



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - IS184853

**RANCANG BANGUN SISTEM SAKELAR OTOMATIS
BERBASIS ARDUINO DENGAN MENGGUNAKAN
SENSOR SUHU DAN KELEMBAPAN (STUDI KASUS:
PT. SIN A SIXFIFTEEN)**

**DESIGN AND DEVELOPMENT OF AUTOMATIC
SWITCH SYSTEM BASED ON ARDUINO USING
TEMPERATURE AND HUMIDITY SENSOR (STUDY
CASE: PT. SIN A SIXFIFTEEN)**

**ADITYA BUDI LAKSONO
NRP 05211640000003**

**Dosen Pembimbing
Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom.**

**DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Elektro Dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020**

TUGAS AKHIR - IS184853

**RANCANG BANGUN SISTEM SAKELAR
OTOMATIS BERBASIS ARDUINO DENGAN
MENGUNAKAN SENSOR SUHU DAN
KELEMBAPAN (STUDI KASUS: PT. SIN A
SIXFIFTEEN)**

**ADITYA BUDI LAKSONO
NRP 0521164000003**

**Dosen Pembimbing
Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom.**

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

UNDERGRADUATE THESIS - IS184853

DESIGN AND DEVELOPMENT OF AUTOMATIC SWITCH SYSTEM BASED ON ARDUINO USING TEMPERATURE AND HUMIDITY SENSOR (STUDY CASE: PT. SIN A SIXFIFTEEN)

ADITYA BUDI LAKSONO
NRP 05211640000003

Supervisor
Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom.

DEPARTMENT OF INFORMATION SYSTEMS
Faculty of Electrical and Intelligent Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2020

LEMBAR PENGESAHAN**Rancang Bangun Sistem Sakelar Otomatis Berbasis Arduino
dengan Menggunakan Sensor Suhu dan Kelembapan (Studi
Kasus: PT. Sin A Sixfifteen)****TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer (S.Kom)

pada

Departemen Sistem Informasi

Fakultas Teknologi Elektro dan Informatika Cerdas (ELECTICS)

Institut Teknologi Sepuluh Nopember

Oleh

Aditya Budi Laksono

05211640000003

Surabaya, 14 Agustus 2020

Kepala Departemen Sistem Informasi

Dr. Mudjabidin, ST., MT.

NIP. 197010102003121001



LEMBAR PERSETUJUAN

RANCANG BANGUN SISTEM SAKELAR OTOMATIS BERBASIS ARDUINO DENGAN MENGGUNAKAN SENSOR SUHU DAN KELEMBAPAN (STUDI KASUS: PT. SIN A SIXFIFTEEN)

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh :

ADITYA BUDI LAKSONO
NRP. 0521164000003

Disetujui Tim Penguji : Tanggal Ujian : 12 Juni 2020
Periode Wisuda : September 2020

Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom. (Pembimbing I)

Nisfu Asrul Sani, S.Kom., M.Sc. (Penguji I)

Bekti Cahyo Hidayanto, S.Si., M. Kom. (Penguji II)

RANCANG BANGUN SISTEM SAKELAR OTOMATIS BERBASIS ARDUINO DENGAN MENGGUNAKAN SENSOR SUHU DAN KELEMBAPAN (STUDI KASUS: PT. SIN A SIXFIFTEEN)

Nama : Aditya Budi Laksono
NRP : 05211640000003
Departemen : Sistem Informasi FTEIC-ITS
Pembimbing : Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom.

ABSTRAK

Otomasi proses bisnis banyak dilakukan oleh perusahaan karena dapat mengurangi dua hal penting dalam sebuah proses bisnis, yaitu waktu dan biaya. PT. Sin A Sixfifteen yang merupakan sebuah perusahaan yang memproduksi permen ingin memanfaatkan teknologi Internet of Things (IoT) ke dalam salah satu proses bisnis mereka, yaitu melakukan pengawasan dan pengendalian suhu dan kelembapan pada tiap ruangan krusial yang ada dalam pabrik. Proses bisnis tersebut menjadi salah satu faktor dalam memastikan jaminan kualitas, baik kualitas produk yang telah jadi ataupun kualitas bahan baku untuk produksi. Saat ini proses bisnis tersebut dilakukan secara manual dimana seseorang akan memeriksa dan mencatat nilai suhu pada tiap ruangan secara berkala. Selain itu pengawasan hanya dilakukan pada variabel suhu dan belum ada variabel kelembapan. Dengan menggunakan teknologi Internet of Things (IoT), proses bisnis dapat dilakukan secara otomatis. Penerapan teknologi Internet of Things (IoT) ke dalam proses bisnis tersebut dilakukan dengan membuat sebuah sistem pengawasan suhu dan kelembapan dengan menggunakan Arduino sebagai mikrokontroler yang dihubungkan ke sensor suhu dan kelembapan. Tidak hanya sensor suhu dan kelembapan, sebuah sakelar yang terhubung dengan relai juga dihubungkan ke Arduino sebagai alat

pengendalian atau tindak lanjut otomatis jika suhu dan atau kelembapan melebihi batas yang diinginkan dengan memasang sebuah alat lain yang dapat mengubah suhu dan atau kelembapan ke dalam sakelar tersebut. Agar data suhu dan kelembapan dapat dibaca oleh pengguna serta dapat mengubah batas suhu dan atau kelembapan untuk mengatur sakelar otomatis maka Arduino dihubungkan ke aplikasi melalui pengiriman data melalui internet. Hasil dari tugas akhir ini adalah sebuah sistem yang dapat menyimpan dan mengolah informasi suhu dan kelembapan ruangan secara otomatis serta dapat mengatur sakelar secara otomatis untuk mengatur alat pengatur suhu dan kelembapan berdasarkan kondisi yang telah diatur sehingga tidak perlu lagi melakukan pencatatan dan tindak lanjut secara manual.

Kata Kunci: Internet of Things, Suhu, Kelembapan, Sakelar Otomatis, Arduino

DESIGN AND DEVELOPMENT OF AUTOMATIC SWITCH SYSTEM BASED ON ARDUINO USING TEMPERATURE AND HUMIDITY SENSOR (STUDY CASE: PT. SIN A SIXFIFTEEN)

Name : Aditya Budi Laksono
NRP : 0521164000003
Department : Information Systems ELECTICS-ITS
Supervisor : Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom.

ABSTRACT

Business process automation is mostly done by companies because it can reduce two important things, which is time and cost. PT. Sin A Sixfifteen, which is a company that produces sweets, wants to utilize Internet of Things (IoT) technology in one of their business processes, which is monitoring and controlling temperature and humidity in every crucial room in the factory. The business process is one of the factors in ensuring quality assurance, both the quality of the finished product or the quality of raw materials for production. Currently the business process is done manually where someone will check and record the temperature value in each room periodically. In addition, supervision is only carried out on the temperature variable. By using Internet of Things (IoT) technology, business processes can be done automatically. The application of Internet of Things (IoT) technology to the business process is carried out by creating a temperature and humidity monitoring system using Arduino as a microcontroller that is connected to the temperature and humidity sensors. Not only temperature and humidity sensors, a switch connected to the relay is also connected to Arduino as an automatic control or follow-up if the temperature and or humidity exceeds the desired limit by attaching another device that can change the temperature and or humidity into the switch so that the

temperature and humidity data can be read by the user and can change the temperature and / or humidity limits to set an automatic switch, Arduino is connected to the application via data transmission via the internet. The final result is a system that can store and process room temperature and humidity information automatically and can set switches automatically to regulate temperature and humidity control devices based on the conditions that have been set up so that there is no need to record and follow up manually.

Keywords: Internet of Things, Temperature, Humidity, Automatic Switch, Arduino

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertandatangan di bawah ini:

Nama : Aditya Budi Laksono
NRP : 0521164000003
Tempat/Tanggal lahir : Serang/16 Juli 1998
Fakultas/Departemen : Fakultas Teknologi Elektro dan
Informatika Cerdas/Sistem Informasi
Nomor Telp/Hp/email : adittbudi@gmail.com

Dengan ini menyatakan dengan sesungguhnya bahwa penelitian/makalah/tugas akhir saya yang berjudul:

RANCANG BANGUN SISTEM SAKELAR OTOMATIS BERBASIS ARDUINO DENGAN MENGGUNAKAN SENSOR SUHU DAN KELEMBAPAN (STUDI KASUS: PT. SIN A SIXFIFTEEN)

Bebas Dari Plagiarisme Dan Bukan Hasil Karya Orang Lain.

Apabila dikemudian hari ditemukan seluruh atau sebagian penelitian/makalah/tugas akhir tersebut terdapat indikasi plagiarisme, maka saya bersedia menerima sanksi sesuai peraturan dan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sesungguhnya dan untuk dipergunakan sebagaimana mestinya.

Surabaya, 4 Agustus 2020



Aditya Budi Laksono
NRP. 0521164000003

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT, Tuhan Semesta Alam yang telah memberikan kekuatan serta hidayah-Nya kepada penulis sehingga penulis dapat menyelesaikan tugas akhir ini yang merupakan salah satu syarat kelulusan di Departemen Sistem Informasi Fakultas Teknologi Elektro dan Informatika Cerdas Institut Teknologi Sepuluh Nopember Surabaya.

Terima kasih penulis sampaikan kepada pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik berupa materil maupun moril demi tercapainya tujuan pembuatan tugas akhir ini. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada :

1. Segenap keluarga besar terutama kedua orang tua dan adik penulis, Bapak Bandel Sutrisno, Ibu Riani Yanuarsih, dan Alissa Rachmasanti yang senantiasa mendoakan, memberikan motivasi dan semangat, sehingga penulis mampu menyelesaikan pendidikan sarjana ini dengan baik.
2. Dr. Mudjahidin, S.T., M.T. selaku Kepala Departemen Sistem Informasi ITS, Bapak Ahmad Mukhlason, S.Kom., M.Sc., Ph.D. selaku Ketua Program Studi Sarjana Departemen Sistem Informasi ITS, serta seluruh dosen pengajar beserta staf dan karyawan Departemen Sistem Informasi ITS selama penulis menjalani perkuliahan.
3. Bapak Tony Dwi Susanto, S.T., M.T., Ph.D. sebagai dosen wali penulis selama menempuh pendidikan di Departemen Sistem Informasi ITS.
4. Bapak Dr.Eng. Febriliyan Samopa, S. Kom., M. Kom. selaku dosen pembimbing yang telah banyak meluangkan waktu untuk membimbing, mengarahkan, dan mendukung dengan memberikan ilmu, petunjuk, dan motivasi dalam penyelesaian tugas akhir ini.

5. Bapak Nisfu Asrul Sani, S. Kom., M.Sc. dan Bapak Bakti Cahyo Hidayanto, S.Si., M. Kom. selaku dosen penguji yang telah memberikan kritik dan saran dalam penyempurnaan tugas akhir ini.
6. Teman-teman Sistem Informasi angkatan 2016 (Artemis) yang senantiasa menemani dan memberikan motivasi bagi penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir ini.
7. Serta seluruh pihak-pihak lain yang tidak dapat disebutkan satu per satu, yang telah banyak membantu penulis selama perkuliahan hingga dapat menyelesaikan Tugas Akhir ini.

Penyusunan laporan tugas akhir ini masih jauh dari kata sempurna, sehingga penulis menerima adanya kritik maupun saran yang membangun untuk perbaikan di masa yang akan datang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, 10 Juni 2020

Penulis

Aditya Budi Laksono

DAFTAR ISI

ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xvii
DAFTAR KODE	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Relevansi	5
BAB II TINJAUAN PUSTAKA	7
2.1 Studi Literatur	7
2.2 Dasar Teori	9
2.2.1 Arduino	9
2.2.2 Relai	10
2.2.3 Suhu dan Kelembapan Ruangan	11
2.2.4 MySQL	11
2.2.5 CodeIgniter	12
2.2.6 DHT22	12
2.2.7 Wemos D1 Mini	13
2.2.8 Kondisi Lingkungan PT. Sin A Sixfifteen	14
BAB III METODOLOGI	17
3.1 Studi Literatur	18
3.2 Analisis	18
3.3 Desain	18
3.3.1 Desain Perangkat Keras	18
3.3.2 Desain Perangkat Lunak	19
3.4 Implementasi	19
3.4.1 Implementasi Perangkat Keras	19
3.4.2 Implementasi Perangkat Lunak	19
3.5 Testing	19

3.6 Dokumentasi Tugas Akhir	20
BAB IV PERANCANGAN	21
4.1 Analisis Kebutuhan	21
4.2 Desain Sistem.....	24
4.3 Desain Perangkat Keras	26
4.3.1 Skema Perangkat Keras.....	27
4.3.2 Desain Program Arduino.....	29
4.4 Desain Perangkat Lunak	32
4.4.1 Desain Basis Data	32
4.4.2 <i>Use Case</i> Aplikasi Web	37
4.4.3 Desain Aplikasi Web.....	43
4.4.4 <i>Class Diagram</i> Aplikasi Web	49
4.4.5 Desain <i>Routing</i> Aplikasi Web	50
4.5 Desain <i>Testing</i> Perangkat Keras.....	51
4.6 Desain <i>Testing</i> Perangkat Lunak.....	51
BAB V IMPLEMENTASI	55
5.1 Lingkungan Implementasi Sistem.....	55
5.2 Implementasi Perangkat Keras.....	55
5.2.1 Lingkungan Implementasi Perangkat Keras	55
5.2.2 Implementasi Rangkaian Perangkat Arduino.....	56
5.2.3 Implementasi Program Perangkat Arduino	57
5.3 Implementasi Perangkat Lunak.....	61
5.3.1 Lingkungan Implementasi Perangkat Lunak.....	62
5.3.2 Implementasi Basis Data.....	62
5.3.3 Implementasi Aplikasi Web	63
5.3.3.1 Implementasi <i>Model</i>	63
5.3.3.2 Implementasi <i>Controller</i>	66
5.3.3.3 Implementasi <i>View</i>	77
5.3.3.4 Implementasi <i>Routing</i>	81
5.4 <i>Testing</i> Sistem	82
5.4.1 <i>Testing</i> Perangkat Keras.....	82
5.4.2 <i>Testing</i> Perangkat Lunak.....	85
BAB VI HASIL DAN PEMBAHASAN.....	89
6.1 Arsitektur Sistem.....	89
6.2 Komunikasi Perangkat Arduino	90
6.3 Program Perangkat Arduino.....	92
BAB VII KESIMPULAN DAN SARAN	97

