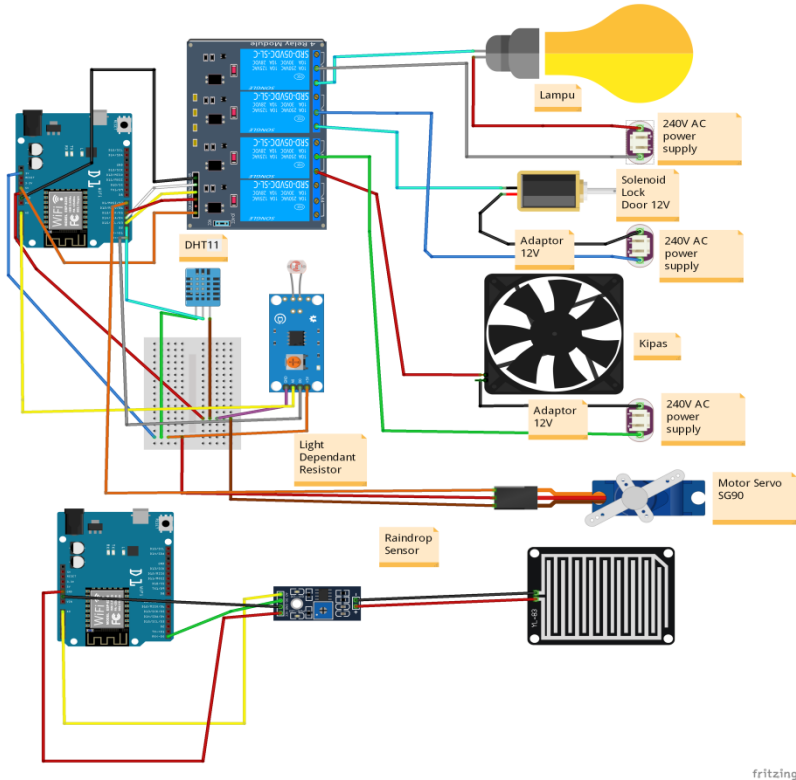
Kode Sumber diatas merupakan kode sumber untuk mengubah status jemuran. Jika kondisi diatas terpenuhi maka *Microcontroller* akan mengubah status jemuran sesuai dengan kondisi yg terpenuhi.

```
void setservo(String jmr){
  if(jmr=="1"){
    while(angle<=180){
      servo.write(angle);
      delay(20);
      angle++;
    }
  }
  else if(jmr=="0"){
    while(angle>=0){
      servo.write(angle);
      delay(20);
      angle--;
    }
  }
}
```

**Gambar 5.22 Kode Sumber Untuk Mengatur Servo**

Setelah *microcontroller* memperoleh status jemuran, maka selanjutnya menggunakan fungsi *setservo* untuk mengubah posisi jemuran.

### 5.2.1.5 Rangkaian Sistem Rumah Cerdas



**Gambar 5.23 Rangkaian Sistem Rumah Cerdas**

Gambar diatas merupakan gambar keseluruhan rangkaian alat-alat sistem rumah cerdas. Sensor-sensor sudah terhubung dengan micro controller dan perangkat seperti kipas, lampu dan *Solenoid Door Lock* sudah terhubung ke relay yang terhubung ke *microcontroller*

### 5.3 Implementasi Web

Pada implementasi ini penulis menggunakan *framework* CodeIgniter 3. Untuk melakukan konfigurasi database dengan framework digunakan kode sumber seperti dibawah

```
$db['default'] = array(
    'dsn'      => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'smarthome',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !==
'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

**Gambar 5.24 Konfigurasi Database**

Pada implementasi sistem monitoring dan controlling terdapat 8 controller, seperti berikut:

#### 1. Admin.php

Controller ini digunakan untuk membuat role admin yang berfungsi untuk menambahkan *user*, menghapus user dan mengedit data *user*.

**2. Login.php**

Controller ini berfungsi untuk membuat sistem log in dan Log out untuk user.

**3. Controll.php**

Controller ini berfungsi untuk monitoring dan controlling pada sistem pintu.

**4. Jemuran.php**

Controller ini berfungsi untuk monitoring pada sistem jemuran.

**5. Lampu.php**

Controller ini berfungsi untuk monitoring dan controlling pada sistem lampu.

**6. Kipas.php**

Controller ini berfungsi untuk monitoring dan controlling pada sistem Kipas

**7. Log\_pintu.php**

Controller ini berfungsi untuk melihat log akses pada sistem pintu

**8. Sensor.php**

Controller ini berfungsi untuk menghubungkan sensor-sensor dengan sistem web

**5.3.1 Manage User Oleh Admin**

Admin dapat memiliki 3 fungsi utama yaitu untuk menambahkan user baru, menghapus *user* dan juga mengedit data *user* yang sudah terdaftar di sistem.

```

1.  function user_add(){
2.      $this->form_validation-
>set_rules('email', 'Email', 'required');
3.      $this->form_validation-
>set_rules('username', 'Username', 'required');
4.      $this->form_validation-
>set_rules('password', 'Password', 'required');
5.
6.      if ($this->form_validation->run() != FALSE) {
7.          // var_dump($this->input->post());
8.          $insert = array(
9.              'username' => $this->input-
>post('username'),
10.             'email' => $this->input->post('email'),
11.             'password' => md5( $this->input-
>post('password')),
12.             );
13.             $this->db->insert('user',$insert);
14.             redirect('admin/user_add');
15.         } else {
16.             $data['user'] = $this->db->get("user")-
>result();
17.             $view = array(
18.                 $this->load->view('template/v_header'),
19.                 $this->load-
>view('content/Admin/v_adduser', $data),
20.                 $this->load->view('template/v_footer')
21.             );
22.         }
23.     }

```

**Gambar 5.25 Fungsi Add User**

Fungsi diatas merupakan fungsi yang terdapat pada controller Admin.php yang berfungsi untuk menambah *user* baru

```

1. public function user_del($id=null)
2.     {
3.         if (!isset($id)) show_404();
4.
5.         if ($this->db-
>delete('user', array("id_user" => $id)))
6.         {
7.             redirect('admin/user_add');
8.         }

```

**Gambar 5.26 Fungsi Delete User**

Fungsi diatas merupakan fungsi yang berfungsi untuk menghapus user oleh admin.

```

1. if ($this->form_validation->run() != FALSE) {
2.     if (!$this->input->post('passwordEdit')){
3.         $update = array(
4.             'username' => $this->input-
>post('usernameEdit'),
5.             'email' => $this->input-
>post('emailEdit'),
6.             'password' => md5( $this->input-
>post('passwordEdit')),
7.             'role' => $this->input-
>post('roleEdit'),
8.         );
9.     } else {
10.        $update = array(
11.            'username' => $this->input-
>post('usernameEdit'),
12.            'email' => $this->input-
>post('emailEdit'),
13.            'password' => md5( $this->input-
>post('passwordEdit')),
14.            'role' => $this->input-
>post('roleEdit'),
15.        );
16.    }

```

**Gambar 5.27 Fungsi Edit User**

Fungsi diatas merupakan fungsi untuk mengubah data user yang sudah terdaftar di sistem

### 5.3.2 *Monitoring dan Controlling Perangkat*

Untuk melakukan *monitoring* dan *controlling* perangkat yang terhubung ke sistem. Sistem akan mengecek status dari masing-masing perangkat. Jika melakukan *controlling* maka sistem akan mengupdate status perangkat dengan menjalankan fungsi update

```

1.     function update($id,$status_perangkat){
2.         $where = $check1 = array('id_perangkat' =
> $id);
3.         if($status_perangkat==0){
4.             $update = array(
5.                 'status_perangkat' => 1
6.             );
7.         }elseif($status_perangkat==1){
8.             $update = array(
9.                 'status_perangkat' => 0
10.            );
11.        }
12.        $this->db->where($where);
13.        $this->db-
>update('(tabel perangkat)', $update);
14.        redirect(base_url('(tabel perangkat)'));
15.    }

```

**Gambar 5.28 Fungsi Update Status Perangkat**

Jika menggunakan sensor untuk mengaktifkan perangkat Maka sistem akan mengecek status sensor. Sistem akan mengecek status sensor sama seperti saat mengecek status perangkat dengan menggunakan fungsi berikut

```

1.     function update_sensor($id,$sensor_perangkat){
2.         $where = $check1 = array('(id perangkat)' =>
   $id);
3.         if($sensor_perangkat==0){
4.             $update = array(
5.                 'sensor_perangkat' => 1
6.             );
7.         }elseif($sensor_perangkat==1){
8.             $update = array(
9.                 '(sensor perangkat)' => 0
10.            );
11.        }
12.        $this->db->where($where);
13.        $this->db-
>update('(tabel perangkat',$update);
14.        redirect(base_url('(tabel perangkat)'));
15.    }

```

**Gambar 5.29 Fungsi Update Status Sensor**

### 5.3.3 Deteksi Sensor

Sensor yang terhubung akan mendeteksi kondisi sekitar yang nantinya akan disimpan hasil input nya di file cahaya.json