8.1 Kesimpulan	97
8.2 Saran	97
DAFTAR PUSTAKA	99
BIODATA PENULIS	101

DAFTAR GAMBAR

Gambar 2. 1 Struktur Sederhana Relai	10
Gambar 2. 2 Flowchart Cara Kerja CodeIgniter	12
Gambar 2. 3 Dimensi DHT22 dalam Milimeter.....	13
Gambar 2. 4 Board Wemos D1 Mini.....	14
Gambar 3. 1 Tahapan Pengerjaan Tugas Akhir.....	17
Gambar 4. 1 Desain Sistem	25
Gambar 4. 2 Alur Perangkat Arduino dengan Aplikasi Web .	26
Gambar 4. 3 Desain Skema Perangkat Arduino	29
Gambar 4. 4 Desain Alur Program Perangkat Arduino.....	31
Gambar 4. 5 Class Diagram Program Perangkat Arduino.....	32
Gambar 4. 6 Desain Basis Data.....	32
Gambar 4. 7 Diagram <i>Use Case</i> Aplikasi Web.....	38
Gambar 4. 8 <i>Class Diagram</i> Aplikasi Web.....	49
Gambar 4. 9 Desain <i>Routing</i> Aplikasi Web	50
Gambar 5. 1 Implementasi Rangkaian Perangkat Arduino	57
Gambar 5. 2 Implementasi Arsitektur Aplikasi Web	63
Gambar 5. 3 <i>User Interface View</i> dashboard.....	77
Gambar 5. 4 <i>User Interface View</i> device.....	78
Gambar 5. 5 <i>User Interface View</i> devicemanagement	79
Gambar 5. 6 <i>User Interface View</i> deviceThreshold	80
Gambar 5. 7 <i>User Interface View</i> login.....	80
Gambar 5. 8 <i>User Interface View</i> usermanagement	81
Gambar 6. 1 Arsitektur Sistem	89
Gambar 6. 2 Daftar Perangkat Arduino yang Terdaftar	91
Gambar 6. 3 <i>Update</i> Data dengan <i>Key</i> Terdaftar	91
Gambar 6. 4 <i>Update</i> Data dengan <i>Key</i> Tidak Terdaftar.....	92
Gambar 6. 5 <i>Serial Monitor</i> Perangkat Arduino	92
Gambar 6. 6 <i>Fingerprint</i> Domain Web.....	93
Gambar 6. 7 Pengiriman Data Protokol HTTP	95
Gambar 6. 8 Pengiriman Data Protokol HTTPS	95

DAFTAR TABEL

Tabel 2. 1 Studi Literatur	7
Tabel 2. 2 Ruangan Studi Kasus	14
Tabel 4. 1 Pembuatan Keputusan	21
Tabel 4. 2 Kebutuhan Fungsional	22
Tabel 4. 3 Kebutuhan Non Fungsional.....	24
Tabel 4. 4 Pin DHT22 pada Wemos D1 Mini	27
Tabel 4. 5 Penjelasan Pin Relai	28
Tabel 4. 6 Deskripsi Tiap Tabel Basis Data	33
Tabel 4. 7 Deskripsi Kolom Tabel device.....	34
Tabel 4. 8 Deskripsi Kolom Tabel monitoring_data	34
Tabel 4. 9 Deskripsi Kolom Tabel kalibrasi_log	35
Tabel 4. 10 Deskripsi Kolom Tabel device_switch	35
Tabel 4. 11 Deskripsi Kolom Tabel device_threshold	35
Tabel 4. 12 Deskripsi Kolom Tabel device_switch_threshold	36
Tabel 4. 13 Deskripsi Kolom Tabel board	36
Tabel 4. 14 Deskripsi Kolom Tabel board_pin_switch.....	36
Tabel 4. 15 Deskripsi Kolom Tabel user.....	37
Tabel 4. 16 Deskripsi Kolom Tabel user_log	37
Tabel 4. 17 <i>Use Case Description 1</i>	38
Tabel 4. 18 <i>Use Case Description 2</i>	39
Tabel 4. 19 <i>Use Case Description 3</i>	39
Tabel 4. 20 <i>Use Case Description 4</i>	40
Tabel 4. 21 <i>Use Case Description 5</i>	40
Tabel 4. 22 <i>Use Case Description 6</i>	41
Tabel 4. 23 <i>Use Case Description 7</i>	41
Tabel 4. 24 <i>Use Case Description 8</i>	42
Tabel 4. 25 Daftar <i>Model</i> pada Aplikasi Web.....	43
Tabel 4. 26 Daftar <i>Controller</i> pada Aplikasi Web	44
Tabel 4. 27 Daftar <i>View</i> pada Aplikasi Web	44
Tabel 4. 28 Fungsi pada <i>Controller</i> AdminUser	45
Tabel 4. 29 Fungsi pada <i>Controller</i> AdminDevice	46
Tabel 4. 30 Fungsi pada <i>Controller</i> Api	46
Tabel 4. 31 Fungsi pada <i>Controller</i> Dashboard	46
Tabel 4. 32 Fungsi pada <i>Controller</i> Device	47
Tabel 4. 33 Fungsi pada <i>Controller</i> DeviceThreshold	48
Tabel 4. 34 Fungsi pada <i>Controller</i> UserSession.....	48

Tabel 4. 35 <i>Test Case</i> Perangkat Keras	51
Tabel 4. 36 <i>Test Case</i> Perangkat Lunak	52
Tabel 5. 1 Spesifikasi Komputer Implementasi	55
Tabel 5. 2 Teknologi Pengembangan Perangkat Arduino	55
Tabel 5. 3 Teknologi Pengembangan Aplikasi Web	62
Tabel 5. 4 Hasil <i>Test Case</i> Perangkat Keras.....	83
Tabel 5. 5 Pengujian Jarak Akurasi Sensor DHT22.....	84
Tabel 5. 6 Pengujian Pin Wemos D1 Mini untuk Relai	84
Tabel 5. 7 Hasil <i>Test Case</i> Perangkat Lunak.....	85

DAFTAR KODE

Kode 5. 1 <i>Global Variable</i> Program Arduino	58
Kode 5. 2 <i>Pseudocode</i> setup() Perangkat Arduino.....	58
Kode 5. 3 <i>Pseudocode</i> loop() Program Arduino	59
Kode 5. 4 <i>Pseudocode</i> connectWifi() Program Arduino	60
Kode 5. 5 <i>Pseudocode</i> readSensor() Program Arduino.....	60
Kode 5. 6 <i>Pseudocode</i> relayUpdate() Program Arduino.....	61
Kode 5. 9 <i>Pseudocode Model</i> “Device_model”	65
Kode 5. 10 <i>Pseudocode Model</i> “User_model”	66
Kode 5. 11 <i>Pseudocode Controller</i> AdminDevice.....	67
Kode 5. 12 <i>Pseudocode Controller</i> AdminUser	69
Kode 5. 13 <i>Pseudocode Controller</i> Api	70
Kode 5. 14 <i>PseudoCode Controller</i> Dashboard.....	71
Kode 5. 15 <i>Pseudocode Controller</i> Device.....	73
Kode 5. 16 <i>Pseudocode Controller</i> DeviceThreshold.....	75
Kode 5. 17 <i>Pseudocode Controller</i> UserSession	76
Kode 5. 18 Implementasi <i>Routing</i> pada Aplikasi Web	82
Kode 6. 1 Pengaturan <i>Host</i> Program Perangkat Arduino.....	94

Halaman ini sengaja dikosongkan

BAB I PENDAHULUAN

Pada bagian ini akan dijelaskan tentang pendahuluan dari tugas akhir yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan manfaat penelitian yang akan diperoleh dari penelitian tugas akhir ini.

1.1 Latar Belakang

Saat ini Indonesia sedang berusaha untuk menerapkan revolusi Industri 4.0 ke setiap bidang industri yang ada di Indonesia. Industri 4.0 adalah sebuah istilah dalam sebuah perindustrian dimana pelaku industri menggunakan komputer yang saling berkomunikasi satu sama lain untuk membuat keputusan dalam proses bisnis tanpa adanya keterlibatan manusia. Membuat sebuah pabrik pintar adalah tujuan akhir dari Industri 4.0 dimana pabrik menggunakan konsep *Internet of Things* (IoT), *Cloud Computing*, *Cyber Security*, *Big Data*, serta konsep-konsep lain yang berkembang sehingga proses bisnis pelaku industri dapat terbantu oleh teknologi [1].

PT. Sin A Sixfifteen adalah sebuah perusahaan produksi kembang gula yang terletak di Pasuruan Jawa Timur. Perusahaan ini fokus pada memproduksi berbagai jenis permen jahe berkualitas yang dibuat tanpa bahan pengawet dan dari bahan natural pilihan. PT. Sin A Sixfifteen dibangun pada tahun 1997 dengan tujuan untuk mengembangkan lebih lanjut usaha pada pabrik utamanya yaitu PT. Sindu Amritha. Pada awalnya, perusahaan ini fokus pada melakukan perdagangan makanan dan minuman. Namun pada tahun 2002, PT. Sin A Sixfifteen menjadi *co-manufacturer* untuk perusahaan utamanya yaitu PT. Sindu Amritha untuk memproduksi produk permen jahe. Saat ini PT. Sin A Sixfifteen sudah bukan menjadi *co-manufacturer* PT. Sindu Amritha namun tetap memproduksi jenis permen jahe yang sama dimana hasil produksinya hanya untuk di ekspor ke luar negeri [2].

Sebagai salah satu langkah dalam menerapkan Industri 4.0 ke dalam perusahaan, PT. Sin A Sixfifteen ingin memanfaatkan konsep *Internet of Things* (IoT) ke dalam salah satu proses bisnis mereka, yaitu melakukan pengawasan dan pengendalian suhu dan kelembapan pada tiap area krusial yang ada dalam pabrik. PT. Sin A Sixfifteen ingin proses bisnis tersebut terotomasi dengan campur tangan manusia yang minim. Pengendalian dan pengawasan suhu dan kelembapan adalah salah satu cara untuk melakukan jaminan mutu pada produk makanan yang di produksi. Sebagai perusahaan yang memproduksi permen, perubahan suhu dan kelembapan perlu diawasi dengan baik karena berpengaruh pada kualitas permen. Jika pada saat penyimpanan terdapat permen yang terpapar pada suhu yang tinggi maka permen dapat meleleh dan melekat pada bungkus. Selain itu, kelembapan yang tinggi dapat membuat air meresap ke dalam produk permen jadi maupun bahan baku. Permen yang tidak melekat pada bungkusnya dan persentase tepat kadar air dalam produk permen adalah salah satu syarat lulus uji *quality control* pada PT. Sin A Sixfifteen.

Saat ini pengawasan dan pengendalian dilakukan dengan secara manual dimana seseorang akan memeriksa dan mencatat nilai suhu pada tiap ruangan secara berkala. Selain itu pengawasan hanya dilakukan pada variabel suhu dan belum ada variabel kelembapan.

Untuk menerapkan konsep *Internet of Things* (IoT) ke dalam proses bisnis tersebut, dibutuhkan sebuah sistem yang mampu mengukur suhu dan kelembapan di udara dan mampu membantu mengatur alat yang dapat mengubah suhu dan kelembapan secara otomatis sesuai dengan pembacaan suhu dan kelembapan yang didapat. Sebuah mikrokontroler Arduino dapat digunakan sebagai basis alat untuk menjalankan tugas tersebut. Arduino dipilih menjadi basis alat untuk sistem yang akan dibuat karena berlisensi *open source*, memiliki ukuran yang kecil, serta mudah digunakan [3]. Selain Arduino sebagai basis alat, dibutuhkan juga aplikasi yang dapat menyimpan dan mengatur informasi dari Arduino yang didapat. Aplikasi

berbasis web dapat menjalankan tugas ini dengan baik karena aplikasi web dapat diakses kapan saja dan dimana saja.

Arduino sebagai mikrokontroler dihubungkan ke sensor suhu dan kelembapan untuk mendapatkan informasi suhu dan kelembapan di lingkungan sekitar. Tidak hanya sensor suhu dan kelembapan, sebuah sakelar yang terhubung dengan relai juga dihubungkan ke Arduino sebagai alat pengendalian atau tindak lanjut otomatis jika suhu dan atau kelembapan melebihi batas yang telah ditentukan sebelumnya dengan memasang sebuah alat lain yang dapat mengubah suhu dan atau kelembapan ke dalam sakelar tersebut. Agar data suhu dan kelembapan dapat dibaca oleh pengguna serta dapat mengubah batas suhu dan atau kelembapan untuk mengatur sakelar otomatis maka Arduino dihubungkan ke aplikasi melalui pengiriman data melalui internet.

Dengan demikian, pengerjaan tugas akhir ini diharapkan dapat menghasilkan sebuah alat yang dapat mengotomasi proses bisnis pengawasan dan pengendalian suhu dan kelembapan pada PT. Sin A Sixfifteen.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, maka rumusan masalah pada tugas akhir ini adalah sebagai berikut:

1. Bagaimana mengukur suhu dan kelembapan menggunakan sensor yang kompatibel dengan Arduino?
2. Bagaimana mendapatkan hasil pembacaan dari sensor ke dalam Arduino?
3. Bagaimana membuat sakelar otomatis yang berdasar pada nilai yang diberikan pada Arduino?
4. Bagaimana membuat antarmuka yang dapat mengontrol sakelar otomatis serta mendapatkan informasi dari sensor yang ada pada Arduino?

5. Bagaimana menghubungkan antara antarmuka dengan Arduino menggunakan protokol HTTP?

1.3 Batasan Masalah

Berdasarkan permasalahan yang telah diuraikan sebelumnya, adapun batasan masalah dalam tugas akhir ini adalah sebagai berikut :

1. Suhu dan kelembapan yang diukur adalah suhu dan kelembapan ruangan.
2. Sakelar otomatis digunakan untuk memutus atau menyambung aliran listrik stopkontak.
3. Stopkontak yang diatur oleh relai menggunakan arus listrik AC 220V.
4. Antarmuka untuk menampilkan data sensor dan pengaturan sakelar otomatis dibuat dalam bentuk *website*.
5. Antarmuka yang dikembangkan menggunakan bahasa pemrograman PHP.

1.4 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah disebutkan sebelumnya, adapun tujuan dari penelitian tugas akhir ini adalah mengembangkan sebuah prototipe sistem yang mampu mengumpulkan, mengolah, dan menyimpan informasi suhu dan kelembapan pada suatu ruangan serta memberikan umpan balik hasil pembacaan suhu dan kelembapan kepada sakelar otomatis sesuai dengan kondisi yang telah diatur.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dapat diperoleh dari penelitian tugas akhir ini adalah sebagai berikut:

1. Menjadi panduan untuk PT. Sin A Sixfifteen untuk membangun sistem sakelar otomatis berbasis Arduino berdasarkan suhu dan kelembapan yang dapat membantu proses bisnis pengawasan dan pengendalian suhu dan kelembapan ruangan.

2. Membantu individu atau kelompok untuk mengembangkan sebuah sistem sakelar otomatis berbasis Arduino berdasarkan suhu dan kelembapan.
3. Menjadi referensi untuk melakukan pengembangan pada bidang *Internet of Things*.

1.6 Relevansi

Tugas akhir ini disusun untuk memenuhi salah satu syarat kelulusan sebagai Sarjana Komputer di Departemen Sistem Informasi Fakultas Teknologi Informasi dan Komunikasi (FTIK) Institut Teknologi Sepuluh Nopember (ITS). Tugas akhir ini sesuai dengan penerapan mata kuliah dari laboratorium Infrastruktur dan Keamanan Teknologi Informasi (IKTI) Departemen Sistem Informasi FTIK ITS, yaitu Internet Untuk Segala. Selain itu, juga berkaitan dengan mata kuliah wajib, yaitu Pemrograman Berbasis Web dan Rancang Bangun Perangkat Lunak.

Halaman ini sengaja dikosongkan

BAB II TINJAUAN PUSTAKA

Pada bagian ini akan dijelaskan tentang studi literatur penelitian sebelumnya yang berhubungan dengan tugas akhir yang akan dikerjakan dan dasar teori yang akan menjadi landasan dalam pengerjaan tugas akhir ini.

2.1 Studi Literatur

Dalam proses pengerjaan penelitian tugas akhir ini, beberapa penelitian dijadikan sebagai referensi dalam pengerjaan tugas akhir ini. Hasil pencarian penelitian disajikan pada Tabel 2.1.