```

1.     $stpintu = $pintu['Status'];
2.     $stkipas = $kipas['status_kipas'];
3.     $stlampu = $lampu['status_lampu'];
4.     $sensorlp = $lampu['sensor_lampu'];
5.     $sensorkp = $kipas['sensor_kipas'];
6.     $result = $this->db->get("jemuran", $where)-
>row_array();
7.     $fsta = file_get_contents("application/views/content/cahaya.json");
8.     $data = json_decode($fsta,true);
9.     echo $data['valldr'].'.$result['status_jemuran'].
'.$stpintu.'.$stkipas.'.$stlampu.'.$sensorlp.'
'.$sensorkp;

```

**Gambar 5.30 Fungsi Get Data Sensor**



### 5.3.4 Kode Verifikasi Akses Pintu

Untuk mengakses pintu. Nantinya sistem akan mengirim kode verifikasi ke email pengguna yang telah terdaftar di database. Dengan melakukan konfigurasi maka nanti sistem akan mampu mengirim kode verifikasi melalui email

```

1.     $code = random_string("numeric",6);
2.     $from_email = "smarthomeiot141@gmail.com";

3.     $this->load->library('email');
4.     $config = array();
5.     $config['protocol'] = 'smtp';
6.     $config['smtp_host'] = 'ssl://smtp.gmail.co
m';
7.     $config['smtp_user'] = 'smarthomeiot141@ma
il.com';
8.     $config['smtp_pass'] = 'haeKal141';
9.     $config['smtp_port'] = 465;

```

**Gambar 5.31 Kode Sumber Konfigurasi Email**

```

1. $this->email->initialize($config);
2.     $this->email->set_newline("\r\n");
3.     $this->email-
>from($from_email, 'Identification');
4.     $this->email->to($this->session-
>userdata('email'));
5.     $this->email-
>subject('Send Email Codeigniter');
6.     $this->email->message($code);
7.     $this->email->send();
8.     $this->db->where(['id_device' => $id]);
9.     $this->db-
>update('pintu',['qr_code' => $code]);

```

**Gambar 5.32 Kode Sumber Mengirim Kode Verifikasi**

***[Halaman ini sengaja dikosongkan]***

## BAB V UJICOBA DAN EVALUASI

Pada bab ini dijelaskan tentang uji coba dan evaluasi dari implementasi yang telah dilakukan pada Tugas Akhir ini.

### 6.1 Lingkup Uji Coba

Pada lingkup uji coba, penulis menggunakan jaringan koneksi rumah “*Home Network*” untuk melakukan uji coba Sistem Rumah Cerdas, sistem *monitoring* dan *controlling* dijalankan pada laptop dengan spesifikasi sebagai berikut:

**Tabel 6.1 Tabel Spesifikasi Laptop**

Komponen	Spesifikasi
CPU	Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz (8 CPUs), ~2.8GHz
Sistem Operasi	Windows 10 Home Single Language 64-bit (10.0, Build 18362)
Memori	16384MB RAM
Bios	BIOS Date: 04/22/16 14:05:07 Ver: 05.0000B
Penyimpanan	1 TB

### 6.2 Skenario Pengujian

Pada bagian ini akan dibahas mengenai proses uji coba yang akan digunakan. Pada proses uji coba, pengujian dilakukan dengan menjalankan serangkaian perintah terhadap sistem menggunakan kasus penggunaan untuk menguji kinerja dari sistem. pengujian ini akan dilakukan terhadap 4 fitur yang dimiliki Sistem Rumah Cerdas yaitu:

1. *Monitoring* dan *controlling* sistem pintu. Pengujian akan dilakukan untuk menguji apakah sistem mampu menjalankan fungsinya untuk melakukan monitoring pintu dan melakukan controlling pintu melalui web dan menguji komponen-komponen yang terhubung berfungsi dengan baik

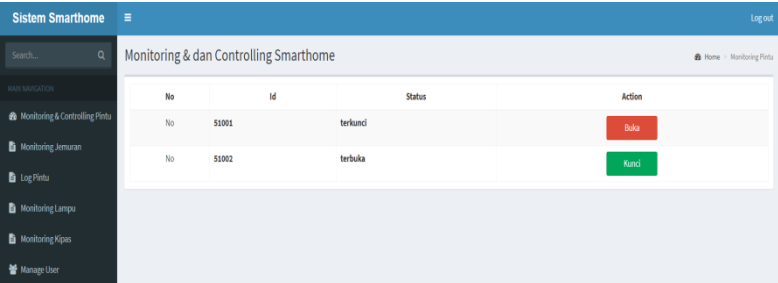
2. *Monitoring* dan *controlling* sistem lampu. Pengujian akan dilakukan untuk menguji apakah sistem mampu menjalankan fungsinya untuk melakukan *monitoring* lampu dan melakukan *controlling* lampu melalui web dan menguji komponen-komponen yang terhubung berfungsi dengan baik
3. *Monitoring* dan *controlling* sistem kipas. Pengujian akan dilakukan untuk menguji apakah sistem mampu menjalankan fungsinya untuk melakukan *monitoring* kipas dan melakukan *controlling* kipas melalui web dan menguji komponen-komponen yang terhubung berfungsi dengan baik
4. *Monitoring* sistem jemuran. Pengujian akan dilakukan untuk menguji apakah sistem mampu menjalankan fungsinya untuk melakukan *monitoring* sistem jemuran melalui web dan menguji komponen-komponen yang terhubung berfungsi dengan baik

## 6.2.1 Monitoring dan Controlling Sistem Pintu

Pada bagian ini akan dilakukan pengujian untuk fitur monitoring dan controlling sistem pintu.

### 6.2.1.1 Skenario Monitoring Sistem Pintu

Pada scenario pengujian ini dilakukan untuk menguji apakah pengguna dapat melakukan monitoring sistem pintu melalui web. Untuk melakukan pengujian ini pengguna masuk pada menu home dan pilih menu “Monitoring dan Controlling Pintu”



No	Id	Status	Action
No	51001	terkunci	Buka
No	51002	terbuka	Kunci

**Gambar 6.1 Menu Monitoring Sistem Pintu**

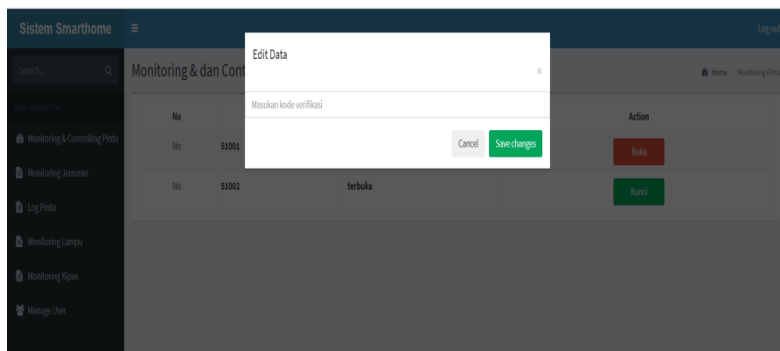
Gambar diatas merupakan tampilan menu monitoring, informasi yang ditampilkan adalah id pintu yang terhubung dengan sistem dan juga status pintu

**Tabel 6.2 Hasil Uji Skenario Monitoring Sistem Pintu**

Fungsi	Harapan	Hasil
Menampilkan informasi Pintu	Berhasil menampilkan Informasi Pintu	OK

### 6.2.1.2 Skenario Controlling Sistem Pintu

Setelah memilih menu “Monitoring dan *Controlling* Pintu” untuk membuka pintu yang terkunci maka pengguna dapat memilih tombol “Buka” pada kolom action. Selanjutnya pengguna akan diminta kode verifikasi yang telah dikirim ke email pengguna dengan asumsi pengguna sudah terdaftar di database Sistem Rumah Cerdas.



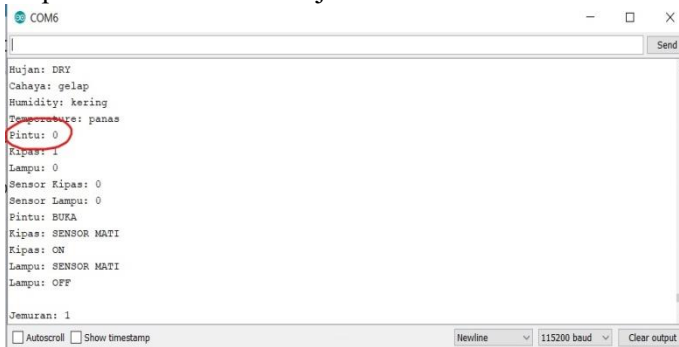
**Gambar 6.2 Masukan Kode Verifikasi**

Sistem akan melakukan generate kode verifikasi setiap user menekan tombol buka. jika sudah muncul tampilan seperti diatas berarti kode verifikasi sudah dikirim ke email pengguna.



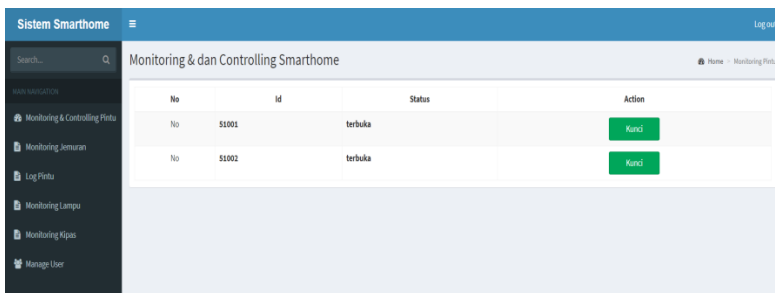
**Gambar 6.3 Pengiriman Kode Verifikasi Ke Email**

Setelah memasukkan kode verifikasi yang benar maka status pintu akan berubah menjadi “terbuka”.



**Gambar 6.4 Status Pintu Terbuka**

Jika status pintu = 0 maka artinya status pintu telah terbuka dan status akan di update ke web untuk ditampilkan.



**Gambar 6.5 Status Pintu Terbuka Pada Web**

Jika pengguna menekan tombol kunci maka status pintu akan berubah menjadi status dan web akan menampilkan informasi seperti pada **gambar 5.1**

```
Humidity: kering
Temperature: panas
Pintu: 1
Kipas: 1
Lampu: 0
Sensor Kipas: 0
Sensor Lampu: 0
```

**Gambar 6.6 Status Pintu Terkunci**

**Tabel 6.3 Hasil Uji Skenario Controlling Pintu**

Fungsi	Harapan	Hasil
Sistem mengirim kode verifikasi melalui email	Sistem Berhasil Mengirim Kode Verifikasi melalui Email	OK
Pintu dapat dibuka melalui sistem controlling pada web	Pintu berhasil terbuka melalui sistem controlling pada web	OK
Update status pintu	Sistem berhasil melakukan Update status pintu pada web	OK

### 6.2.1.3 Skenario Melihat Log Pintu

Jika pengguna melakukan aksi membuka atau mengunci pintu. Nantinya sistem akan mencatat Log akses pintu dan akan ditampilkan pada menu “Log Pintu”.



No	2020-06-17 21:20:51	51001	1	terkunci
No	2020-06-17 21:21:27	51001	1	terbuka
No	2020-06-17 21:37:40	51001	1	terkunci
No	2020-06-17 21:40:13	51001	1	terbuka
No	2020-06-17 21:40:23	51001	1	terkunci
No	2020-06-20 00:31:56	51001	1	terbuka
No	2020-06-20 00:33:21	51001	1	terkunci

**Gambar 6.7 Menampilakn informasi Log Pintu**

**Tabel 6.4 Hasil Uji Skenario Menampilkan Log Pintu**

Fungsi	Harapan	Hasil
Sistem menampilkan log pintu	Sistem berhasil menampilkan log pintu	OK

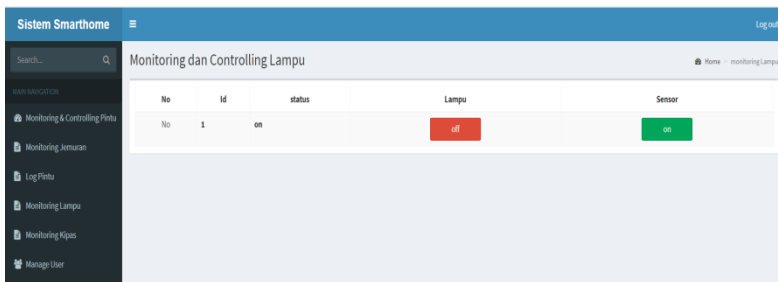
## 6.2.2 Monitoring dan Controlling Sistem Lampu

Pada bagian ini akan dilakukan pengujian untuk fitur monitoring dan controlling lampu.

### 6.2.2.1 Skenario Monitoring Lampu

Untuk melakukan pengujian ini, pengguna memilih menu “Monitoring Lampu”. Setelah itu sistem akan menampilkan informasi lampu yang terhubung ke sistem meliputi id lampu, status lampu dan juga status sensor.





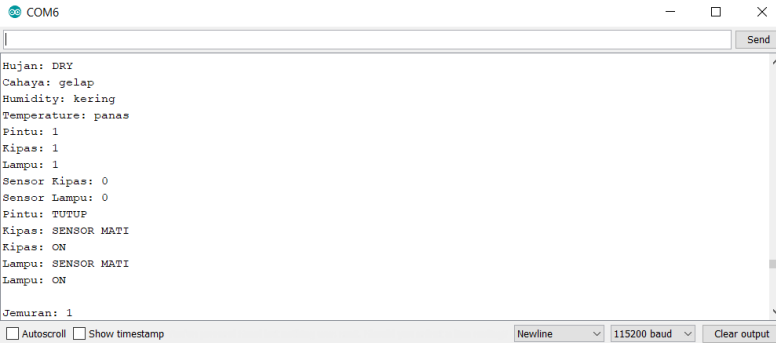
**Gambar 6.8 Menampilkan Status Informasi Lampu**

**Tabel 6.5 Hasil Uji Skenario Monitoring Lampu**

Fungsi	Harapan	Hasil
Sistem menampilkan informasi lampu	Sistem berhasil menampilkan informasi lampu	OK

### 6.2.2.2 Skenario Controlling Lampu

Pada menu Monitoring Lampu sistem menampilkan tombol Off/On pada kolom lampu dan juga pada kolom sensor. Jika pengguna ingin melakukan controlling melalui web maka pengguna dapat menekan tombol off/on pada kolom lampu. Setelah itu maka sistem akan mengaktifkan lampu dan mengupdate status lampu sesuai input yang diperoleh.



**Gambar 6.9 Serial Monitor Status Lampu On**

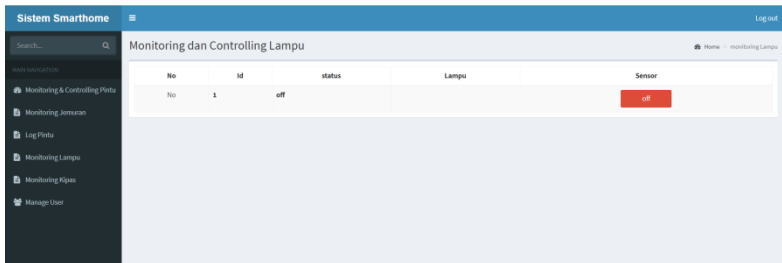
Jika diaktifkan maka status lampu akan berubah menjadi 1 yang berarti lampu sedang aktif dan sistem akan menampilkannya di web seperti pada **gambar 5.8**

**Tabel 6.6 Hasil Uji Skenario Controlling Lampu**

Fungsi	Harapan	Hasil
Sistem dapat mengaktifkan lampu	Sistem berhasil mengaktifkan lampu	OK
Sistem mengupdate dan menampilkan status lampu	Sistem berhasil mengupdate dan menampilkan status lampu	OK

### 6.2.2.3 Skenario Deteksi Sensor Sistem Lampu

Pada menu monitoring lampu pengguna dapat mengaktifkan atau menonaktifkan sistem sensor dengan cara menekan tombol On/Off pada kolom “Sensor”. Jika sensor dalam keadaan aktif maka tombol On//Off tidak ditampilkan pada kolom “Lampu”.



**Gambar 6.10 Tampilan Informasi Status Sensor On**

Jika sensor aktif maka sensor nanti akan mendeteksi kondisi cahaya di sekitar. Jika sensor mendeteksi cahaya di sekitar dalam kondisi terang maka lampu akan dalam keadaan nonaktif dan jika dalam kondisi gelap maka lampu akan aktif.

```
Hujan: DRY
Cahaya: gelap
Humidity: kering
Temperature: panas
Pintu: 1
Kipas: 1
Lampu: 1
Sensor Kipas: 0
Sensor Lampu: 1
```

**Gambar 6.11 Sensor On Lampu On**

```
Cahaya: terang
Humidity: kering
Temperature: panas
Pintu: 1
Kipas: 1
Lampu: 0
Sensor Kipas: 0
Sensor Lampu: 1
Pintu: TUTUP
```

**Gambar 6.12 Sensor On Lampu Off**

Sistem akan mengupdate status terbaru dari lampu dan akan menampilkannya di web seperti **gambar 5.10**

**Tabel 6.7 Hasil Uji Skenario Deteksi Sensor**

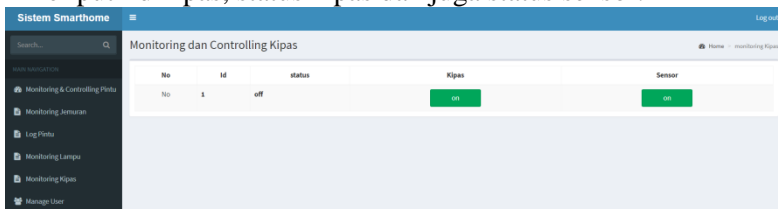
Fungsi	Harapan	Hasil
Sensor dapat mendeteksi kondisi cahaya	Sensor berhasil mendeteksi kondisi cahaya	OK
Sistem dapat mengaktifkan lampu berdasarkan hasil deteksi sensor	Sistem berhasil mengaktifkan lampu berdasarkan hasil deteksi sensor	OK
Sistem dapat mengupdate dan menampilkan informasi di web	Sistem berhasil mengupdate dan menampilkan informasi di web	OK

## 6.2.3 Monitoring dan Controlling Sistem Kipas

Pada bagian ini akan dilakukan pengujian untuk fitur monitoring dan controlling kipas.

### 6.2.3.1 Skenario Monitoring kipas

Untuk melakukan pengujian ini, pengguna memilih menu “Monitoring Kipas”. Setelah itu sistem akan menampilkan informasi kipas yang terhubung ke sistem meliputi id kipas, status kipas dan juga status sensor.



**Gambar 6.13 Menampilkan Informasi Status Kipas Off**

**Tabel 6.8 Hasil Uji Skenario Monitoring Kipas**

Fungsi	Harapan	Hasil
Sistem menampilkan informasi kipas	Sistem berhasil menampilkan informasi kipas	OK

### 6.2.3.2 Skenario Controlling kipas

Pada menu “Monitoring Kipas” sistem menampilkan tombol Off/On pada kolom kipas dan juga pada kolom sensor. Jika pengguna ingin melakukan controlling melalui web maka pengguna dapat menekan tombol off/on pada kolom kipas. Setelah itu maka sistem akan mengaktifkan kipas dan mengupdate status kipas sesuai input yang diperoleh.

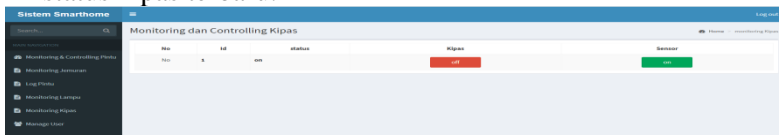
```