Tabel 2. 1 Studi Literatur

1) Perancangan Prototipe Sistem Kontrol Suhu dan Kelembapan pada Gedung Walet dengan Mikrokontroller Berbasis Mobile [4]
Penulis; Tahun Suti Kurnia Dewi, Rudy Dwi Nyoto, Elang Derdian Marindani; 2018
Pembahasan Sistem kontrol suhu dan kelembapan untuk gedung walet dibuat dengan menghubungkan mikrokontroller Wemos D1 Mini dengan sensor suhu dan kelembapan DHT11, sensor pendeteksi kualitas udara MQ135, serta relai. Penggunaan relai bertujuan untuk mengontrol alat seperti kipas dan pompa air yang bertujuan untuk mengontrol suhu dan kelembapan dengan memutus atau menyambungkan aliran listrik ke alat-alat yang terhubung ke relai. Data yang didapat dari sensor seperti suhu, kelembapan, serta nilai kualitas udara dikirim oleh mikrokontroller dengan cara terhubung ke Wi-Fi terlebih dahulu yang kemudian berkomunikasi dengan aplikasi web untuk menyimpan data-data tersebut. Dari data yang didapat kemudian diolah untuk menentukan apakah perlu adanya penyesuaian suhu dan kelembapan.

<p>Keterkaitan Penelitian tugas akhir yang dilakukan berkaitan dengan penggunaan sensor suhu dan kelembapan untuk mengukur suhu dan kelembapan di suatu ruangan serta penggunaan relai yang terhubung dengan mikrokontroler untuk memutus dan menyambung aliran listrik.</p>
<p>2) A keyless Entry System based on Arduino board with Wi-Fi technology [5]</p>
<p>Penulis; Tahun Marwa F. Areed; 2019</p>
<p>Pembahasan Mengatur suatu alat secara jarak jauh atau <i>remote</i> dapat dilakukan dengan menggunakan <i>board</i> Arduino yang terhubung ke internet dengan menggunakan teknologi Wi-Fi. Teknologi Wi-Fi adalah teknologi yang paling sesuai untuk mengatur alat secara jarak jauh karena Wi-Fi memiliki kapabilitas untuk terhubung ke internet sehingga tidak memiliki batasan jarak dalam mengatur alat secara jarak jauh. Sebuah <i>board</i> Arduino dapat ditambah dengan Wi-Fi <i>shield</i> yang berfungsi untuk menangkap sinyal Wi-Fi dari lingkungan sekitar. Beberapa baris kode tambahan dalam memprogram <i>board</i> Arduino diperlukan sehingga Wi-Fi <i>shield</i> yang telah dipasang ke <i>board</i> Arduino dapat berjalan. Sebuah aplikasi diperlukan untuk mengatur alat yang ingin diatur. Sebuah <i>website</i> dibuat untuk mengatur alat melalui halaman webnya serta untuk berkomunikasi dengan Arduino.</p>
<p>Keterkaitan Penelitian tugas akhir yang dilakukan berkaitan dengan penggunaan teknologi Wi-Fi yang berfungsi untuk mengatur sebuah alat secara jarak jauh dimana sebuah aplikasi khusus dapat berkomunikasi dua arah dengan alat yang ingin diatur.</p>

3) Rancang Bangun Sistem Keamanan Rumah Menggunakan Relay [6]

Penulis; Tahun

Muhammad Saleh, Munnik Haryanti; 2017

Pembahasan

Sebuah relai adalah sebuah komponen elektrik yang mampu memutus dan menyambung aliran listrik. Relai menggunakan elektromagnetik untuk menggerakkan saklar sehingga arus listrik yang bertegangan tinggi dapat diputus atau disambungkan hanya dengan menggunakan listrik bertegangan kecil. Sistem keamanan rumah menggunakan relai yang mampu menyalakan alarm dan seluruh lampu yang ada jika ditemukan indikasi memasuki rumah secara paksa.

Keterkaitan

Paper ini memiliki keterkaitan dengan penggunaan komponen elektronik relai yang berfungsi untuk memutus atau menyambung arus listrik pada skema elektronik yang akan dibuat pada tugas akhir.

2.2 Dasar Teori

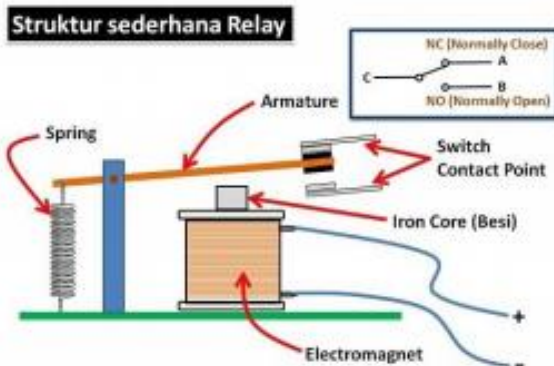
2.2.1 Arduino

Arduino adalah sebuah platform *hardware* dan *software open-source* yang mudah digunakan untuk melakukan pemrograman alat-alat elektronik. Hardware dari Arduino berupa sebuah *board* yang memiliki mikrokontroler yang mampu membaca *input* atau memberikan *output* berupa sinyal digital atau analog ke alat lain yang dihubungkan ke Arduino. Sinyal digital hanya dapat menerima atau memberikan dua nilai sedangkan sinyal analog dapat menerima atau memberikan nilai voltase. Untuk memprogram sebuah *board* Arduino dapat menggunakan Arduino IDE yang merupakan sebuah perangkat lunak yang berfungsi untuk menulis program apa yang ingin dijalankan oleh *board* dan mengunggahnya ke dalam *board* tersebut. Bahasa pemrograman yang digunakan dalam menulis program

Arduino adalah bahasa C. Karena lisensi *software* dan *hardware* Arduino merupakan *open source*, maka desain dari *hardware* Arduino dapat dibuat dan diubah menyesuaikan kebutuhan proyek yang ingin dijalankan. Variasi Arduino yang paling umum digunakan untuk sebuah proyek adalah Arduino UNO karena memiliki *hardware* yang paling sederhana dan paling mudah digunakan [3].

2.2.2 Relai

Relai adalah sebuah komponen elektromekanikal yang berfungsi untuk memutuskan atau menghubungkan arus listrik. Relai bekerja dengan cara menggerakkan saklar menggunakan arus listrik dengan prinsip elektromagnetik. Komponen relai terdiri dari empat sub-komponen, yaitu saklar mekanik, koil elektromagnetik, *armature*, dan pegas.



Gambar 2. 1 Struktur Sederhana Relai

Dalam menggunakan relai, terdapat dua kontak poin yang penting untuk diperhatikan. Kontak poin *normally close* (NC) digunakan jika ingin kondisi awal sebelum koil diaktifkan adalah arus listrik terhubung. Sedangkan kontak poin *normally open* (NO) digunakan jika ingin kondisi awal sebelum koil diaktifkan adalah arus listrik terputus. Pada umumnya, relai memiliki empat pin, yaitu pin *normally close*, pin *normally open*, pin koil positif, dan pin koil negatif [6].

2.2.3 Suhu dan Kelembapan Ruangan

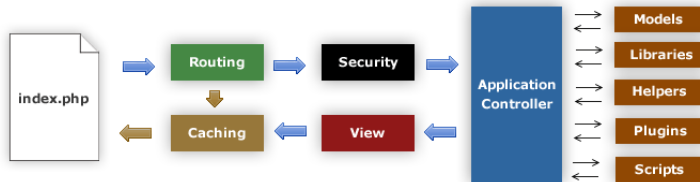
Suhu ruangan dan kelembapan ruangan saling berpengaruh satu sama lain. Perubahan yang terjadi pada suhu ruangan dapat mempengaruhi tingkat kelembapan ruangan. Pada dasarnya, suhu ruangan memiliki batas natural tentang banyaknya kelembapan yang dapat dimiliki oleh udara sebelum udara menjadi jenuh. Suhu udara yang dingin memiliki tingkat kapasitas kelembapan yang lebih rendah daripada suhu udara yang panas. Tingkat kelembapan di udara dapat diukur dapat diukur secara absolut maupun relatif. Kelembapan absolut diukur dengan mengukur banyaknya konten air yang ada dalam suatu ukuran, sedangkan kelembapan relatif diukur dengan persentase dari kapasitas maksimum konten air yang ada dalam udara. Kapasitas maksimum konten air dapat berubah seiring dengan perubahan temperatur, sehingga pengukuran kelembapan relatif hanya spesifik pada tiap temperatur [7].

2.2.4 MySQL

MySQL adalah sebuah sistem manajemen basis data yang dikembangkan oleh perusahaan Oracle. Maksud dari sistem manajemen basis data adalah sebuah perangkat lunak yang dapat mengatur sekumpulan data yang terstruktur sesuai dengan keinginan pengguna. Contoh dari penggunaan sistem manajemen basis data sangat beragam, mulai dari menyimpan daftar belanja sederhana hingga menyimpan informasi penting perusahaan. MySQL menggunakan penyimpanan data dengan model relasional dimana data dapat disimpan secara terpisah daripada meletakkan seluruh data dalam satu tabel besar. Dengan menggunakan model relasional, kecepatan akses data dapat menjadi lebih optimal serta penyimpanan data menjadi fleksibel karena dapat mengatur jenis hubungan antar data seperti *one-to-one* dan *one-to-many*. Untuk mengakses basis data, MySQL menggunakan bahasa SQL yang merupakan singkatan dari *structured query language*. Bahasa SQL adalah Bahasa terstandar yang digunakan untuk mengakses basis data. MySQL adalah perangkat lunak *open source* yang berarti siapapun dapat menggunakan dan memodifikasi perangkat lunak tersebut [8].

2.2.5 CodeIgniter

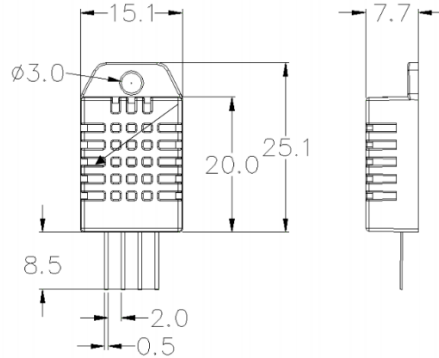
CodeIgniter adalah sebuah *framework* untuk membuat sebuah aplikasi berbasis web menggunakan bahasa pemrograman PHP. Tujuan dari adanya *framework* ini adalah untuk membangun sebuah aplikasi lebih cepat daripada membangun aplikasi dari nol. CodeIgniter memiliki sekumpulan *libraries* yang berisi sekumpulan program yang sering digunakan yang dapat digunakan untuk mempercepat proses pengembangan. Pengembangan aplikasi menggunakan CodeIgniter menggunakan pola desain MVC atau Model-View-Controller dimana aplikasi yang dibuat memisahkan antara logika aplikasi dari tampilan visual yang dibuat. Model dalam MVC merepresentasikan struktur data. View dalam MVC adalah informasi yang di tampilkan kepada *user*. *Controller* dalam MVC berfungsi sebagai jembatan antara *Model* dan *View* dimana *Controller* berisi logika aplikasi [9].



Gambar 2. 2 Flowchart Cara Kerja CodeIgniter

2.2.6 DHT22

DHT22 atau yang juga bisa disebut AM2302 adalah sebuah sensor yang mampu mengukur suhu dan kelembapan di lingkungan sekitarnya. Sensor ini mengeluarkan sinyal digital dalam mengirimkan informasi hasil pembacaannya. Dalam penggunaannya, DHT22 menggunakan sumber listrik sebesar 3.3-6V DC dengan konsumsi daya yang rendah. DHT22 mampu mengukur kelembapan mulai dari 0-100% RH (*Relative Humidity*) dan untuk suhu mampu mengukur mulai dari -40°C hingga 80°C. Keakuratan untuk nilai kelembapan adalah $\pm 2\%$ RH hingga $\pm 5\%$ RH dan untuk suhu adalah $\pm 0.5^\circ\text{C}$. DHT22 membutuhkan waktu 5ms untuk mengirimkan data [10].



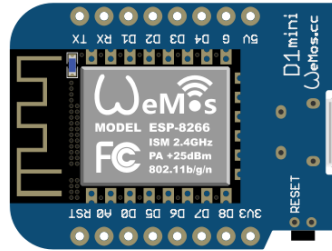
Gambar 2. 3 Dimensi DHT22 dalam Milimeter

DHT22 memiliki empat pin dengan fungsinya masing-masing. Berikut adalah fungsi tiap pin dimulai dari pin paling kiri:

- Pin 1: VDD (Sumber daya)
- Pin 2: DATA (Sinyal)
- Pin 3: NULL (Tidak digunakan)
- Pin 4: GND (*Ground*)

2.2.7 Wemos D1 Mini

Wemos D1 Mini adalah sebuah *board* dengan mikrokontroler ESP8266EX. *Board* ini adalah alternatif dari *board* Arduino UNO karena memiliki harga yang lebih murah, dimensi yang lebih kecil, serta memiliki kemampuan untuk berkomunikasi dengan Wi-Fi. ESP8266EX adalah mikrokontroler 32-bit yang ditenagai oleh 3.3V dan berjalan pada kecepatan 80Mhz. Wi-Fi dari mikrokontroler ini menggunakan standar 802.11b/g/n dengan menggunakan ISM band 2.4GHz. Dimensi dari *board* ini berukuran 34.2 mm x 25.6 mm dan memiliki berat hanya sebesar 10 gram. Pin yang ada dalam Wemos D1 Mini antara lain adalah sebelas digital I/O pin, satu analog pin, satu 5V pin, satu 3,3V pin, satu *ground* pin, serta satu reset pin. Wemos D1 Mini dapat diprogram oleh Arduino IDE seperti *board* Arduino lainnya [11, 12].



Gambar 2. 4 Board Wemos D1 Mini

2.2.8 Kondisi Lingkungan PT. Sin A Sixfifteen

Berdasarkan hasil wawancara pada perusahaan PT. Sin A Sixfifteen, terdapat delapan area yang perlu dilakukan pengawasan dan pengendalian terhadap suhu dan kelembapan. Area tersebut antara lain adalah produksi *soft candy*, produksi *hard candy*, *2nd packing*, gudang *finished goods*, gudang bahan, ruang *quality control/lab*, *cooling storage*, dan ruang *chemical*. Pada area produksi *soft candy* dan produksi *hard candy*, area tersebut terdapat pada sebuah ruangan yang dipisah oleh sekat berdasarkan fungsi masing-masing *sub-area*. Tabel 2.2 menjelaskan spesifikasi tiap area dengan informasi *threshold* suhu dan kelembapan yang perlu dijaga serta ukuran ruangan tiap area.