Temperature: panas
Pintu: 1
Kipas: 1
Lampu: 0
Sensor Kipas: 1
Sensor Lampu: 0

```

**Gambar 6.14 Status Kipas Aktif**

Setelah sistem mendapatkan data status kipas. Maka sistem akan mengupdatenya dan menampilkan informasi status kipas terbaru.

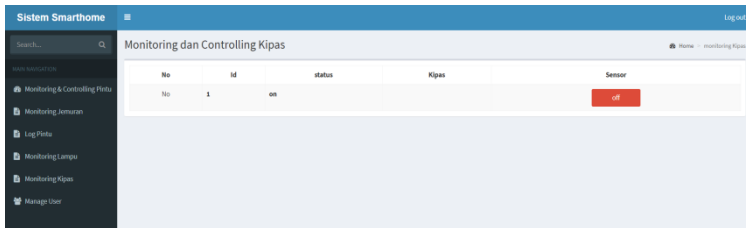
**Gambar 6.15 Status Kipas On**

**Tabel 6.9 Hasil Uji Skenario Controlling Kipas**

Fungsi	Harapan	Hasil
Sistem dapat mengaktifkan kipas	Sistem berhasil mengaktifkan kipas	OK
Sistem mengupdate dan menampilkan status kipas	Sistem berhasil mengupdate dan menampilkan status kipas	OK

### 6.2.3.3 Skenario Deteksi Sensor Sistem Kipas

Pada menu monitoring kipas pengguna dapat mengaktifkan atau menonaktifkan sistem sensor dengan cara menekan tombol On/Off pada kolom “Sensor”. Jika sensor dalam keadaan aktif maka tombol On//Off tidak ditampilkan pada kolom “Kipas”.

**Gambar 6.16 Status Sensor Kipas Aktif**

Jika sensor aktif maka sensor nanti akan mendeteksi kondisi suhu di sekitar. Jika sensor mendeteksi suhu di sekitar dalam kondisi dingin maka kipas akan dalam keadaan nonaktif dan jika dalam kondisi panas maka kipas akan aktif.