Tabel 2. 2 Ruang Studi Kasus

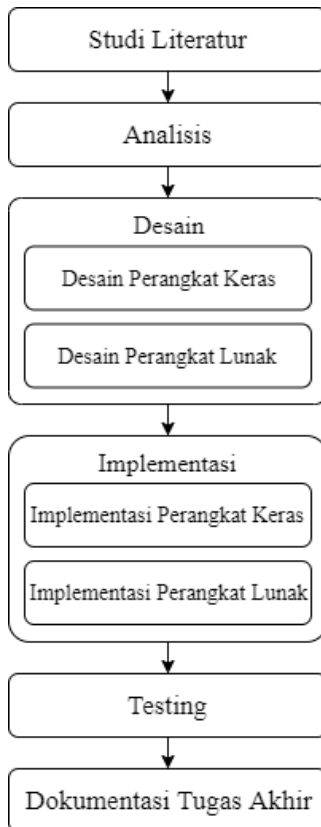
No	Area	Ukuran Ruang	Suhu Min.	Suhu Maks.	RH Min.	RH Maks.
1	Produksi <i>Soft Candy</i>					
	<i>Cooking</i>	135m ²	30°C	40°C	35%	55%
	<i>Cooling</i>	135m ²	30°C	35°C	35%	55%
	<i>Cutting</i>	140m ²	20°C	30°C	35%	50%
	<i>Pillow Packing</i>	144m ²	20°C	30°C	35%	50%
2	Produksi <i>Hard Candy</i>					
	<i>Cooking</i>	45m ²	30°C	40°C	30%	50%
	<i>Cooling</i>	30m ²	20°C	30°C	30%	45%
	<i>Pillow Packing</i>	60m ²	20°C	30°C	30%	45%

3	<i>2nd Packing</i>	300m ²	25°C	30°C	40%	55%
4	<i>Gudang Finished Goods</i>	850m ²	28°C	32°C	45%	55%
5	<i>Gudang Bahan</i>	190m ²	30°C	35°C	50%	70%
6	<i>Ruang QC/Lab</i>	36m ²	25°C	35°C	40%	55%
7	<i>Cooling Storage</i>	15m ²	14°C	16°C	-	-
8	<i>Ruang Chemical</i>	16m ²	25°C	35°C	40%	55%

Halaman ini sengaja dikosongkan

BAB III METODOLOGI

Pada bagian metodologi akan menjelaskan bagaimana langkah pengerjaan tugas akhir dengan disertakan deskripsi dari setiap penjelasan untuk masing-masing tahapan beserta jadwal kegiatan pengerjaan tugas akhir. Gambar 3.1 menjelaskan tahapan pengerjaan tugas akhir.



Gambar 3. 1 Tahapan Pengerjaan Tugas Akhir

3.1 Studi Literatur

Tahap studi literatur adalah tahap pertama dalam mengerjakan tugas akhir. Pada tahap ini dilakukan pengumpulan dan mengkaji literatur. Literatur yang dikumpulkan adalah literatur yang memiliki keterkaitan dengan judul tugas akhir yang dikerjakan. Hasil dari studi literatur adalah konsep, acuan, teknologi, serta metode yang telah dipahami yang dapat digunakan untuk menyelesaikan tugas akhir.

3.2 Analisis

Seluruh pengetahuan yang telah didapat dari hasil studi literatur akan digunakan sebagai panduan dalam tahap analisis. Pada tahap analisis dilakukan penggalian kebutuhan sistem yang akan dibangun yang kemudian disesuaikan dengan pengetahuan yang telah didapat dari studi literatur. Penggalian kebutuhan sistem dilakukan dengan cara melakukan wawancara pada narasumber yang memegang peran penting pada penyelesaian sistem yang akan dibuat untuk perusahaan studi kasus. Hasil dari tahap ini adalah spesifikasi kebutuhan pembuatan perangkat lunak dan perangkat keras dari sistem yang akan dibuat.

3.3 Desain

Tahap desain adalah tahap dimana melakukan desain sistem yang akan dibuat berdasarkan spesifikasi sistem yang dibutuhkan yang sudah dibuat pada tahap analisis. Tahap desain akan dibagi menjadi dua tahap, yaitu tahap desain perangkat keras dan desain perangkat lunak.

3.3.1 Desain Perangkat Keras

Pada tahap ini dilakukan proses desain pada perangkat keras yaitu mikrokontroler Arduino beserta dengan sensor dan relai. Tahap ini akan menjelaskan bagaimana desain rangkaian elektronik Arduino yang akan dibuat, desain program yang akan dijalankan pada Arduino, serta alur proses bagaimana Arduino bekerja.

3.3.2 Desain Perangkat Lunak

Pada tahap ini dilakukan proses desain pada perangkat lunak. Tahap ini akan menjelaskan bagaimana alur proses dari aplikasi web, desain struktur aplikasi web, desain penerimaan, penyimpanan, dan pengiriman data Arduino, desain tampilan visualisasi data yang didapat, dan desain pengaturan Arduino.

3.4 Implementasi

Pada tahap implementasi dilakukan proses pembuatan dan perakitan sistem berdasarkan desain yang telah dibuat pada tahap sebelumnya. Tahap implementasi dibagi menjadi dua tahap, yaitu tahap implementasi perangkat keras dan tahap implementasi perangkat lunak.

3.4.1 Implementasi Perangkat Keras

Pada tahap ini dilakukan proses perakitan perangkat keras berdasarkan komponen-komponen yang telah ditentukan sebelumnya pada tahap desain. Sensor, relai, dan mikrokontroler Arduino dirangkai menjadi satu. Arduino yang telah terhubung dengan sensor dan relai diprogram sehingga mampu menerima data hasil pembacaan sensor, mampu mengatur relai, mampu terkoneksi dengan Wi-Fi, serta mampu melakukan penerimaan dan pengiriman data dengan aplikasi web.

3.4.2 Implementasi Perangkat Lunak

Pada tahap ini dilakukan pembuatan aplikasi web yang berfungsi untuk memonitor data yang diterima oleh sensor serta mengatur relai yang ada pada Arduino. Pembuatan aplikasi web akan mengacu pada desain perangkat lunak yang telah dibuat pada tahap sebelumnya. Aplikasi web yang dibuat akan menggunakan *framework* PHP bernama CodeIgniter dan akan berkomunikasi dengan Arduino.

3.5 Testing

Pada tahap testing akan dilakukan pengujian terhadap keseluruhan sistem yang telah dibuat. Pengujian terhadap sistem dilakukan dengan cara melakukan *test case* dimana setiap *case*

memiliki skenario yang harus dilakukan. Jika *test case* yang dilakukan terdapat skenario yang tidak sesuai dengan ekspektasi, maka proses pengerjaan kembali ke tahap implementasi dimana di tahap tersebut dilakukan perbaikan *bug* yang ditemukan pada saat *test case* dilakukan. Tahap ini akan selesai jika seluruh *test case* telah berhasil dijalankan tanpa kendala. Luaran dari tahap ini adalah dokumentasi seluruh *test case* yang telah dijalankan.

3.6 Dokumentasi Tugas Akhir

Dokumentasi tugas akhir adalah tahap akhir dari metodologi pengerjaan tugas akhir dimana seluruh luaran dari tiap tahap metodologi yang telah dilakukan sebelumnya di dokumentasi serta kesimpulan dari tugas akhir yang dikerjakan dan saran untuk penelitian selanjutnya.

BAB IV PERANCANGAN

Pada bagian ini akan menjelaskan tentang proses perancangan sistem yang akan dijadikan panduan dalam proses implementasi. Perancangan sistem meliputi perancangan perangkat keras dan perancangan perangkat lunak.

4.1 Analisis Kebutuhan

Proses desain sistem dimulai dengan mengumpulkan informasi mengenai kebutuhan apa saja yang diperlukan pada sistem yang akan diimplementasikan serta mengetahui kondisi lapangan dimana sistem akan dipasang. Pengumpulan informasi dilakukan dengan mewawancarai narasumber pada perusahaan studi kasus. Narasumber wawancara adalah Daniel Dwi Rahma yang memiliki peran sebagai Kepala Departemen Teknologi Informasi pada perusahaan PT. Sin A Sixfifteen.

Sistem dibuat dengan tujuan untuk melakukan pengawasan pada suhu dan kelembapan ruangan serta mengatur relai berdasarkan suhu dan kelembapan tersebut. Relai tersebut akan digunakan untuk mengatur stopkontak 220V yang nantinya akan terhubung pada alat yang mampu menjaga suhu dan kelembapan tetap berada dalam *threshold* yang telah ditentukan. Penyimpanan informasi serta visualisasi informasi suhu dan kelembapan juga menjadi poin penting dalam sistem ini sehingga pengguna dapat melakukan pengawasan lebih mudah. Untuk dapat menentukan alat mana saja yang perlu dihidupkan atau dimatikan melalui relai, diperlukan informasi suhu dan kelembapan yang didapatkan dengan sensor suhu dan kelembapan. Tabel 4.1 menjelaskan pembuatan keputusan yang akan dilakukan jika suhu atau kelembapan pada suatu ruangan melebihi *threshold* yang telah ditetapkan.

Tabel 4. 1 Pembuatan Keputusan

Informasi	Keputusan
Suhu melebihi batas atas <i>threshold</i>	Menghidupkan stopkontak yang terhubung dengan alat

	menurunkan suhu menggunakan relai
Suhu melebihi batas bawah <i>threshold</i>	Menghidupkan stopkontak yang terhubung dengan alat menaikkan suhu menggunakan relai
Kelembapan melebihi batas atas <i>threshold</i>	Menghidupkan stopkontak yang terhubung dengan alat menurunkan kelembapan menggunakan relai
Kelembapan melebihi batas bawah <i>threshold</i>	Menghidupkan stopkontak yang terhubung dengan alat menaikkan kelembapan menggunakan relai

Secara garis besar, sistem terbagi menjadi dua bagian yaitu perangkat Arduino dan aplikasi web. Perangkat Arduino berfungsi untuk mengetahui informasi suhu dan kelembapan serta untuk mengatur relai. Sedangkan aplikasi web berfungsi untuk mencatat dan menampilkan informasi suhu dan kelembapan kepada pengguna serta untuk melakukan konfigurasi kondisi relai. Jumlah perangkat Arduino yang terhubung dengan aplikasi web dapat bertambah sesuai dengan jumlah ruangan yang perlu dilakukan pengawasan sehingga diperlukan aplikasi web yang mampu menerima koneksi dari banyak perangkat. Tabel 4.2 adalah kebutuhan fungsional dari sistem yang didapatkan dari hasil wawancara dengan narasumber. Tabel 4.3 adalah kebutuhan non fungsional dari sistem yang didapatkan dari hasil diskusi lanjut bersama narasumber serta saran-saran yang disetujui oleh narasumber.

Tabel 4. 2 Kebutuhan Fungsional

No	Kategori	Kebutuhan
1	Perangkat Arduino	Perangkat dapat mengetahui informasi suhu dan kelembapan dengan menggunakan sensor.

2	Perangkat Arduino	Perangkat dapat mengirimkan informasi suhu dan kelembapan secara berkala ke aplikasi web dengan jarak tiap waktu pengiriman yang dapat ditentukan.
3	Perangkat Arduino	Perangkat dapat mengatur relai sesuai dengan <i>threshold</i> suhu dan atau kelembapan yang telah ditentukan pada aplikasi web.
4	Perangkat Arduino	Perangkat dapat menerima informasi konfigurasi perangkat yang diatur pada aplikasi web.
5	Aplikasi web	Aplikasi dapat menerima informasi suhu data dan kelembapan dari tiap perangkat Arduino yang telah terdaftar.
6	Aplikasi web	Aplikasi dapat menyimpan informasi suhu dan kelembapan yang didapat dari tiap perangkat.
7	Aplikasi web	Aplikasi dapat mengolah informasi suhu dan kelembapan dari tiap perangkat untuk didapatkan pengetahuan baru dari data tersebut.
8	Aplikasi web	Aplikasi dapat melakukan visualisasi informasi suhu dan kelembapan dari tiap perangkat.
9	Aplikasi web	Aplikasi dapat mengatur pengaturan tiap relai pada tiap perangkat Arduino berdasarkan <i>threshold</i> suhu dan atau kelembapan.
10	Aplikasi web	Aplikasi dapat mengirimkan konfigurasi perangkat kepada tiap perangkat yang terhubung.
11	Aplikasi web	Aplikasi dapat menambahkan dan menghapus perangkat Arduino yang dapat terhubung ke aplikasi.
12	Aplikasi web	Aplikasi memiliki sistem manajemen pengguna untuk mengatur hak akses pengguna.

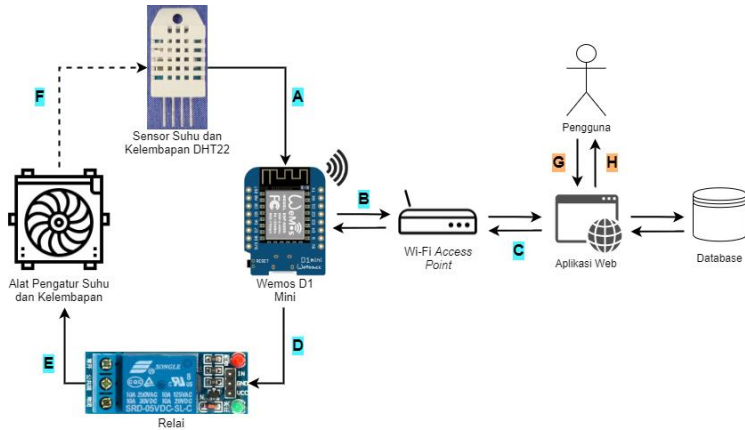
13	Aplikasi web	Aplikasi dapat melakukan kalibrasi akurasi informasi suhu dan kelembapan di tiap perangkat Arduino
14	Aplikasi web	Pengguna aplikasi dapat mengunduh informasi suhu dan kelembapan tiap perangkat

Tabel 4.3 Kebutuhan Non Fungsional

No	Kategori	Kebutuhan
1	Perangkat Arduino	Perangkat dapat terhubung secara otomatis ke jaringan Wi-Fi yang telah ditentukan
2	Perangkat Arduino	Perangkat dapat berjalan dengan stabil
3	Perangkat Arduino	Perangkat melakukan komunikasi dengan aplikasi web menggunakan protokol HTTPS
4	Aplikasi web	Aplikasi menerapkan sistem <i>key</i> sebagai otorisasi untuk menerima dan mengirim informasi ke perangkat Arduino
5	Aplikasi web	Aplikasi dapat berjalan pada <i>web browser</i>
6	Aplikasi web	Aplikasi dapat berjalan dengan stabil
7	Aplikasi Web	Aplikasi web menggunakan protokol HTTPS

4.2 Desain Sistem

Desain sistem yang dibuat akan menjadi panduan dalam melakukan pengembangan sistem pada tahap implementasi. Desain sistem akan menjelaskan bagaimana sistem bekerja baik dari sisi perangkat lunak maupun perangkat keras.

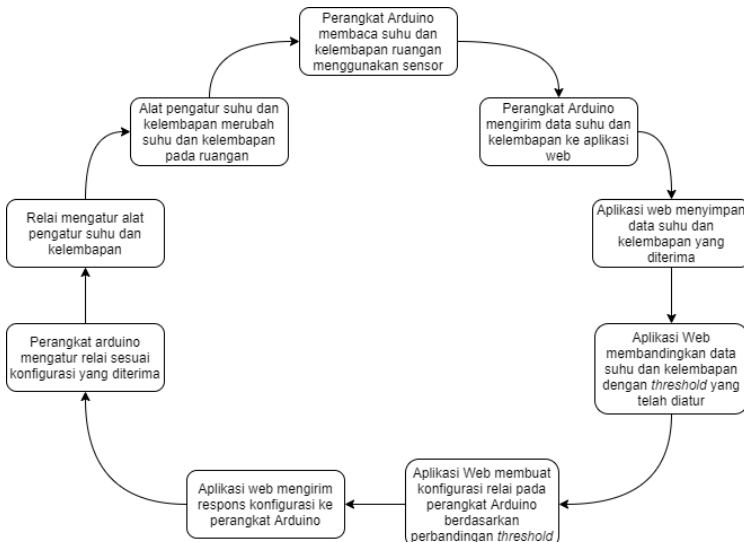


- A. Sensor DHT22 mengirim data suhu dan kelembapan ke Wemos D1 Mini.
- B. Wemos D1 Mini mengirim data suhu dan kelembapan ke aplikasi web untuk disimpan pada database.
- C. Aplikasi web memberi respons konfigurasi ke perangkat Arduino.
- D. Arduino mengatur relai berdasarkan konfigurasi yang didapat dari aplikasi web.
- E. Relai menghidupkan atau mematikan alat pengatur suhu dan kelembapan.
- F. Alat pengatur suhu dan kelembapan mempengaruhi sensor DHT22.
- G. Pengguna mengkonfigurasi threshold dan relai perangkat Arduino.
- H. Pengguna mengawasi suhu dan kelembapan perangkat Arduino.

Gambar 4. 1 Desain Sistem

Gambar 4.1 menjelaskan bagaimana desain sistem secara keseluruhan mulai dari pembacaan data suhu dan kelembapan hingga mengontrol relai yang terhubung dengan alat pengatur suhu dan kelembapan. Pertama, Arduino membaca data suhu dan kelembapan dengan menggunakan sensor. Selanjutnya Arduino mengirimkan data tersebut ke aplikasi web melalui *access point* Wi-Fi yang terhubung dengan internet. Data suhu dan kelembapan yang diterima oleh aplikasi web akan disimpan pada *database* dan ditampilkan pada aplikasi web. Setelah data suhu dan kelembapan disimpan, aplikasi web akan memberikan respons balik kepada Arduino berupa data relai mana yang akan dihidupkan atau dimatikan sesuai dengan perbandingan antara *threshold* dengan informasi suhu dan kelembapan yang didapat. *Threshold* suhu dan kelembapan dan relai dari setiap perangkat Arduino diatur oleh pengguna melalui aplikasi web yang

memiliki alur sendiri terpisah dari alur utama. Setiap relai akan terhubung dengan satu stopkontak dimana setiap stopkontak dihubungkan ke alat yang dapat mengendalikan suhu dan kelembapan. Alat pengatur suhu dan kelembapan yang diatur dapat mempengaruhi sensor suhu dan kelembapan yang menyebabkan terciptanya *feedback loop*. Alur proses mulai dari Arduino menerima data sensor hingga menghidupkan atau mematikan alat memiliki interval yang dapat diatur pada aplikasi web. Untuk lebih jelasnya bagaimana alur antara perangkat Arduino dengan aplikasi web bekerja dapat melihat diagram pada Gambar 4.2



Gambar 4. 2 Alur Perangkat Arduino dengan Aplikasi Web

4.3 Desain Perangkat Keras

Desain perangkat keras dibuat dengan tujuan untuk menjadi acuan dalam membuat perangkat Arduino yang sesuai dengan kebutuhan yang diinginkan. Desain yang dibuat akan mampu mengetahui info suhu dan kelembapan ruangan serta mampu mengatur relai.

4.3.1 Skema Perangkat Keras

Desain dari perangkat Arduino yang digunakan menggunakan tiga komponen penting, yaitu Wemos D1 Mini, DHT22, dan Relai. Wemos D1 Mini digunakan sebagai komponen utama yang mampu mengatur seluruh komponen-komponen lain karena memiliki mikrokontroler yang dapat diprogram oleh pengguna. DHT22 adalah komponen yang mampu memberikan informasi suhu dan kelembapan dengan suhu menggunakan skala Celsius dan kelembapan menggunakan skala *relative humidity*. Sedangkan relai adalah komponen yang mampu memutus atau menghubungkan aliran listrik yang akan digunakan untuk mengatur alat pengatur suhu dan kelembapan.

Untuk dapat mengetahui suhu dan kelembapan suatu ruangan, Wemos D1 Mini dihubungkan dengan sensor DHT22. secara fisik, DHT22 memiliki empat pin namun pin yang beroperasi hanyalah tiga. Ketiga pin tersebut perlu dihubungkan kepada Wemos D1 Mini agar mampu mendapatkan informasi suhu dan kelembapan. Pin pada DHT22 yang perlu diperhatikan adalah pin Data. Pin tersebut akan mengirim sinyal secara digital ke Wemos D1 Mini tentang pembacaan suhu dan kelembapan. Pada desain yang dibuat, pin tersebut akan dihubungkan pada pin D7 yang ada pada Wemos D1 Mini. Tabel 4.4 menunjukkan hubungan antara pin DHT22 dengan pin Wemos D1 Mini.

Tabel 4. 4 Pin DHT22 pada Wemos D1 Mini

Urutan Pin	Pin DHT22	Pin Wemos D1 Mini
1	5V	5V
2	Data	D7
3	Null	Tidak dihubungkan
4	Ground	Ground

Relai yang digunakan untuk mengatur alat memiliki lima pin. Pin tersebut antara lain adalah *coil+*, *coil-*, COM, *Normally Close* (NC), dan *Normally Open* (NO). Pin *coil* bekerja seperti sebuah tuas untuk menentukan apakah arus listrik akan terhubung atau tidak, pin COM berfungsi sebagai sumber listrik

yang ingin dihubungkan, pin *Normally Close* adalah pin yang akan dialiri listrik ketika tidak ada sinyal listrik menuju *coil*, dan pin *Normally Open* adalah pin yang akan dialiri listrik ketika terdapat sinyal listrik menuju *coil*. Wemos D1 Mini mengatur relai dengan mengirimkan sinyal ke *coil+* dengan menggunakan pin digital yang tersedia. Dari lima pin tersebut, hanya empat pin yang akan digunakan dalam sistem ini. Satu pin yang tidak terpakai adalah *Normally Close (NC)*. Pin *Normally Close* tidak digunakan karena pada sistem ini alat akan menyala hanya pada saat Wemos D1 Mini memberikan sinyal kepada relai melalui *coil*. Tabel 4.5 menjelaskan pin dari relai secara singkat.

Tabel 4. 5 Penjelasan Pin Relai

Pin Relai	Deskripsi pin relai	Terhubung Dengan
<i>Coil+</i>	Pin untuk menentukan apakah arus listrik terhubung atau tidak.	Pin digital Wemos D1 Mini
<i>Coil-</i>	Ground untuk <i>Coil+</i> .	<i>Ground</i>
COM	Arus listrik yang ingin dihubungkan.	Sumber listrik 220V
<i>Normally Close (NC)</i>	Sumber arus listrik terhubung ketika tidak ada sinyal masuk menuju <i>coil</i> .	Tidak dihubungkan
<i>Normally Open (NO)</i>	Sumber arus listrik terhubung ketika terdapat sinyal masuk menuju <i>coil</i> .	Alat pengatur suhu atau kelembapan

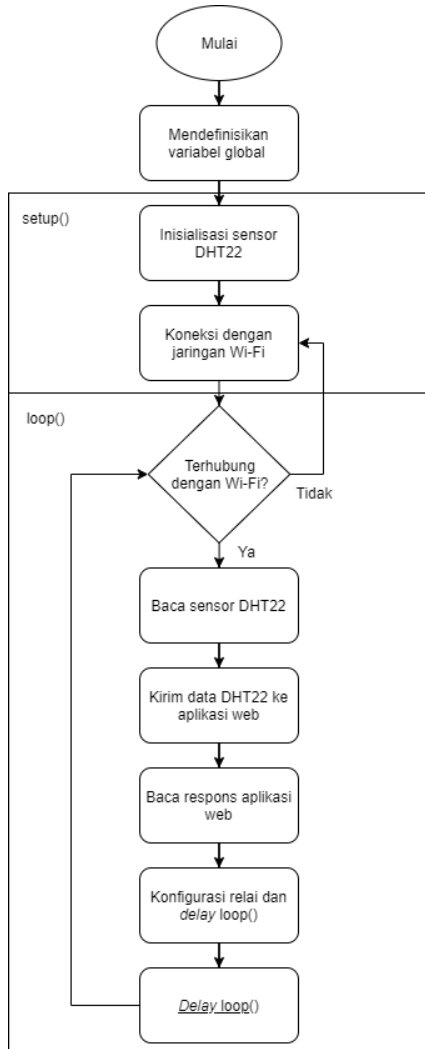
Wemos D1 Mini sebagai komponen utama dari perangkat keras yang dibuat akan ditenagai oleh arus listrik 5V melalui konektor *micro USB* dengan menggunakan *micro USB DC 5V* adapter. Gambar 4.3 adalah desain skema dari perangkat keras yang akan dibuat.

selesai dijalankan, serta konfigurasi untuk sensor DHT22 seperti pin Arduino yang digunakan, variabel suhu dan variabel kelembapan.

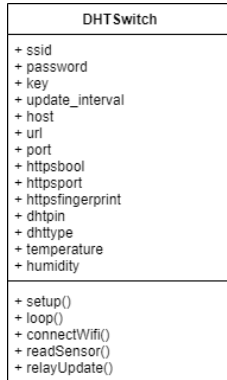
Pada *method* `setup()`, dilakukan inisialisasi pengambilan data dari sensor DHT22. Inisialisasi ini dilakukan dengan tujuan untuk mendapatkan nilai yang akurat. Selain itu pada *method* `setup()` juga terdapat program untuk terhubung dengan jaringan Wi-Fi berdasarkan SSID dan *password* Wi-Fi yang telah ditentukan pada variabel global. Setelah semua program pada *method* `setup()` dijalankan maka *method* `loop()` akan dijalankan secara berulang pada perangkat Arduino.

Pada *method* `loop()`, terdapat urutan program yang dijalankan secara berulang. Pertama adalah memastikan bahwa perangkat Arduino telah benar-benar terhubung dengan Wi-Fi. Jika didapatkan status Wi-Fi yang tidak terhubung, maka perangkat Arduino akan mencoba kembali untuk menghubungkan dengan jaringan Wi-Fi. Setelah memastikan Wi-Fi telah terhubung selanjutnya adalah membaca sensor DHT22 untuk mendapatkan data suhu dan kelembapan ruangan. Data suhu dan kelembapan yang didapat disimpan untuk program selanjutnya yaitu mengirim data ke aplikasi web. Pengiriman data suhu dan kelembapan ke aplikasi web menggunakan protokol HTTP dengan metode POST. Data yang dikirimkan ke aplikasi web adalah *key* untuk autentikasi perangkat Arduino, data suhu, dan data kelembapan yang didapat dari DHT22. Setelah pengiriman data ke aplikasi web maka perangkat Arduino akan mendapat respons dari aplikasi web terkait konfigurasi perangkat Arduino. Terdapat dua konfigurasi penting yang diperbarui setiap mendapat respons dari aplikasi web, yaitu waktu *delay* setiap *method* `loop()` selesai menjalankan semua program dan konfigurasi relai mana saja yang perlu dinyalakan atau dimatikan. Setelah konfigurasi perangkat selesai maka akan dilakukan *delay* sebanyak waktu yang telah dikonfigurasi dan kemudian program yang terdapat pada *method* `loop()` akan diulang. Gambar 4.4 menjelaskan alur program Arduino secara

keseluruhan. Gambar 4.5 adalah *class diagram* hasil dari program Arduino yang telah dijelaskan.



Gambar 4. 4 Desain Alur Program Perangkat Arduino



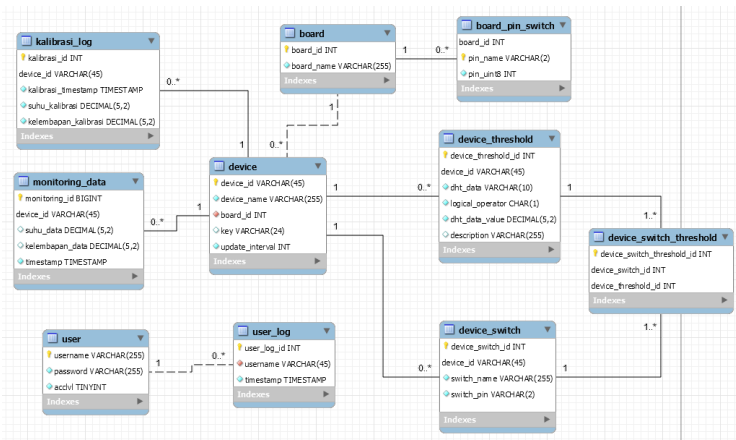
Gambar 4. 5 Class Diagram Program Perangkat Arduino

4.4 Desain Perangkat Lunak

Desain perangkat lunak dibuat sebagai acuan dalam membuat aplikasi web pada tahap implementasi sesuai dengan kebutuhan pada tahap analisis kebutuhan.

4.4.1 Desain Basis Data

Basis data yang digunakan pada aplikasi web menggunakan basis data MySQL. Gambar 4.5 adalah desain dari basis data yang akan digunakan pada aplikasi web.



Gambar 4. 6 Desain Basis Data

Basis data yang digunakan akan terdiri dari delapan tabel. Setiap tabel pada basis data yang digunakan memiliki fungsinya masing-masing pada aplikasi web. Tabel 4.6 menjelaskan nama dari tiap tabel yang digunakan dan fungsi dari tiap tabel.

Tabel 4. 6 Deskripsi Tiap Tabel Basis Data

No	Nama Tabel	Deskripsi
1	device	Menyimpan informasi terkait perangkat Arduino yang dapat terhubung dengan aplikasi web.
2	monitoring_data	Menyimpan informasi suhu dan kelembapan dari semua perangkat Arduino.
3	kalibrasi_log	Menyimpan informasi historis kalibrasi data suhu dan kelembapan dari semua perangkat Arduino.
4	device_switch	Menyimpan informasi terkait relai yang dapat dikonfigurasi dari semua perangkat Arduino.
5	device_threshold	Menyimpan informasi terkait <i>threshold</i> suhu dan kelembapan perangkat Arduino.
6	device_switch_threshold	Menyimpan hubungan antara tabel device_switch dengan device_threshold untuk menentukan relai mana yang harus diaktifkan ketika sebuah <i>threshold</i> terpenuhi.
7	board	Menyimpan jenis <i>board</i> Arduino yang dapat digunakan.
8	board_pin_switch	Menyimpan kode pin dari <i>board</i> Arduino yang digunakan yang dapat dipakai untuk mengatur relai.

9	user	Menyimpan informasi terkait <i>user</i> yang dapat mengakses aplikasi web dan tingkat otoritasnya.
10	user_log	Menyimpan catatan terkait aktivitas <i>user</i> .

Dari delapan tabel yang digunakan, Tabel 4.7 hingga Tabel 4.16 menjelaskan deskripsi tiap kolom yang terdapat pada setiap tabel.

Tabel 4.7 Deskripsi Kolom Tabel *device*

Tabel: device	
Nama Kolom	Deskripsi
device_id (PK)	<i>Primary Key</i> perangkat Arduino.
device_name	Nama Perangkat Arduino.
board_id (FK)	Jenis <i>board</i> Arduino yang digunakan.
key	Kode autentikasi perangkat.
update_interval	Jangka waktu setiap kali perangkat melakukan pembacaan suhu dan kelembapan dalam satuan milidetik.

Tabel 4.8 Deskripsi Kolom Tabel *monitoring_data*

Tabel: monitoring_data	
Nama Kolom	Deskripsi
monitoring_id (PK)	<i>Primary Key</i> data.
device_id (FK)	Sumber perangkat Arduino.
suhu_data	Nilai suhu dalam Celsius.
kelembapan_data	Nilai kelembapan dalam <i>relative humidity</i> (%).
timestamp	Waktu dan tanggal data didapat.