```

Temperature: panas
Pintu: 1
Kipas: 1
Lampu: 0
Sensor Kipas: 1
Sensor Lampu: 0

```

**Gambar 6.17** Sensor Aktif Kipas Aktif

**Tabel 6.10** Hasil Uji Skenario Deteksi Sensor Sistem Kipas

Fungsi	Harapan	Hasil
Sensor dapat mendeteksi kondisi suhu	Sensor berhasil mendeteksi kondisi suhu	OK
Sistem dapat mengaktifkan kipas berdasarkan hasil deteksi sensor	Sistem berhasil mengaktifkan kipas berdasarkan hasil deteksi sensor	OK
Sistem dapat mengupdate dan menampilkan informasi di web	Sistem berhasil mengupdate dan menampilkan informasi di web	OK

## 6.2.4 Monitoring Sistem Jemuran

Pada bagian ini akan dilakukan pengujian untuk fitur monitoring sistem jemuran.

### 6.2.4.1 Skenario Monitoring Jemuran

Pada menu “Monitoring Jemuran” sistem akan menampilkan status jemuran. Status jemuran akan ditampilkan seperti pada gambar berikut



**Gambar 6.18 Menampilkan Status Jemuran**

Status jemuran akan berubah berdasarkan data input yang diperoleh dari senso

**Tabel 6.11 Hasil Uji Skenario Monitoring Sistem Jemuran**

Fungsi	Harapan	Hasil
Sistem menampilkan informasi jemuran	Sistem berhasil menampilkan informasi jemuran	OK

#### 6.2.4.2 Skenario Deteksi Sensor Sistem Jemuran

Pada sistem jemuran digunakan 3 sensor yaitu Sensor *DHT11* untuk mendeteksi kelembapan, sensor *Light Dependant Resistor* untuk mendeteksi kondisi cahaya sekitar dan *Raindrop Sensor* untuk mendeteksi hujan. Ada beberapa kondisi untuk mengubah status jemuran

1. Kondisi cahaya gelap,

Jika sensor *Light Dependant Resistor* Mendeteksi bahwa kondisi cahaya gelap maka jemuran berada dalam kondisi sedang tidak dijemur.

2. Kondisi cahaya terang, tidak hujan

Jika sensor *Light Dependant Resistor* Mendeteksi bahwa kondisi cahaya terang dan *Raindrop Sensor*



mendeteksi kondisi tidak hujan maka jemuran berada dalam kondisi sedang dijemur.

3. Kondisi sedang hujan

Jika *Raindrop Sensor* mendeteksi keadaan sedang hujan maka jemuran akan berada di kondisi tidak dijemur.

4. Kondisi cahaya terang, sedang hujan

Jika *Raindrop Sensor* mendeteksi keadaan sedang hujan namun kondisi cahaya terdeteksi sedang terang maka jemuran akan berada di kondisi tidak dijemur.

5. Kondisi cahaya gelap, sedang hujan

Jika *Raindrop Sensor* mendeteksi keadaan sedang hujan dan sensor cahaya mendeteksi kondisi cahaya sedang gelap maka jemuran akan berada di kondisi tidak dijemur.

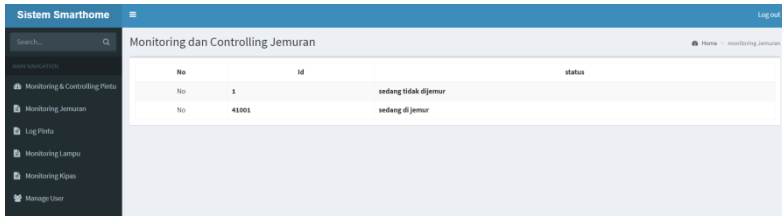
6. Kondisi sedang tidak hujan, cahaya gelap, lembap

Jika *Raindrop Sensor* mendeteksi keadaan sedang tidak hujan, sensor DHT11 mendeteksi keadaan sekitar sedang lembap dan sensor cahaya mendeteksi kondisi cahaya sedang gelap maka jemuran akan berada di kondisi tidak dijemur.

7. Kondisi cahaya terang, lembap

Jika sensor cahaya mendeteksi kondisi cahaya sedang terang dan sensor DHT11 mendeteksi kondisi sedang lembap maka jemuran akan berada di kondisi dijemur.

Setelah mendapatkan input data dari sensor maka sistem akan mengupdate status terbaru ke database dan mengubah posisi jemuran berdasarkan status jemuran yang diperoleh dari database



**Gambar 6.19 Kondisi Jemuran Sedang Tidak Dijemur**

**Tabel 6.12 Hasil Uji Skenario Deteksi Sensor Sistem Jemuran**

Fungsi	Harapan	Hasil
Sensor <i>DHT11</i> dapat mendeteksi kelembapan	Sensor <i>DHT11</i> berhasil mendeteksi kelembapan	OK
<i>Raindrop Sensor</i> dapat mendeteksi Hujan	<i>Raindrop Sensor</i> berhasil mendeteksi Hujan	OK
<i>Light Dependant Resistor</i> dapat mendeteksi kondisi cahaya	<i>Light Dependant Resistor</i> berhasil mendeteksi kondisi cahaya	OK
Sistem dapat mengubah posisi jemuran berdasarkan input data yang diperoleh sensor	Sistem berhasil mengubah posisi jemuran berdasarkan input data yang diperoleh sensor	OK
Sistem dapat mengupdate dan menampilkan status jemuran	Sistem dapat mengupdate dan menampilkan status jemuran	OK

Setelah itu akan dilakukan pengujian performa respons alat ketika melakukan *controlling* dan deteksi sensor saat transisi alat aktif dan nonaktif.

**Tabel 6.13 Uji Respon Sistem Pintu**

Percobaan	Sistem Pintu	
	Pintu Terbuka	Pintu terkunci
1	0.43	0.45
2	0.44	0.45
3	0.44	0.41
4	0.43	0.45
5	0.47	0.45
6	0.48	0.46
7	0.45	0.41
8	0.42	0.45
9	0.45	0.44
10	0.48	0.44
Rata-rata	0.44	0.44

**Tabel 6.14 Uji Respon Sistem Lampu**

Sistem Lampu				
Percobaan	Sensor On		Sensor Off	
	Lampu On	Lampu Off	Lampu On	Lampu Off
1	4.13	4.16	0.35	0.35
2	4.03	4.09	0.36	0.35
3	4.19	4.11	0.37	0.36
4	4.14	4.14	0.37	0.34
5	4.14	4.13	0.35	0.33
6	4.10	4.14	0.40	0.37
7	4.07	4.13	0.38	0.34
8	4.18	4.17	0.37	0.34
9	4.13	4.13	0.33	0.38
10	4.10	4.15	0.35	0.37
Rata-rata	4.12	4.13	0.37	0.35

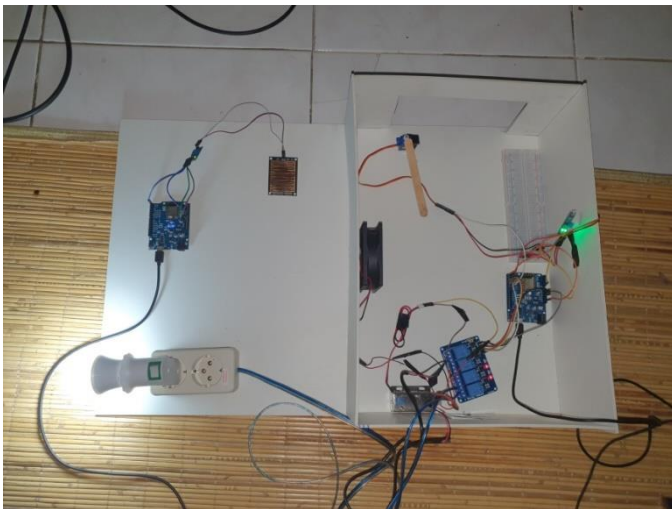
**Tabel 6.15 Uji Respon Sistem Kipas**

Sistem Kipas				
Percobaan	Sensor On		Sensor Off	
	Kipas On	Kipas Off	Kipas On	Kipas Off
1	0.46	0.45	0.37	0.37
2	0.47	0.46	0.40	0.37
3	0.77	0.72	1.45	0.39
4	0.73	0.46	0.35	0.38
5	0.45	0.71	0.44	0.34
6	0.43	0.45	0.38	0.35
7	0.72	0.44	0.39	0.37
8	0.46	0.42	0.40	0.37
9	0.46	0.45	0.36	0.41
10	0.71	0.71	0.36	0.40
Rata-rata	0.56	0.52	0.49	0.37

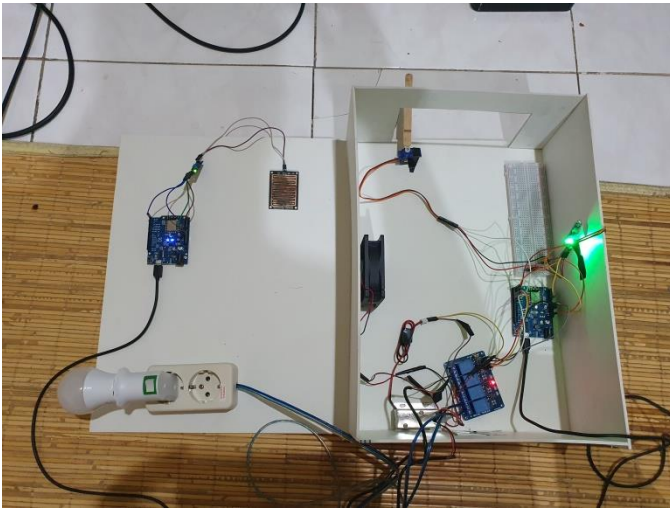
**Tabel 6.16 Uji Respon Sistem Jemuran**

Sistem Jemuran		
Percobaan	Dijemur	Tidak Dijemur
1	4.11	4.16
2	4.05	4.09
3	4.17	4.08
4	4.14	4.09
5	4.12	4.08
6	4.11	4.12
7	4.10	4.25
8	4.11	4.10
9	4.12	4.13
10	4.12	4.10
Rata-rata	4.11	4.12

Setelah dilakukan uji performa, dapat dilihat pada tabel-tabel diatas hasil *response time* dari masing-masing sistem yang diuji. Perbedaan *response time* yang diperoleh saat proses pengujian dapat disebabkan oleh hambatan yang diterima sensor saat melakukan pendeteksian, koneksi yang tidak stabil saat melakukan pengujian dapat mempengaruhi hasil *response time*. Pada sistem jemuran, gangguan pada pergerakan yang dilakukan oleh motor servo dapat mempengaruhi hasil *response time*.



**Gambar 6.20 Prototipe Perangkat Aktif**



**Gambar 6.21 Prototipe Perangkat NonAktif**



## **BAB VI**

### **KESIMPULAN**

Pada bab ini akan dijelaskan kesimpulan dari hasil ujicoba yang telah dilakukan.

#### **7.1 Kesimpulan**

Kesimpulan yang diperoleh dari Tugas Akhir ini didasarkan pada hasil uji coba dan evaluasi. Kesimpulan Tugas Akhir ini adalah sebagai berikut:

1. Website *Monitoring* dan *Controlling* Sistem Rumah Cerdas sudah berhasil diimplementasikan dan berfungsi sesuai dengan tujuan perancangan.
2. Alat yang dihubungkan pada *microcontroller* yang melalui relay berhasil diimplementasikan dan dapat mendeteksi kondisi keadaan sekitar.

#### **7.2 Saran**

Saran yang dapat diberikan dari hasil uji coba dan evaluasi adalah sebagai berikut:

1. Melakukan pengembangan terhadap fitur-fitur yang sudah ada.
2. Menggunakan perangkat-perangkat yang lebih canggih untuk peningkatan kemampuan sistem dan akurasi pendeteksian melalui sensor.

***[Halaman ini sengaja dikosongkan]***

## DAFTAR PUSTAKA

- [1] T. S. Ng, "Microcontroller," in *Studies in Systems, Decision and Control*, 2016.
- [2] B. L. Risteska Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and 10.1016/j.jclepro.2016.10.006.
- [3] B. Sidik, "Framework Codeigniter," *Bandung: Informatika*. 2012.
- [4] Deny Siswanto & Slamet Winardi, "Jemuran Pakaian Otomatis Menggunakan Sensor Hujan," *Narodroid*, 2015.
- [5] Eddi, C. Suhery, and D. Triyanto, "Sistem Penerangan Rumah Otomatis Dengan Sensor Cahaya Berbasis Mikrokontroler," *Tugas Akhir*, 2013.
- [6] A. D. Darusman, M. Dahlan, and F. S. Hilyana, "RANCANG BANGUN PROTOTYPE ALAT PENJEMUR PAKAIAN OTOMATIS BERBASIS ARDUINO UNO," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, 2018, doi: 10.24176/simet.v9i1.2077.
- [7] Pleva GmbH, "DHT11 Humidity & Temperature sensor," *Melliand Textilberichte*, 1995, doi: 10.1007/978-3-319-19303-8\_17.
- [8] C. Wilson, T. Hargreaves, and R. Hauxwell-Baldwin, "Benefits and risks of smart home technologies," *Energy Policy*, 2017, doi: 10.1016/j.enpol.2016.12.047.
- [9] F. Gregorio, G. González, C. Schmidt, and J. Cousseau, "Internet of Things," in *Signals and Communication Technology*, 2020.
- [10] A. Basuki, "Membangun Web Berbasis PHP dengan Framework Codeigniter," *Yogyakarta: Lokomedia*, 2010.
- [11] B. Raharjo, "Belajar Otodidak Framework CodeIgniter," *Informatika Bandung*, 2015.
- [12] D. Uk, "Temperature Sensor DHT 11 Humidity & Temperature Sensor," *D-Robotics*, 2010.

- [13] C. Wilson, T. Hargreaves, and R. Hauxwell-Baldwin, "Smart homes and their users: a systematic analysis and key challenges," *Pers. Ubiquitous Comput.*, 2015, doi: 10.1007/s00779-014-0813-0.
- [14] M. "What is a M. P. 1999. Gilliland, "What is a Microcontroller," "*What is a Microcontroller*". *Parallax*, 1999.

## LAMPIRAN

## 1. Kode Program SQL

```
1. CREATE TABLE `jemuran` (  
2.   `id_jemuran` int(11) NOT NULL,  
3.   `id_micro` varchar(11) NOT NULL,  
4.   `status_jemuran` int(1) NOT NULL,  
5.   PRIMARY KEY (`id_jemuran`)  
6.  
7. CREATE TABLE `kipas` (  
8.   `id_kipas` INT(11) NOT NULL AUTO_INCREMENT,  
9.   `id_micro` VARCHAR(11) NOT NULL,  
10.  `status_kipas` INT(1) NOT NULL,  
11.  PRIMARY KEY (`id_kipas`);  
12.  
13.  
14. CREATE TABLE `lampu` (  
15.  `id_lampu` int(11) NOT NULL AUTO_INCREMENT,  
16.  `id_micro` varchar(11) NOT NULL,  
17.  `status_lampu` int(1) NOT NULL,  
18.  PRIMARY KEY (`id_lampu`)  
19.  
20. CREATE TABLE `pintu` (  
21.  `Status` int(1) NOT NULL,  
22.  `id_device` int(11) NOT NULL AUTO_INCREMENT,  
23.  `id_micro` varchar(11) NOT NULL,  
24.  `qr_code` varchar(25) NOT NULL,  
25.  PRIMARY KEY (`id_device`)  
26.  
27. CREATE TABLE `user` (  
28.  `id_user` int(20) NOT NULL AUTO_INCREMENT,  
29.  `username` varchar(255) NOT NULL,  
30.  `password` varchar(255) NOT NULL,  
31.  `email` varchar(255) NOT NULL,  
32.  `role` varchar(20) NOT NULL DEFAULT 'Member',  
33.  PRIMARY KEY (`id_user`)  
34.  
35. CREATE TABLE `log_pintu` (  
36.  `id_log` int(11) NOT NULL AUTO_INCREMENT,  
37.  `tanggal_log` timestamp NOT NULL DEFAULT CURRENT_TIM  
ESTAMP ON UPDATE CURRENT_TIMESTAMP,
```