Tabel 4. 9 Deskripsi Kolom Tabel kalibrasi_log

Tabel: kalibrasi_log	
Nama Kolom	Deskripsi
kalibrasi_id (PK)	<i>Primary Key</i> relai.
device_id (FK)	Perangkat Arduino yang dikalibrasi
Kalibrasi_timestamp	Waktu dan tanggal kalibrasi
suhu_kalibrasi	Kalibrasi nilai suhu terhadap nilai yang didapat dari perangkat.
kelembapan_kalibrasi	Kalibrasi nilai kelembapan terhadap nilai yang didapat dari perangkat.

Tabel 4. 10 Deskripsi Kolom Tabel device_switch

Tabel: device_switch	
Nama Kolom	Deskripsi
device_switch_id (PK)	<i>Primary Key</i> relai.
device_id (FK)	Sumber perangkat Arduino tempat relai terpasang.
switch_name	Nama relai yang dipasang.
switch_pin	Kode pin pada perangkat Arduino yang terhubung dengan relai.

Tabel 4. 11 Deskripsi Kolom Tabel device_threshold

Tabel: device_threshold	
Nama Variabel	Deskripsi
device_threshold_id (PK)	<i>Primary Key threshold.</i>
device_id (FK)	Perangkat Arduino yang memiliki <i>threshold.</i>
dht_data	Informasi yang akan digunakan untuk membandingkan. Variabel ini hanya terdapat dua pilihan, yaitu suhu atau kelembapan
logical_operator	Notasi matematika yang ingin digunakan untuk melakukan

	perbandingan. Variabel ini hanya terdapat tiga pilihan, yaitu >, <, dan =.
dht_data_value	Nilai yang ingin dibandingkan.
description	Deskripsi dari <i>threshold</i> yang dibuat.

Tabel 4. 12 Deskripsi Kolom Tabel device_switch_threshold

Tabel: device_switch_threshold	
Nama Variabel	Deskripsi
device_switch_threshold_id (PK)	<i>Primary Key</i> hubungan antara device_switch dengan device_threshold.
device_switch_id (FK)	Relai yang ingin diaktifkan.
device_threshold_id (PK)	<i>Threshold</i> yang harus terpenuhi.

Tabel 4. 13 Deskripsi Kolom Tabel board

Tabel: board	
Nama Variabel	Deskripsi
board_id (PK)	<i>Primary Key</i> board Arduino.
board_name	Nama board Arduino yang dapat digunakan.

Tabel 4. 14 Deskripsi Kolom Tabel board_pin_switch

Tabel: board_pin_switch	
Nama Variabel	Deskripsi
board_id (FK)	Board Arduino yang digunakan.
pin_name (PK)	Kode pin yang terdapat pada board Arduino.
pin_uint8	Kode pin yang dimengerti oleh kode program Arduino.

Tabel 4. 15 Deskripsi Kolom Tabel user

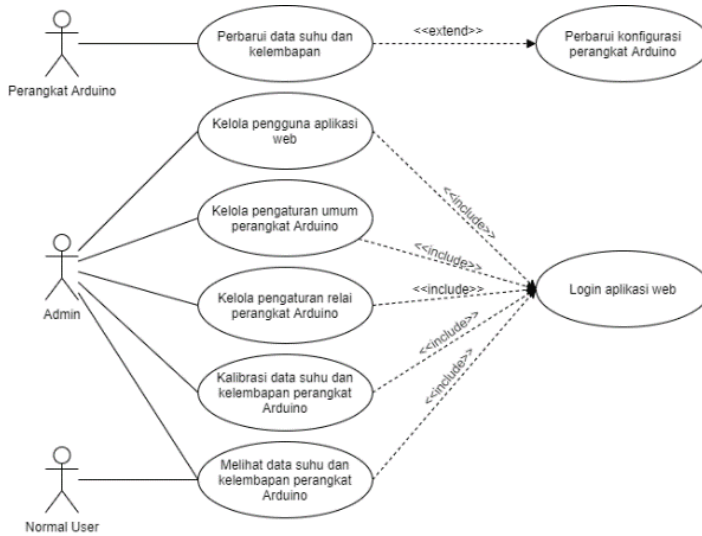
Tabel: user	
Nama Variabel	Deskripsi
Username (PK)	Tanda pengenalan pengguna.
password	Kata sandi pengguna untuk mengakses aplikasi web.
acclvl	Level otoritas user dalam mengakses aplikasi web.

Tabel 4. 16 Deskripsi Kolom Tabel user_log

Tabel: user_log	
Nama Variabel	Deskripsi
user_log_id (PK)	<i>Primary Key</i> dari catatan akses pengguna.
Username (FK)	Tanda pengenalan pengguna yang mengakses aplikasi web.
timestamp	Waktu dan tanggal pengguna mengakses aplikasi web.

4.4.2 Use Case Aplikasi Web

Use case dibuat dengan tujuan untuk mengetahui fungsi utama dari sistem yang akan dibuat dan aktor yang menggunakan fungsi tersebut. *Use case* aplikasi web dibuat dari analisa kebutuhan fungsional yang terdapat pada Tabel 4.2. *Use Case* dari aplikasi web dapat dilihat pada Gambar 4.7. Dari tiap *use case* yang ada, dijabarkan lebih lanjut pada *use case description* pada Tabel 4.17 hingga Tabel 4.24



Gambar 4. 7 Diagram *Use Case* Aplikasi Web

Tabel 4. 17 *Use Case Description 1*

Use Case 1	Perbarui Data Suhu dan Kelembapan
Actor	Perangkat Arduino
Description	Aktor mengirim data suhu dan kelembapan terbaru untuk disimpan oleh aplikasi web.
Pre-Condition	Perangkat Arduino memiliki <i>key</i> , data suhu, dan data kelembapan.
Basic Course	<ol style="list-style-type: none"> 1. Aktor mengirim <i>key</i>, info suhu, dan info kelembapan 2. Sistem memeriksa seluruh parameter sesuai 3. Sistem autentikasi perangkat Arduino 4. Sistem menyimpan data suhu dan kelembapan perangkat Arduino
Alternate Course	2.1. Sistem menemukan parameter yang tidak memiliki nilai

	<ul style="list-style-type: none"> • Sistem memberi notifikasi aktor terdapat parameter yang tidak sesuai <p>3.1. Autentikasi perangkat Arduino gagal</p> <ul style="list-style-type: none"> • Sistem memberi notifikasi aktor terkait perangkat tidak terdaftar dalam sistem
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabel 4. 18 *Use Case Description 2*

Use Case 2	Perbarui Konfigurasi Perangkat Arduino
Actor	Perangkat Arduino
Description	Aplikasi web memperbarui konfigurasi perangkat Arduino untuk menyesuaikan dengan data yang terbaru.
Pre-Condition	Aktor telah memperbarui data suhu dan kelembapan.
Basic Course	<ol style="list-style-type: none"> 1. Sistem menentukan status relai pada perangkat Arduino berdasarkan data suhu dan kelembapan terbaru 2. Sistem memberi notifikasi aktor terkait konfigurasi status relai terbaru dan jangka waktu setiap perbaruan data.
Alternate Course	-

Tabel 4. 19 *Use Case Description 3*

Use Case 3	Login ke Aplikasi Web
Actor	Admin, Normal User
Description	Aktor melakukan autentikasi untuk menentukan otoritas pada aplikasi web.
Pre-Condition	Aktor mengakses aplikasi web.
Basic Course	<ol style="list-style-type: none"> 1. Aktor mengakses halaman <i>login</i> 2. Sistem menampilkan halaman <i>login</i>

	<ol style="list-style-type: none"> 3. Aktor mengisi form <i>username</i> dan <i>password</i> 4. Aktor menekan tombol “Login” 5. Sistem melakukan autentikasi dan menentukan otoritas aktor 6. Sistem mencatat tanggal login aktor 7. Sistem menampilkan halaman <i>dashboard</i>
Alternate Course	<ol style="list-style-type: none"> 5.1. Sistem gagal autentikasi aktor <ul style="list-style-type: none"> • Sistem mengarahkan kembali ke halaman <i>login</i> dengan pesan gagal <i>login</i>.

Tabel 4. 20 Use Case Description 4

Use Case 4 Kelola Pengguna Aplikasi Web	
Actor	Admin
Description	Aktor mampu menambah, menghapus, dan melihat catatan <i>login</i> akun pengguna aplikasi web.
Pre-Condition	Aktor telah <i>login</i>
Basic Course	<ol style="list-style-type: none"> 1. Aktor mengakses halaman manajemen pengguna 2. Sistem menampilkan halaman manajemen pengguna 3. Aktor menggunakan fitur pada manajemen pengguna
Alternate Course	-

Tabel 4. 21 Use Case Description 5

Use Case 5 Kelola Pengaturan Umum Perangkat Arduino	
Actor	Admin

Description	Aktor mampu menambah, mengubah, dan menghapus perangkat Arduino yang dapat terhubung ke aplikasi web.
Pre-Condition	Aktor telah <i>login</i>
Basic Course	<ol style="list-style-type: none"> 1. Aktor mengakses halaman manajemen perangkat 2. Sistem menampilkan halaman manajemen perangkat 3. Aktor menggunakan fitur pada manajemen perangkat
Alternate Course	-

Tabel 4. 22 Use Case Description 6

Use Case 6	Kelola Pengaturan Relai Perangkat Arduino
Actor	Admin
Description	Aktor mampu menambah, mengubah, dan menghapus relai serta menentukan status relai berdasarkan <i>threshold</i> .
Pre-Condition	Aktor telah <i>login</i>
Basic Course	<ol style="list-style-type: none"> 1. Aktor mengakses halaman manajemen <i>threshold</i> 2. Sistem menampilkan halaman manajemen perangkat 3. Aktor menggunakan fitur pada manajemen <i>threshold</i>
Alternate Course	-

Tabel 4. 23 Use Case Description 7

Use Case 7	Kalibrasi Data Suhu dan Kelembapan Perangkat Arduino
Actor	Admin

Description	Aktor melakukan kalibrasi data suhu dan kelembapan yang didapat dari perangkat Arduino.
Pre-Condition	<ol style="list-style-type: none"> 1. Aktor telah <i>login</i> 2. Aktor telah melihat data suhu dan kelembapan perangkat Arduino
Basic Course	<ol style="list-style-type: none"> 1. Aktor menekan tombol kalibrasi suhu dan kelembapan 2. Sistem menampilkan <i>form</i> kalibrasi 3. Aktor memasukkan nilai kalibrasi 4. Aktor menekan tombol “update” 5. Sistem menyimpan kalibrasi yang telah dimasukkan oleh aktor 6. Sistem menampilkan kembali halaman untuk melihat data suhu dan kelembapan
Alternate Course	-

Tabel 4. 24 Use Case Description 8

Use Case 8	Melihat Data Suhu Dan Kelembapan Perangkat Arduino
Actor	Admin, Normal <i>User</i>
Description	Aktor melihat data suhu dan kelembapan yang didapat dari perangkat Arduino.
Pre-Condition	Aktor telah <i>login</i>
Basic Course	<ol style="list-style-type: none"> 1. Aktor mengakses halaman perangkat yang ingin dilihat 2. Sistem menampilkan data suhu dan kelembapan dari perangkat yang dipilih
Alternate Course	-

4.4.3 Desain Aplikasi Web

Aplikasi web yang akan dibangun menggunakan *framework* CodeIgniter dimana *framework* tersebut menggunakan Model-View-Controller *design pattern*. Tiga komponen yang telah disebutkan, yaitu *model*, *view*, dan *controller* memiliki fungsinya masing-masing.

Komponen *model* mewakili data yang digunakan. Komponen *model* adalah sebuah *class* yang memiliki fungsi untuk mengambil, memasukkan, menghapus, atau memperbarui data. Selanjutnya adalah komponen *view*, komponen ini berfungsi untuk mempresentasikan data kepada pengguna melalui antarmuka. Terakhir adalah komponen *controller* yang merupakan sebuah *class* yang berfungsi untuk memproses HTTP *request*. Komponen *controller* adalah perantara antara komponen *view* dengan *model*. Pengguna mengakses aplikasi web dengan menggunakan URL dimana tiap segmen dari URL adalah alamat untuk fungsi dari *controller* yang ingin diakses. Segmen pertama dari URL menunjuk pada *class controller* dan segmen kedua dari URL menunjuk pada fungsi dari *controller*. Sebuah *routing* bisa digunakan untuk melakukan kustomisasi URL. Berdasarkan *framework* yang digunakan dan *use case* yang telah didefinisikan, Tabel 4.25 hingga 4.27 menjelaskan komponen-komponen yang akan digunakan pada aplikasi web sesuai dengan Model-View-Controller *design pattern*.

Tabel 4. 25 Daftar *Model* pada Aplikasi Web

Nama Model	Deskripsi	<i>Use Case</i>
device_model	Berisi fungsi-fungsi untuk mengolah data dari basis data terkait perangkat Arduino.	1, 2, 5, 6, 7, 8
user_model	Berisi fungsi-fungsi untuk mengolah data dari basis data terkait pengguna aplikasi web.	3, 4

Tabel 4. 26 Daftar *Controller* pada Aplikasi Web

Nama Controller	Deskripsi	Use Case
AdminUser	Berisi fungsi untuk memproses dan menampilkan yang berkaitan dengan pengelolaan pengguna aplikasi web.	4
AdminDevice	Berisi fungsi untuk memproses dan menampilkan yang berkaitan dengan pengelolaan perangkat Arduino.	5
Api	Berisi fungsi untuk menerima dan merespons HTTP <i>request</i> dari perangkat Arduino.	1, 2
Dashboard	Berisi fungsi untuk menampilkan halaman utama aplikasi web.	-
Device	Berisi fungsi untuk menampilkan dan mengelola data suhu dan kelembapan dari tiap perangkat Arduino.	7, 8
DeviceThreshold	Berisi fungsi untuk memproses dan menampilkan yang berkaitan dengan pengaturan <i>threshold</i> dan relai pada tiap perangkat Arduino.	6
UserSession	Berisi fungsi yang berkaitan dengan autentikasi dan otorisasi pengguna.	3

Tabel 4. 27 Daftar *View* pada Aplikasi Web

Nama View	Deskripsi	Use Case
dashboard	Menampilkan halaman utama dari aplikasi web.	-

device	Menampilkan halaman yang berisi data suhu dan kelembapan dari perangkat Arduino.	7, 8
devicemanagement	Menampilkan halaman untuk mengelola perangkat Arduino.	5
devicethreshold	Menampilkan halaman untuk mengatur <i>threshold</i> dan relai yang ada pada perangkat Arduino.	6
login	Menampilkan halaman untuk melakukan autentikasi pengguna.	3
usermanagement	Menampilkan halaman untuk mengelola pengguna aplikasi web.	4

Komponen *controller* sebagai perantara antara komponen *view* dan model memiliki beberapa fungsi pada tiap *controller* yang telah disebutkan pada Tabel 4.24. Fungsi-fungsi tersebut memudahkan pengguna untuk menampilkan data atau mengolah data yang ada pada basis data. Tabel 4.28 hingga 4.34 menjelaskan deskripsi masing-masing fungsi yang digunakan pada tiap *controller*.