```

38. `id_device` int(11) DEFAULT NULL,
39. `id_user` int(20) DEFAULT NULL,
40. `status` varchar(50) DEFAULT NULL,
41. PRIMARY KEY (`id_log`),
42. KEY `fkpintu` (`id_device`),
43. KEY `fkuser` (`id_user`),
44. CONSTRAINT `fkpintu` FOREIGN KEY (`id_device`) REFER
ENCES `pintu` (`id_device`) ON DELETE CASCADE ON UPDAT
E CASCADE,
45. CONSTRAINT `fkuser` FOREIGN KEY (`id_user`) REFERENC
ES `user` (`id_user`) ON DELETE CASCADE ON UPDATE CASC
ADE

```

## 2. Kode Program Admin.php

```

1. <?php
2.
3. class Admin extends CI_Controller{
4.
5.     function __construct(){
6.         parent::__construct();
7.
8.         if($this->session-
>userdata('status') != "Login"){
9.             redirect(base_url("Login"));
10.        }
11.    }
12.
13.    function user_add(){
14.        $this->form_validation-
>set_rules('email', 'Email', 'required');
15.        $this->form_validation-
>set_rules('username', 'Username', 'required');
16.        $this->form_validation-
>set_rules('password', 'Password', 'required');
17.
18.        if ($this->form_validation-
>run() != FALSE) {
19.            // var_dump($this->input->post());

```

```

20.         $insert = array(
21.             'username' => $this->input-
>post('username'),
22.             'email' => $this->input-
>post('email'),
23.             'password' => md5( $this->input-
>post('password')),
24.         );
25.         $this->db->insert('user',$insert);
26.         redirect('admin/user_add');
27.     } else {
28.         $data['user'] = $this->db->get("user")-
>result();
29.         $view = array(
30.             $this->load-
>view('template/v_header'),
31.             $this->load-
>view('content/Admin/v_adduser', $data),
32.             $this->load-
>view('template/v_footer')
33.         );
34.     }
35. }
36.
37. public function user_del($id=null)
38. {
39.     if (!isset($id)) show_404();
40.
41.     if ($this->db-
>delete('user', array("id_user" => $id))) {
42.         redirect('admin/user_add');
43.     }
44. }
45.
46. function user_edit($id=null){
47.
48.     if (!isset($id)) show_404();
49.
50.     $this->form_validation-
>set_rules('emailEdit', 'Email', 'required');
51.     $this->form_validation-
>set_rules('usernameEdit', 'Username', 'required');
52.

```

```
53.         if ($this->form_validation-
>run() != FALSE) {
54.             if(!$this->input->post('passwordEdit')){
55.                 $update = array(
56.                     'username' => $this->input-
>post('usernameEdit'),
57.                     'email' => $this->input-
>post('emailEdit'),
58.                     'password' => md5( $this->input-
>post('passwordEdit')),
59.                     'role' => $this->input-
>post('roleEdit'),
60.                 );
61.             } else {
62.                 $update = array(
63.                     'username' => $this->input-
>post('usernameEdit'),
64.                     'email' => $this->input-
>post('emailEdit'),
65.                     'password' => md5( $this->input-
>post('passwordEdit')),
66.                     'role' => $this->input-
>post('roleEdit'),
67.                 );
68.             }
69.             $this->db-
>where(array('id_user' => $id));
70.             $this->db->update('user',$update);
71.
72.         } redirect('admin/user_add');
73.     }
74. }
```



### 3. Kode Program Controll.php

```
1. <?php
2. defined('BASEPATH') OR exit('No direct script access a
   llowed');
3.
4. class Control extends CI_Controller{
5.
6.     function __construct(){
7.         parent::__construct();
8.         $this->load->helper("string");
9.         if($this->session-
   >userdata('status') != "Login"){
10.             redirect(base_url("login"));
11.         }
12.     }
13.
14.     function index(){
15.         $data['control'] = $this->db->get("pintu")-
   >result();
16.         $view = array(
17.             $this->load->view('template/v_header'),
18.             $this->load-
   >view('content/control/v_control', $data),
19.             $this->load->view('template/v_footer')
20.         );
21.     }
22.
23.     function unlock(){
24.         $output = [];
25.         if($this->input->post('id')){
26.             $this->update_code($this->input-
   >post('id'));
27.             $output[1] = $this->input-
   >post('id');
28.         }
29.         header('Content-Type: application/json');
30.         echo json_encode($output);
31.     }
32.
33.
34.
```

```

35.     function update($id,$status){
36.         $verif= $this->input->post('unlockcode');
37.         $where = $check1 = array('id_device' => $id);

38.         if($status==0){
39.             $update = array(
40.                 'Status' => 1
41.             );
42.             $data=[ 'id_device' => $id, 'id_user' => $t
his->session-
>userdata('id_user'), 'status' => "terkunci"];
43.             $this->db->where($where);
44.             $this->db->update('pintu',$update);
45.             $this->db->insert('log_pintu',$data);
46.             redirect(base_url('control'));
47.         }elseif($status==1){
48.             $data = $this->db-
>get_where("pintu",array('id_device'=> $id))-
>row_array();
49.             $status = $data['qr_code'];
50.             //var_dump($status)
51.             if($status==$verif){
52.                 $update = array(
53.                     'Status' => 0
54.                 );
55.                 $data=[ 'id_device' => $id, 'id_user' =
> $this->session-
>userdata('id_user'), 'status' => "terbuka"];
56.                 $this->db->where($where);
57.                 $this->db->update('pintu',$update);
58.                 $this->db-
>insert('log_pintu',$data);
59.                 redirect(base_url('control'));
60.             } else{
61.                 echo "Kode verifikasi salah";
62.             }
63.         }
64.     }
65. }
66.
67.     public function test(){
68.         $devid = $this->input->post('devid');

```

```

69.         $data = $this->db-
>get_where("t_lock",array('t_devid'=> $devid))-
>row_array();
70.         $status = (int)($data['t_status']);
71.         if($status==0){
72.             echo 'tutup';
73.         }elseif($status==1){
74.             echo 'buka';
75.         }
76.     }
77.
78.     public function update_code($id = null){
79.         $code = random_string("numeric",6);
80.         $from_email = "smarthomeiot141@gmail.com";
81.         $this->load->library('email');
82.         $config = array();
83.         $config['protocol'] = 'smtp';
84.         $config['smtp_host'] = 'ssl://smtp.gmail.com';
85.         $config['smtp_user'] = 'smarthomeiot141@gmail.
com';
86.         $config['smtp_pass'] = 'haeKal141';
87.         $config['smtp_port'] = 465;
88.         $this->email->initialize($config);
89.         $this->email->set_newline("\r\n");
90.         $this->email-
>from($from_email, 'Identification');
91.         $this->email->to($this->session-
>userdata('email'));
92.         $this->email-
>subject('Send Email Codeigniter');
93.         $this->email->message($code);
94.         $this->email->send();
95.         $this->db->where(['id_device' => $id]);
96.         $this->db-
>update('pintu',['qr_code' => $code]);
97.     }
98.
99. }

```

#### 4. Kode Program Jemuran.php

```
5. <?php
6. defined('BASEPATH') OR exit('No direct script access a
   llowed');
7.
8. class Jemuran extends CI_Controller{
9.
10.     function __construct(){
11.         parent::__construct();
12.         if($this->session-
   >userdata('status') != "Login"){
13.             redirect(base_url("login"));
14.         }
15.     }
16.
17.     function index(){
18.         $data['history'] = $this->db->get("jemuran")-
   >result();
19.         $view = array(
20.             $this->load->view('template/v_header'),
21.             $this->load-
   >view('content/jemuran/v_jemuran', $data),
22.             $this->load->view('template/v_footer')
23.         );
24.     }
25.     function update($id,$status_jemuran){
26.         $where = $check1 = array('id_jemuran' => $id);
27.         if($status_jemuran==0){
28.             $update = array(
29.                 'Status' => 1
30.             );
31.         }elseif($status_jemuran==1){
32.             $update = array(
```

```

33.             'Status' => 0
34.         );
35.     }
36.     $this->db->where($where);
37.     $this->db->update('jemuran',$update);
38.     redirect(base_url('jemuran'));
39.
40. }
41.
42.     public function test(){
43.         $devid = $this->input->post('devid');
44.         $data = $this->db-
>get_where("t_lock",array('t_devid'=> $devid))-
>row_array();
45.         $status = (int)($data['t_status']);
46.         if($status_jemuran==0){
47.             echo 'tutup';
48.         }elseif($status_jemuran==1){
49.             echo 'buka';
50.         }
51.     }
52. }

```

## 5. Kode Program Kipas.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access a
llowed');
3.
4. class Kipas extends CI_Controller{
5.
6.     function __construct(){
7.         parent::__construct();

```

```

8.         if($this->session-
>userdata('status') != "Login"){
9.             redirect(base_url("login"));
10.        }
11.    }
12.
13.    function index(){
14.        $data['kipas'] = $this->db->get("kipas")-
>result();
15.        $view = array(
16.            $this->load->view('template/v_header'),
17.            $this->load-
>view('content/kipas/v_kipas', $data),
18.            $this->load->view('template/v_footer')
19.        );
20.    }
21.    function update($id,$status_kipas){
22.        $where = $check1 = array('id_kipas' => $id);
23.        if($status_kipas==0){
24.            $update = array(
25.                'status_kipas' => 1
26.            );
27.        }elseif($status_kipas==1){
28.            $update = array(
29.                'status_kipas' => 0
30.            );
31.        }
32.        $this->db->where($where);
33.        $this->db->update('kipas',$update);
34.        redirect(base_url('kipas'));
35.
36.    }
37.
38.    function update_sensor($id,$sensor_kipas){
39.        $where = $check1 = array('id_kipas' => $id);
40.        if($sensor_kipas==0){
41.            $update = array(
42.                'sensor_kipas' => 1
43.            );
44.        }elseif($sensor_kipas==1){
45.            $update = array(
46.                'sensor_kipas' => 0
47.            );