Tabel 4. 28 Fungsi pada *Controller AdminUser*

Nama <i>Controller</i> : AdminUser	
Nama Fungsi	Deskripsi
view()	Menampilkan halaman manajemen pengguna.
addUser()	Menambah pengguna aplikasi web
deleteUser()	Menghapus pengguna aplikasi web.

Tabel 4. 29 Fungsi pada *Controller AdminDevice*

Nama <i>Controller</i> : AdminDevice	
Nama Fungsi	Deskripsi
view()	Menampilkan halaman manajemen perangkat Arduino.
addDevice()	Menambah perangkat Arduino yang dapat terhubung.
deleteDevice()	Menghapus perangkat Arduino yang dapat terhubung.
editDevice()	Mengubah informasi perangkat Arduino.
newKeyDevice()	Memperbarui <i>key</i> perangkat Arduino untuk autentikasi.

Tabel 4. 30 Fungsi pada *Controller Api*

Nama <i>Controller</i> : Api	
Nama Fungsi	Deskripsi
dhtswitch()	Menerima dan menyimpan data suhu dan kelembapan dari perangkat Arduino serta memberi respons konfigurasi terbaru pada perangkat Arduino.

Tabel 4. 31 Fungsi pada *Controller Dashboard*

Nama <i>Controller</i> : Dashboard	
Nama Fungsi	Deskripsi
index()	Menampilkan halaman utama aplikasi web.
allDeviceLatestMonData()	Mendapatkan data suhu dan kelembapan terbaru dari tiap perangkat Arduino untuk ditampilkan pada halaman utama

Tabel 4. 32 Fungsi pada *Controller Device*

Nama <i>Controller</i>: Device	
Nama Fungsi	Deskripsi
view()	Menampilkan data suhu dan kelembapan dari perangkat Arduino.
downloadCSV()	Mengunduh data suhu dan kelembapan perangkat Arduino dengan format CSV.
deleteMonitoringData()	Menghapus data suhu dan kelembapan perangkat Arduino.
calibrateMonitoringDataDevice()	Memperbarui kalibrasi data suhu dan kelembapan yang didapat dari perangkat Arduino.
deviceLatestMonitoringData()	Mendapatkan data suhu dan kelembapan terbaru yang belum ditampilkan sebelumnya.
deviceDateRangeMonitoringData()	Mendapatkan data suhu dan kelembapan dengan jarak tanggal yang diinginkan.
deviceInsightMonitoringData()	Mendapatkan info suhu dan kelembapan maksimum, minimum, dan rata-rata dari keseluruhan data.

Tabel 4. 33 Fungsi pada *Controller DeviceThreshold*

Nama <i>Controller</i> : DeviceThreshold	
Nama Fungsi	Deskripsi
view()	Menampilkan halaman manajemen <i>threshold</i> perangkat Arduino.
newDeviceThreshold()	Menambah <i>threshold</i> suhu atau kelembapan dari perangkat Arduino.
editDeviceThreshold()	Mengubah <i>threshold</i> suhu atau kelembapan perangkat Arduino.
deleteDeviceThreshold()	Menghapus <i>threshold</i> yang dipilih dari perangkat Arduino.
newSwitchDevice()	Menambah relai yang dapat diatur pada aplikasi web.
editSwitchDevice()	Mengubah informasi relai yang diatur.
deleteSwitchDevice()	Menghapus relai yang dapat diatur.
getSwitchThresholdTable()	Mendapatkan data <i>threshold</i> yang dapat diaktifkan pada relai.
enableSwitchThresholdCheck()	Mengaktifkan pemeriksaan <i>threshold</i> pada relai.

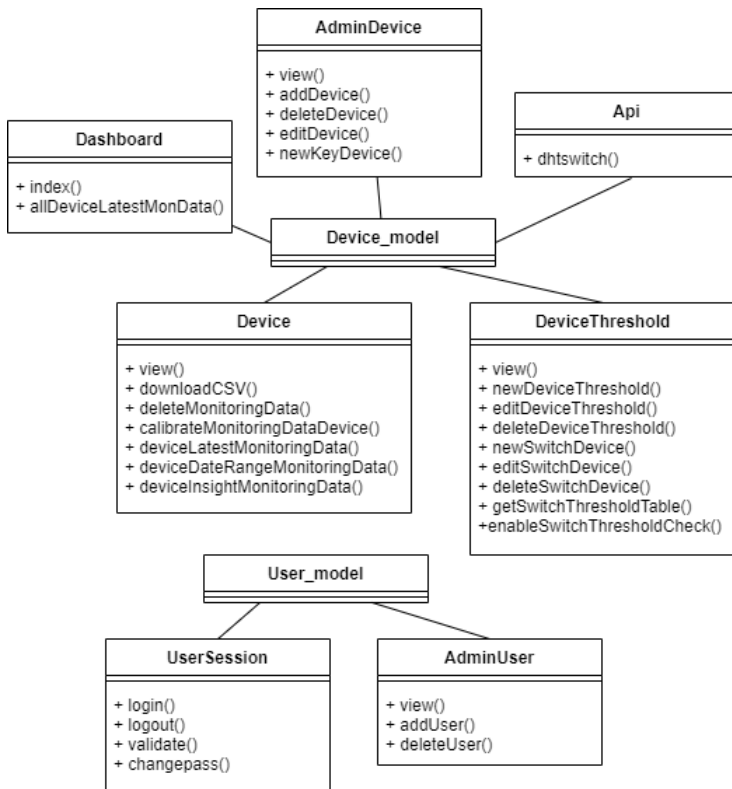
Tabel 4. 34 Fungsi pada *Controller UserSession*

Nama <i>Controller</i> : UserSession	
Nama Fungsi	Deskripsi
login()	Menampilkan halaman <i>login</i>
logout()	Menghilangkan akses pengguna pada aplikasi web.

validate()	Melakukan autentikasi dan menentukan otorisasi pengguna.
changepass()	Memperbarui <i>password</i> pengguna.

4.4.4 Class Diagram Aplikasi Web

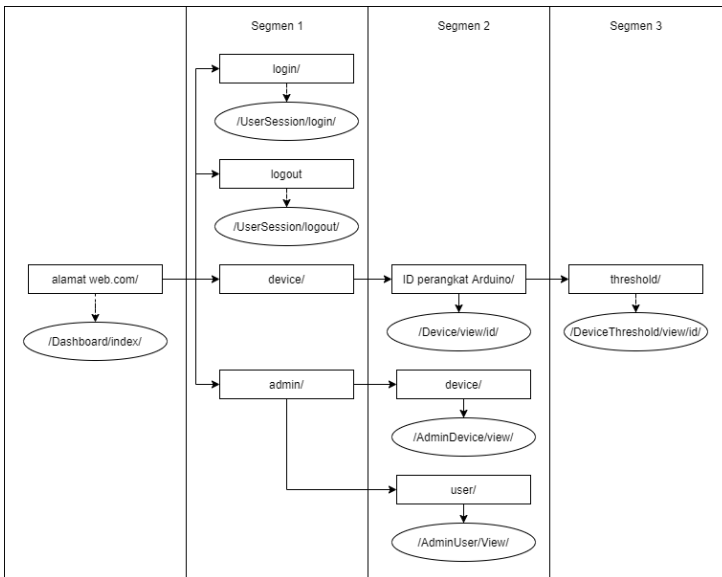
Class diagram dibuat dengan tujuan untuk mengetahui secara konseptual sistem yang akan dibuat. *Class diagram* berisi *class* apa saja yang akan digunakan pada sistem dan hubungan tiap *class*. *Class diagram* yang dibuat adalah hasil dari tahap sebelumnya. *Class diagram* dapat dilihat pada Gambar 4.8.



Gambar 4. 8 Class Diagram Aplikasi Web

4.4.5 Desain *Routing* Aplikasi Web

Framework CodeIgniter memiliki fitur *routing* yang dapat memetakan URL yang akan diakses pengguna aplikasi web ke sumber daya yang diinginkan dengan menggunakan nama lain dari sumber daya yang diakses. Dalam hal ini, sumber daya yang dimaksud adalah fungsi yang terdapat pada *controller*. Secara *default*, Pengguna dapat mengakses fungsi pada *controller* dengan menggunakan menggunakan pola URL “alamatweb.com/segmen1/segmen2/segmen3” dimana segmen 1 adalah nama *class* dari *controller*, segmen 2 adalah nama fungsi yang ingin diakses pada *controller* tersebut, dan segmen 3 adalah parameter dari fungsi yang ingin diakses.



Gambar 4. 9 Desain *Routing* Aplikasi Web

Gambar 4.9 adalah desain kustomisasi URL dengan menggunakan *routing* pada aplikasi web. Pada gambar tersebut dijelaskan urutan *routing* dari awal alamat web hingga segmen terakhir. Setiap kustomisasi URL pada segmen memiliki alamat *controller* dan fungsi yang akan dijalankan ketika mengakses URL hasil *routing*.

4.5 Desain *Testing* Perangkat Keras

Testing dari perangkat Arduino akan menggunakan metode pengujian *black box*. Beberapa *test case* akan dibuat untuk menguji apakah perangkat Arduino yang telah diimplementasikan sudah sesuai dengan kebutuhan. Setiap *test case* memiliki ekspektasi yang diharapkan dari pengujian *test case* tersebut. Selain menggunakan *test case* dilakukan juga uji akurasi sensor DHT22 dan uji pin Wemos D1 Mini untuk relai. Tabel 4.33 adalah daftar *test case* yang akan diuji pada perangkat Arduino.

Tabel 4. 35 *Test Case* Perangkat Keras

<i>Test Case</i>	<i>Ekspektasi</i>
Terhubung dengan jaringan Wi-Fi	Perangkat Arduino terkoneksi dengan jaringan Wi-Fi yang telah ditentukan
Pembacaan sensor DHT22	Data suhu dan kelembapan dapat terbaca oleh perangkat Arduino dari sensor DHT22 dan ditampilkan pada Serial.
Mengirim data ke aplikasi web dengan protokol HTTP	HTTP <i>respons code</i> 200 dan mendapatkan data respons dari aplikasi web
Mengirim data ke aplikasi web dengan protokol HTTPS	HTTP <i>respons code</i> 200 dan mendapatkan data respons dari aplikasi web
Memperbarui konfigurasi relai dan <i>delay</i>	Konfigurasi relai dan <i>delay</i> menyesuaikan konfigurasi pada aplikasi web

4.6 Desain *Testing* Perangkat Lunak

Sama seperti desain *testing* perangkat keras, aplikasi web diuji coba menggunakan metode *black box testing*. Tabel 4.34 adalah daftar *test case* yang akan diuji pada aplikasi web.

Tabel 4. 36 Test Case Perangkat Lunak

Test Case	Skenario	Ekspektasi
<i>Login</i>	Pengguna mengetik <i>username</i> dan <i>password</i> dengan benar	Pengguna masuk ke halaman utama
	Pengguna mengetik <i>username</i> dan atau <i>password</i> yang salah	Pengguna mendapat <i>warning username</i> atau <i>password</i> salah
Menerima data dari perangkat Arduino	Parameter dan <i>key</i> sesuai	Data suhu dan kelembapan disimpan pada basis data dan memberi respons konfigurasi perangkat.
	Parameter dan atau <i>key</i> tidak sesuai	Memberi respons kesalahan parameter.
Melihat <i>dashboard</i>	Pengguna yang sudah <i>login</i> mengakses halaman utama	Halaman utama ditampilkan
Melihat data suhu dan kelembapan	Pengguna yang sudah <i>login</i> memilih perangkat Arduino pada <i>navigation bar</i>	Data suhu dan kelembapan dalam bentuk rangkuman, grafik, dan tabel ditampilkan
Kalibrasi data suhu dan kelembapan	Pengguna level admin mengisi <i>form</i> kalibrasi	Data kalibrasi tersimpan pada basis data

dari perangkat Arduino	suhu dan kelembapan perangkat Arduino	
Menghapus data suhu dan kelembapan	Pengguna level admin menekan tombol hapus pada halaman data suhu dan kelembapan perangkat Arduino	Data suhu dan kelembapan pada perangkat Arduino yang dipilih terhapus dari basis data
Mengunduh data suhu dan kelembapan perangkat Arduino	Pengguna menekan tombol unduh pada halaman data suhu dan kelembapan perangkat Arduino	Data suhu dan kelembapan dalam format CSV diunduh ke komputer pengguna.
Mengelola relai perangkat Arduino	Pengguna level admin menambah, mengubah, dan menghapus relai pada perangkat Arduino	Konfigurasi relai yang dilakukan pengguna tersimpan di basis data
Mengelola <i>threshold</i> perangkat Arduino	Pengguna level admin menambah, mengubah, dan menghapus <i>threshold</i> perangkat Arduino	Konfigurasi <i>threshold</i> tersimpan di basis data

Mengelola daftar perangkat Arduino yang terdaftar	Pengguna level admin menambah, mengubah, dan menghapus perangkat Arduino dan konfigurasinya	Perangkat Arduino dan konfigurasinya tersimpan pada basis data.
Mengelola daftar pengguna pada aplikasi web	Pengguna level admin menambah dan menghapus pengguna pada aplikasi web	Konfigurasi pengguna tersimpan pada basis data

BAB V IMPLEMENTASI

Pada bagian ini akan menjelaskan mengenai implementasi sistem berdasarkan desain yang telah dibuat pada tahap sebelumnya.

5.1 Lingkungan Implementasi Sistem

Pengembangan program yang digunakan oleh sistem baik program pada perangkat keras atau perangkat lunak menggunakan komputer dengan spesifikasi yang tertera pada tabel 5.1.

Tabel 5. 1 Spesifikasi Komputer Implementasi

<i>Processor</i>	Intel® Core™ i7-6700HQ CPU @2.60GHz
<i>Memory (RAM)</i>	16 GB
<i>Sistem Operasi</i>	Microsoft Windows 10 Home 64Bit
<i>Graphic Card</i>	Nvidia GeForce GTX 1060

5.2 Implementasi Perangkat Keras

Pengembangan perangkat Arduino dimulai dengan menentukan lingkungan implementasi yang kemudian dilanjutkan dengan merakit perangkat Arduino yang telah didesain sebelumnya dan membuat program untuk menjalankan perangkat Arduino tersebut.