```

```

48.     }
49.     $this->db->where($where);
50.     $this->db->update('kipas',$update);
51.     redirect(base_url('kipas'));
52. }
53.
54.     public function test(){
55.         $devid = $this->input->post('devid');
56.         $data = $this->db-
>get_where("t_lock",array('t_devid'=> $devid))-
>row_array();
57.         $status = (int)($data['t_status']);
58.         if($status_lampu==0){
59.             echo 'on';
60.         }elseif($status_lampu==1){
61.             echo 'off';
62.         }
63.     }
64. }

```

## 6. Kode Program Lampu.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access a
llowed');
3.
4. class Lampu extends CI_Controller{
5.
6.     function __construct(){
7.         parent::__construct();
8.         if($this->session-
>userdata('status') != "Login"){
9.             redirect(base_url("login"));
10.        }
11.    }
12.
13.    function index(){
14.        $data['lampu'] = $this->db->get("lampu")-
>result();

```

```

15.         $view = array(
16.             $this->load->view('template/v_header'),
17.             $this->load-
>view('content/lampu/v_lampu', $data),
18.             $this->load->view('template/v_footer')
19.         );
20.     }
21.     function update($id,$status_lampu){
22.         $where = $check1 = array('id_lampu' => $id);
23.         if($status_lampu==0){
24.             $update = array(
25.                 'status_lampu' => 1
26.             );
27.         }elseif($status_lampu==1){
28.             $update = array(
29.                 'status_lampu' => 0
30.             );
31.         }
32.         $this->db->where($where);
33.         $this->db->update('lampu',$update);
34.         redirect(base_url('lampu'));
35.     }
36. }
37.
38. function update_sensor($id,$sensor_lampu){
39.     $where = $check1 = array('id_lampu' => $id);
40.     if($sensor_lampu==0){
41.         $update = array(
42.             'sensor_lampu' => 1
43.         );
44.     }elseif($sensor_lampu==1){
45.         $update = array(
46.             'sensor_lampu' => 0
47.         );
48.     }
49.     $this->db->where($where);
50.     $this->db->update('lampu',$update);
51.     redirect(base_url('lampu'));
52. }
53.
54. public function test(){
55.     $devid = $this->input->post('devid');

```



```

56.         $data = $this->db-
>get_where("t_lock",array('t_devid'=> $devid))-
>row_array();
57.         $status = (int)($data['t_status']);
58.         if($status_lampu==0){
59.             echo 'on';
60.         }elseif($status_lampu==1){
61.             echo 'off';
62.         }
63.     }
64. }

```

## 7. Kode Program Log\_pintu.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access a
llowed');
3.
4. class Log_pintu extends CI_Controller{
5.
6.     function __construct(){
7.         parent::__construct();
8.         if($this->session-
>userdata('status') != "Login"){
9.             redirect(base_url("login"));
10.        }
11.    }
12.
13.    function index(){
14.        $data['control'] = $this->db-
>get("log_pintu")->result();
15.        $view = array(
16.            $this->load->view('template/v_header'),
17.            $this->load-
>view('content/Log/v_log_pintu', $data),
18.            $this->load->view('template/v_footer')
19.        );
20.    }
21.
22. }

```

## 8. Kode Program Login.php

```
1. <?php
2.
3. class Login extends CI_Controller{
4.
5.     function __construct(){
6.         parent::__construct();
7.         $this->load->model('m_login');
8.
9.     }
10.
11.     function index(){
12.         $this->load->view('content/Login/v_login');
13.     }
14.
15.     function aksi_login(){
16.         $username = $this->input->post('username');
17.         $password = $this->input->post('password');
18.         $where = array(
19.             'username' => $username,
20.             'password' => md5($password)
21.         );
22.         $cek = $this->m_login->cek_login("user",$where)->row();
23.         if($cek > 0){
24.
25.             $data_session = array(
26.                 'nama' => $username,
27.                 'status' => "Login",
28.                 'id_user' => $cek->id_user,
29.                 'email' => $cek->email,
30.                 'role' => $cek->role,
31.             );
32.
33.             $this->session->set_userdata($data_session);
34.
35.             redirect(base_url("control"));
36.
37.         }else{
38.             echo "Username dan password salah !";
```

```

39.     }
40. }
41.
42. function logout(){
43.     $this->session->sess_destroy();
44.     redirect(base_url('login'));
45. }
46. }

```

## 9. Kode Program Sensor.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access a
  llowed');
3.
4. class Sensor extends CI_Controller{
5.
6.     function __construct(){
7.         parent::__construct();
8.     }
9.
10.    function sethujan(){
11.        $hujan = $this->input->get('sensorhujan');
12.        $fsta = file_get_contents("application/views/c
  ontent/cahaya.json");
13.        $isihujan = json_decode($fsta,true);
14.        $isihujan['valldr'] = $hujan;
15.        file_put_contents('application/views/content/c
  ahaya.json',json_encode($isihujan));
16.    }
17.
18.    function getdata(){
19.        $index = '1';
20.        $where = array(
21.            'id_micro' => $index
22.        );
23.        $pintu = $this->db-
  >get_where("pintu",array('id_device' => '51001'))-
  >row_array();

```

```

24.         $kipas = $this->db-
>get_where("kipas",array('id_kipas' => '1'))-
>row_array();
25.         $lampu = $this->db-
>get_where("lampu",array('id_lampu' => '1'))-
>row_array();
26.
27.         $stpintu = $pintu['Status'];
28.         $stkipas = $kipas['status_kipas'];
29.         $stlampu = $lampu['status_lampu'];
30.         $sensorlp = $lampu['sensor_lampu'];
31.         $sensorkp = $kipas['sensor_kipas'];
32.         $result = $this->db->get("jemuran", $where)-
>row_array();
33.         $fsta = file_get_contents("application/views/c
ontent/cahaya.json");
34.         $data = json_decode($fsta,true);
35.         echo $data['valldr'].'.$result['status_jemur
an'].'.$stpintu.'.$stkipas.'.$stlampu.'.$senso
rlp.'.$sensorkp;
36.     }
37.
38.     function setjemuran(){
39.         $id = 1;
40.         $status = $this->input->get('jemuran');
41.         $update = array(
42.             'status_jemuran' => $status
43.         );
44.         $where = array(
45.             'id_jemuran' => $id
46.         );
47.         $this->db->where($where)-
>update("jemuran",$update);
48.     }
49.
50.     function updatelampu($id){
51.         $status_lampu = $this->input->get('status');
52.         $where = array('id_lampu' => $id);
53.         $update = array(
54.             'status_lampu' => $status_lampu
55.         );
56.         $this->db->where($where);
57.         $this->db->update('lampu',$update);

```

```

58.     }
59.
60.     function updatekipas($id){
61.         $status_kipas = $this->input->get('status');
62.         $where = array('id_kipas' => $id);
63.         $update = array(
64.             'status_kipas' => $status_kipas
65.         );
66.         $this->db->where($where);
67.         $this->db->update('kipas',$update);
68.     }
69. }

```

## 10. Kode Program Arduino

```

11. #include "DHT.h"
12. #include <ESP8266WiFi.h>
13. #include <WiFiClient.h>
14. #include <ESP8266WebServer.h>
15. #include <ESP8266HTTPClient.h>
16. #include <Servo.h>
17. #define INPUT_SIZE 15
18. #define DHTPIN D2 // Data di pin A0
19.
20. #define DHTTYPE DHT11 // yang kita gunakan adalah DHT 11
21. // #define DHTTYPE DHT22 // kalau pakai DHT 22 silahkan uncomment
22. const char *ssid = "Rumah yang Yahud"; //ENTER YOUR WIFI SETTINGS
23. const char *password = "haikalkucing";
24. String URL_ADDRESS="http://192.168.0.16/smarthome/sensor/";
25.
26. DHT dht(DHTPIN, DHTTYPE); //Deklarasi objek dht menggunakan class DHT pada library

```

```
27.
28. char input[INPUT_SIZE+1];
29. byte size = Serial.readBytes(input, INPUT_SIZE);
30. // Add the final 0 to end the C string
31. char *token;
32. const char s[2] = ",";
33.
34. int angle = 0;
35. float h;
36. float t;
37. String cahaya;
38. String hujan;
39. String jemuran;
40. String temperature;
41. String humidity;
42. int pintu, lampu, kipas, sensorlp, sensorkp;
43. int chylp, shkp;
44. int servoPin = D7;
45. unsigned long currTime;
46. unsigned long prevTime;
47. int durasi;
48. int in1 = D4;
49. int in2 = D3;
50. int in3 = D5;
51. Servo servo;
52. void setup() {
53.   Serial.begin(115200);
54.   // Serial.println("DHTxx test!");
55.   connectWiFi();
56.   servo.attach(servoPin);
57.   dht.begin();
58.   pinMode(in1, OUTPUT);
59.   pinMode(in2, OUTPUT);
```

```
60. pinMode(in3, OUTPUT);
61. digitalWrite(in1, HIGH);
62. digitalWrite(in2, HIGH);
63. digitalWrite(in3, HIGH);
64. }
65.
66.
67. void loop() {
68.   prevTime = millis();
69.   getData("getdata");
70.   packData();
71.   if(sensorlp==1)getCahaya();
72.   if(sensorkp==1)getDHT();
73.   fuzzy(jemuran, hujan, cahaya, humidity);
74.   Serial.print("Jemuran: ");
75.   Serial.println(jemuran);
76.   printSensor();
77.   setservo(jemuran);
78.   setPerangkat(pintu, kipas, lampu, sensorlp, sensorkp
);
79.   Serial.println("");
80.   currTime= millis();
81.   durasi = cekdurasi(currTime);
82.   print_time(durasi);
83.   delay(7000); // delay 2 detik
84. }
85.
86. void getCahaya(){
87.   unsigned int AnalogValue;
88.   AnalogValue = analogRead(A0);
89.   if(AnalogValue<850){
90.     cahaya = "terang";
91.     chylp=0;
```

```
92.         }else{
93.             cahaya = "gelap";
94.             chylp=1;
95.         }
96.         Serial.println(AnalogValue);
97.     }
98.
99. void getDHT(){
100.     // Baca kelembapan udara dalam satuan persen
101.     (%)
102.     h = dht.readHumidity();
103.     // Baca suhu udara dalam satuan Celsius (default)
104.     t = dht.readTemperature();
105.
106.     if(h>80){humidity = "lembab";}
107.     else{humidity = "kering";}
108.
109.     if(t<28){
110.         temperature = "dingin";
111.         shkp=0;
112.     }
113.     else{
114.         temperature = "panas";
115.         shkp=1;
116.     }
117. }
118. void setPerangkat(int pnt, int kps, int lmp, int
119.     t srlp, int srkp){
120.     Serial.print("Pintu: ");
121.     if(pnt==0){
122.         Serial.println("BUKA");
123.         digitalWrite(in2, LOW);
```



```
123.         }else{
124.             Serial.println("TUTUP");
125.             digitalWrite(in2, HIGH);
126.         }
127.
128.         if(srkp==0){
129.             Serial.println("Kipas: SENSOR MATI");
130.             Serial.print("Kipas: ");
131.             if(kps==1){
132.                 Serial.println("ON");
133.                 digitalWrite(in3, LOW);
134.             }else{
135.                 Serial.println("OFF");
136.                 digitalWrite(in3, HIGH);
137.             }
138.         }else if(srkp==1){
139.             Serial.println("Kipas: SENSOR HIDUP");
140.             Serial.print("Kipas: ");
141.             if(shkp==0){
142.                 Serial.println("OFF");
143.                 // Serial.print("digin");
144.                 digitalWrite(in3, HIGH);
145.                 getData("updatekipas/1?status=0");
146.             }else if(shkp==1){
147.                 Serial.println("ON");
148.                 // Serial.print("panas");
149.                 digitalWrite(in3, LOW);
150.                 getData("updatekipas/1?status=1");
151.             }
152.         }
153.
154.         if(sr1p==0){
155.             Serial.println("Lampu: SENSOR MATI");
```

```
156.         Serial.print("Lampu: ");
157.         if(lmp==1){
158.             Serial.println("ON");
159.             digitalWrite(in1, LOW);
160.         }else{
161.             Serial.println("OFF");
162.             digitalWrite(in1, HIGH);
163.         }
164.     }else if(srlp==1){
165.         Serial.println("Lampu: SENSOR HIDUP");
166.         Serial.print("Lampu: ");
167.         if(chylp==0){
168.             Serial.println("OFF");
169.             //     Serial.print("terang");
170.             digitalWrite(in1, HIGH);
171.             getData("updatelampu/1?status=0");
172.         }else if(chylp==1){
173.             Serial.println("ON");
174.             //     Serial.print("gelap");
175.             digitalWrite(in1, LOW);
176.             getData("updatelampu/1?status=1");
177.         }
178.     }
179. }
180.
181. void fuzzy(String jmr, String hjn, String chy,
    String hmd){
182.     if(hjn=="WET"){
183.         delay(200);
184.         jemuran="0";
185.         getData("setjemuran?jemuran="+jemuran);
186.         return;
187.     }
```

```
188.
189.     if(chy=="gelap"){
190.         delay(200);
191.         jemuran="0";
192.         getData("setjemuran?jemuran="+jemuran);
193.         return;
194.     }
195.
196.     if((chy=="terang") && (hjn=="DRY")){
197.         delay(200);
198.         jemuran="1";
199.         getData("setjemuran?jemuran="+jemuran);
200.         return;
201.     }
202. }
203.
204. void setservo(String jmr){
205.     if(jmr=="1"){
206.         while(angle<=180){
207.             servo.write(angle);
208.             delay(20);
209.             angle++;
210.         }
211.     }else if(jmr=="0"){
212.         while(angle>=0){
213.             servo.write(angle);
214.             delay(20);
215.             angle--;
216.         }
217.     }
218. }
219.
```

```

220.     // -----
        Function Connect to WiFi and MySQL
221.     void connectWiFi(){
222.         WiFi.mode(WIFI_OFF);           //Prevents reconn
        ection issue (taking too long to connect)
223.         delay(1000);
224.         WiFi.mode(WIFI_STA);           //This line hides
        the viewing of ESP as wifi hotspot
225.
226.         WiFi.begin(ssid, password);     //Connect to
        your WiFi router
227.         Serial.println("");
228.
229.         Serial.print("Connecting");
230.         // Wait for connection
231.         while (WiFi.status() != WL_CONNECTED) {
232.             delay(500);
233.             Serial.print(".");
234.         }
235.
236.         //If connection successful show IP address in
        serial monitor
237.         Serial.println("");
238.         Serial.print("Connected to ");
239.         Serial.println(ssid);
240.         Serial.print("IP address: ");
241.         Serial.println(WiFi.localIP()); //IP address
        assigned to your ESP
242.     }
243.
244.     void getData(String data){
245.         WiFiClient client;
246.         HTTPClient https;
247.
248.         //         Serial.print("[HTTPS] begin...\n");

```

```
249.         if (https.begin(client, URL_ADDRESS+dat
250.             a)) { // HTTPS
251.                 // Serial.print("[HTTPS] GET...\n");
252.                 int httpCode = https.GET();
253.                 if (httpCode > 0) {
254.                     // Serial.printf("[HTTPS] GET... cod
255.                         e: %d\n", httpCode);
256.                     if (httpCode == HTTP_CODE_OK || htt
257.                         pCode == HTTP_CODE_MOVED_PERMANENTLY) {
258.                         String payload = https.getString(
259.                             );
260.                         payload.toCharArray(input, sizeof
261.                             (input));
262.                     }
263.                     } else {
264.                         Serial.printf("[HTTPS] GET... failed,
265.                             error: %s\n", https.errorToString(httpCode).c_str());
266.                     }
267.                     https.end();
268.                 } else {
269.                     Serial.printf("[HTTPS] Unable to conn
270.                         ect\n");
271.                 }
272.             }
273.         }
274.         void packData(){
275.             token = strtok(input, ",");
276.             hujan = token;
277.             token = strtok(0, ",");
278.             jemuran = token;
279.             token = strtok(0, ",");
280.             pintu = atoi(token);
281.             token = strtok(0, ",");
```

```
276.         kipas = atoi(token);
277.         token = strtok(0, ",");
278.         lampu = atoi(token);
279.         token = strtok(0, ",");
280.         sensorlp = atoi(token);
281.         token = strtok(0, ",");
282.         sensorkp = atoi(token);
283.     }
284.
285.     void printSensor(){
286.         Serial.print("Hujan: ");
287.         Serial.println(hujan);
288.         Serial.print("Cahaya: ");
289.         Serial.println(cahaya);
290.         Serial.print("Humidity: ");
291.         Serial.println(humidity);
292.         Serial.print("Temperature: ");
293.         Serial.println(temperature);
294.         Serial.print("Pintu: ");
295.         Serial.println(pintu);
296.         Serial.print("Kipas: ");
297.         Serial.println(kipas);
298.         Serial.print("Lampu: ");
299.         Serial.println(lampu);
300.         Serial.print("Sensor Kipas: ");
301.         Serial.println(sensorkp);
302.         Serial.print("Sensor Lampu: ");
303.         Serial.println(sensorlp);
304.     }
305.
306.     long cekdurasi(unsigned long time_millis){
307.         durasi = time_millis - prevTime;
308.         return durasi;
```

```
309.     }  
310.  
311.     void print_time(unsigned long time_millis){  
312.         Serial.print("Response Time: ");  
313.         Serial.print(time_millis/1000.0);  
314.         Serial.println("s");  
315.     }
```

***[Halaman ini sengaja dikosongkan]***



## BIODATA PENULIS



**Mohammad Haekal Alawy**, lahir di Cirebon tanggal 9 Juni 1998. Penulis merupakan anak pertama dari dua bersaudara. Penulis telah menempuh pendidikan formal di SD Kartika Siliwangi VII (2004-2010), SMP Negeri 1 Kota Cirebon (2010-2013), SMA Negeri 1 Kota Cirebon (2013-2016). Penulis melanjutkan studi dengan berkuliah pada program sarjana (S1) di Departemen Informatika ITS.

Selama kuliah di Informatika ITS, penulis mengambil bidang minat Arsitektur dan Jaringan Komputer (AJK). Selama menempuh perkuliahan Penulis aktif mengikuti organisasi kemahasiswaan, yaitu sebagai wakil kepala departement PSDM di UKM musik ITS. Selain aktif mengikuti organisasi, penulis juga aktif dalam kegiatan kepanitiaan Schematics, yaitu sebagai staff Reeva pada tahun 2017 dan menjadi BPH 2 Reeva tahun 2019. Penulis dapat dihubungi melalui surel di [mhaekalalawy15@gmail.com](mailto:mhaekalalawy15@gmail.com).