5.2.1 Lingkungan Implementasi Perangkat Keras

Perangkat Arduino dikembangkan dengan menggunakan beberapa teknologi yang tersedia seperti *Integrated Development Environment (IDE)*, bahasa pemrograman, dan *library* Arduino yang ditunjukkan pada Tabel 5.2

Tabel 5. 2 Teknologi Pengembangan Perangkat Arduino

IDE	Arduino IDE ver. 1.8.8
Bahasa Pemrograman	C/C++

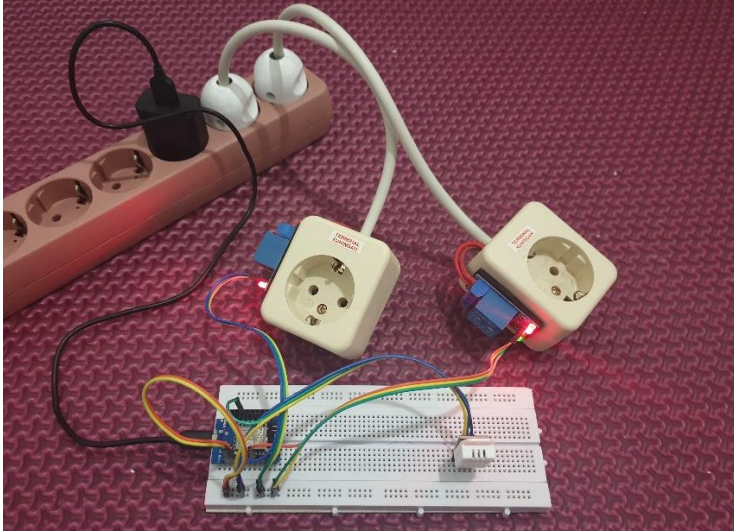
<i>Library</i> Arduino	<ol style="list-style-type: none"> 1. esp8266 ver. 2.7.1 2. Adafruit Unified Sensor ver. 1.0.3 3. DHT Sensor Library ver. 1.3.7 4. ArduinoJSON ver. 6.15.1
---------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5.2.2 Implementasi Rangkaian Perangkat Arduino

Perangkat Arduino dirangkai sesuai dengan skema perancangan yang telah dibuat pada Gambar 4.3. Dalam proses implementasi rangkaian perangkat Arduino, diperlukan sebuah alat tambahan, yaitu *breadboard*. Fungsi dari *breadboard* adalah untuk memudahkan dalam membuat prototipe dari skema perangkat Arduino dengan menyediakan media untuk menghubungkan pin dari Arduino menuju komponen yang diperlukan. Pada implementasi ini, *breadboard* digunakan untuk menambah pin 5V dan *ground* yang didapat dari Wemos D1 Mini. Penambahan pin 5V dan *ground* dibuat untuk memenuhi kebutuhan listrik komponen relai dan sensor DHT22. Sedangkan untuk memenuhi kebutuhan listrik Wemos D1 Mini, menggunakan 5V DC Adapter yang dihubungkan dengan Wemos D1 Mini menggunakan kabel *micro* USB.

Sebuah komponen stopkontak dan relai dibuat menjadi seperti modul dimana satu relai dan satu stopkontak dibuat menjadi satu dan terpisah dengan stopkontak dan relai yang lain. Pembuatan modul stopkontak dan relai yang dijadikan satu bertujuan untuk memudahkan penambahan atau pengurangan stopkontak sesuai dengan kebutuhan.

Sesuai dengan skema yang telah dibuat, sensor DHT22 menggunakan pin D7 untuk mengirimkan data ke Wemos D1 Mini. Gambar 5.1 menampilkan hasil implementasi rangkaian perangkat Arduino.



Gambar 5. 1 Implementasi Rangkaian Perangkat Arduino

5.2.3 Implementasi Program Perangkat Arduino

Pengembangan program perangkat Arduino dibuat menggunakan teknologi IDE dan *library* yang dijelaskan pada Tabel 5.2. Program dikembangkan menyesuaikan rancangan yang telah dibuat pada bab sebelumnya.

Kode 5.1 adalah potongan kode dari bagian awal program perangkat Arduino, yaitu mendefinisikan *global variable* yang akan digunakan pada program. Pada kode tersebut mendefinisikan konfigurasi dari perangkat seperti pengaturan Wi-Fi, pengaturan *key* yang didapat dari aplikasi web, pengaturan alamat aplikasi web, dan pengaturan sensor DHT22. Pada pengaturan alamat aplikasi web, jika ingin menggunakan koneksi HTTPS maka diperlukan *fingerprint* dari sertifikat HTTPS *host* aplikasi web.

```

1. // Network Config=====
2. const char* ssid = "SSID WiFi";
3. const char* password = "Password WiFi";
4. //=====
5.

```

```

6. //Device Config=====
7. const String key = "FFFFFFFFFFFFFFF";
8. int update_interval = 0;
9. //=====
10.
11. //Host Config=====
12. const bool https = true;
13. const char* url =
14. "https://[host aplikasi web]/api/dhtswitch/";
15. //Fingerprint HTTPS
16. const uint8_t fingerprint[20] = {0x5b, 0xfb, 0xd1, 0xd4,
    0x49, 0xd3, 0x0f, 0xa9, 0xc6, 0x40, 0x03, 0x34, 0xba,
    0xe0, 0x24, 0x05, 0xaa, 0xd2, 0xe2, 0x01};
17. //=====

```

Kode 5.1 *Global Variable* Program Arduino

```

1. void setup() {
2.   Atur baud rate untuk serial monitor menjadi 115200.
3.   Inisialisasi pembacaan sensor DHT22.
4.   connectWifi();
5. }

```

Kode 5.2 *Pseudocode* setup() Perangkat Arduino

Kode 5.2 mengimplementasikan *method* setup() dimana *method* tersebut hanya dijalankan sebanyak satu kali pada saat perangkat Arduino dinyalakan. Pada *method* tersebut, perangkat Arduino melakukan konfigurasi *baud rate* menjadi 115200, mengaktifkan *library* untuk menerima informasi dari sensor DHT22, dan mencoba terhubung dengan Wi-Fi sesuai konfigurasi pada *global variable*.

```

47. void loop() {
48.   if(Wi-Fi tidak terhubung dengan jaringan) {
49.     connectWifi();
50.   }
51.   readSensor();
52.
53.   switch(https) {
54.     case true:
55.       Konfigurasi koneksi ke aplikasi web dengan
56.       protokol HTTPS menggunakan fingerprint.
57.       break;
58.     case false:

```

```

59.     Konfigurasi koneksi ke aplikasi web dengan
60.     protokol HTTP.
61.     break;
62. }
63. postData =
64.     Key perangkat Arduino,
65.     Data suhu pembacaan sensor,
66.     Data kelembapan pembacaan sensor.
67.
68.     httpCode =
69.     Kirim POST request aplikasi web dengan data postData.
70.     if(httpCode == 200) {
71.         Perbarui nilai update_interval.
72.         Perbarui konfigurasi relai menggunakan method
73.         relayUpdate().
74.     } else {
75.         update_interval = 5000;
76.     }
77.
78.     Akhiri koneksi dengan aplikasi web.
79.     Menunggu loop selanjutnya menggunakan delay dengan waktu
80.     Sebanyak update_interval.
81. }

```

Kode 5.3 *Pseudocode* loop() Program Arduino

Kode 5.3 adalah *pseudocode method* loop() yang akan dijalankan berulang-ulang setelah *method setup* dijalankan. Pada awal *method*, dilakukan pengecekan apakah perangkat Arduino sudah terhubung dengan Wi-Fi. Jika belum maka akan menjalankan *method* connectWifi() untuk terhubung dengan jaringan Wi-Fi. Setelah memastikan perangkat Arduino telah terhubung, maka perangkat Arduino akan membaca data suhu dan kelembapan dari sensor DHT22.

Untuk mengirim data suhu dan kelembapan ke aplikasi web, hal pertama yang dilakukan adalah menentukan protokol HTTP atau HTTPS yang akan digunakan. Protokol ini ditentukan pada konfigurasi *global variable*. Protokol HTTPS memerlukan *fingerprnt* untuk mengenkripsi data yang dikirim. Data dikirim dengan menggunakan metode POST dengan mengirimkan tiga variabel yaitu *key*, data suhu, dan data kelembapan. Setelah mengirim data dengan menggunakan POST, perangkat Arduino

akan menunggu respons dari aplikasi web. Jika respons kode HTTP yang didapat adalah 200 OK, maka perangkat Arduino akan membaca respons yang didapat. Jika gagal, perangkat Arduino akan *delay* selama lima detik dan mencoba ulang dengan menjalankan kembali *method* `loop()`.

Pembacaan respons dilakukan untuk memperbarui konfigurasi relai dan *delay method* `loop()` sesuai dengan konfigurasi yang ada pada aplikasi web. Respons yang berupa JSON dipecah menggunakan *library* `ArduinoJSON` untuk memudahkan mengubah konfigurasi sesuai respons.

```

121. void connectWifi() {
122.   Atur mode Wi-Fi perangkat Arduino menjadi client
123.   (WIFI_STA).
124.
125.   Lakukan percobaan koneksi dengan jaringan Wi-Fi sesuai
126.   dengan SSID dan password.
127.
128.   if(Wi-Fi tidak terhubung lebih dari 20 detik) {
129.     Reset perangkat Arduino.
130.   }
131. }
```

Kode 5.4 Pseudocode `connectWifi()` Program Arduino

Pseudocode `connectWifi()` pada Kode 5.4 adalah *method* pendukung untuk melakukan koneksi pada jaringan Wi-Fi. Perangkat diatur pada mode `WIFI_STA` yang berarti perangkat akan bertindak sebagai *client* yang akan terhubung ke *access point* yang telah ditentukan pada *global variables*. Jika lebih dari 20 detik dan perangkat Arduino gagal terhubung dengan jaringan Wi-Fi, maka akan dilakukan reset. *Method* `connectWifi()` digunakan pada *method* `setup()` dan pada *method* `loop()` disaat Wemos D1 Mini terputus dari koneksi Wi-Fi.

```

147. void readSensor() {
148.   temp = Baca nilai suhu dari sensor DHT22.
149.   humid = Baca nilai kelembapan dari sensor DHT22.
150. }
```

Kode 5.5 Pseudocode `readSensor()` Program Arduino

Kode 5.5 adalah *method* pendukung `readSensor()` untuk melakukan pembacaan sensor DHT22. Pembacaan nilai suhu dan kelembapan dari sensor dibantu oleh *library* yang memudahkan proses pengambilan nilai suhu dan kelembapan.

```

150. void relayUpdate(JsonArray json) {
151.   for(semua pengaturan relay yang didapat dari aplikasi
152.     web) {
153.     switch_pin = Kode pin relay.
154.     enable = Status relay.
155.
156.     if(enable==0) {
157.       Matikan relay yang terhubung dengan pin
158.         switch_pin.
159.     } else {
160.       Nyalakan relay yang terhubung dengan pin
161.         switch_pin.
162.     }
163.   }
164. }

```

Kode 5. 6 *Pseudocode* `relayUpdate()` Program Arduino

Method pendukung `relayUpdate()` pada Kode 5.6 digunakan untuk memperbarui konfigurasi relay yang terdapat pada perangkat Arduino. *Method* ini digunakan pada *method* `loop()` ketika berhasil mendapatkan respons dari aplikasi web. Dari *JSON Array* yang didapat, setiap *array* memiliki kode pin dari relay yang terpasang dan statusnya apakah harus dinyalakan atau tidak. Untuk mengatur seluruh konfigurasi yang didapat digunakan perintah *for* sebanyak jumlah konfigurasi relay. Setiap perulangan perintah *for*, relay dimatikan atau dinyalakan dengan cara mengubah mode pin yang terhubung dengan relay.

5.3 Implementasi Perangkat Lunak

Pengembangan perangkat lunak pada sistem ini mengacu pada desain aplikasi web yang telah dibuat pada tahap sebelumnya. Aplikasi web dikembangkan dengan membuat kode program yang mampu memenuhi kebutuhan pengguna.

5.3.1 Lingkungan Implementasi Perangkat Lunak

Perangkat lunak dikembangkan dengan menggunakan beberapa teknologi yang tersedia seperti *web server*, bahasa pemrograman, *framework*, *database*, *Integrated Development Environment (IDE)*, dan *library* yang ditunjukkan pada Tabel

Tabel 5. 3 Teknologi Pengembangan Aplikasi Web

<i>Web Server</i>	Apache
Bahasa Pemrograman	1. PHP ver. 7.2.27 2. HTML 3. JavaScript 4. SQL
<i>Framework</i>	CodeIgniter ver. 3.1.11
<i>Database</i>	MySQL
IDE	1. MySQL Workbench ver. 8.0 2. PhpStorm 2019.3
<i>Library</i>	SB Admin ver. 6

5.3.2 Implementasi Basis Data

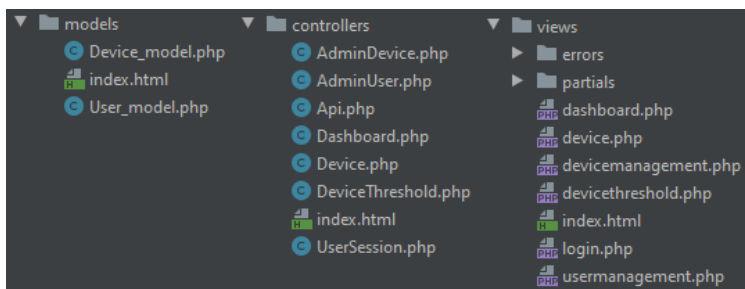
Implementasi basis data yang digunakan untuk menyimpan data yang digunakan oleh aplikasi web menggunakan sistem basis data MySQL. Penggunaan IDE MySQL Workbench memudahkan proses implementasi karena tidak membutuhkan menulis kode secara manual. IDE yang digunakan memiliki fitur “*Forward Engineering*” dimana kode implementasi basis data dapat dihasilkan secara otomatis berdasarkan rancangan basis data yang dibuat. Kode hasil dari “*Forward Engineering*” dapat langsung dimasukkan ke dalam *server*.

Implementasi *data definition language (DDL)* dari basis data yang akan digunakan pada aplikasi web disesuaikan dengan desain yang dibuat pada tahap sebelumnya. Terdapat delapan tabel yang dibuat dengan tiap tabel memiliki konfigurasi masing-masing. Setiap tabel didefinisikan nama tabel, nama tiap kolom pada tabel, tipe data tiap kolom, *primary key* tabel, serta *foreign key* (jika ada) untuk mengetahui hubungan tabel dengan tabel lainnya. Sedangkan untuk implementasi *data*

manipulation language (DML) pada basis data yang digunakan, beberapa data *default* dimasukkan pada tabel yang telah dibuat. Terdapat tiga tabel yang memiliki nilai data *default*, yaitu tabel “board”, “user”, dan “board_pin_switch”. Tabel “board” secara *default* memiliki satu data berisi nama Arduino *board* yang akan digunakan oleh perangkat Arduino yaitu Wemos D1 Mini. Tabel “user” berisi satu data yaitu *username* admin dan passwordnya. Terakhir adalah tabel “board_pin_switch” yang berisi kode pin *board* Arduino yang dapat digunakan untuk mengatur relai.

5.3.3 Implementasi Aplikasi Web

Aplikasi web diimplementasikan sesuai dengan rancangan pada tahap sebelumnya. Dengan menggunakan *framework* CodeIgniter, implementasi arsitektur dengan menggunakan *design pattern* MVC menjadi lebih mudah karena telah disediakan direktori untuk setiap *model*, *view*, dan *controller* yang dibutuhkan oleh aplikasi web. Gambar 5.2 adalah implementasi arsitektur dari aplikasi web.



Gambar 5. 2 Implementasi Arsitektur Aplikasi Web

5.3.3.1 Implementasi Model

Terdapat dua *model* yang diimplementasikan pada aplikasi web, yaitu Device_model dan User_model. Sebuah model dalam implementasi aplikasi web berfungsi untuk mengambil dan atau mengubah data pada tabel basis data.

```
1. class Device_model extends CI_Model {
2.     public function getAllDevicesInfo() {
```

```
3.     Select query pada database informasi umum perangkat
4.     Arduino yang terdaftar.
5.     return hasil query;
6. }
7.
8.     public function updateDeviceMonitoringData($device_id,
9.         $suhu_data, $kelembapan_data) {
10.
11.     Insert query pada database hasil penjumlahan data
12.     suhu dan kelembapan yang didapat dengan data
13.     kalibrasi pada perangkat tersebut.
14. }
15.
16.     public function getAllDeviceLatestMonitoringData() {
17.     Select query pada database data suhu dan kelembapan
18.     terbaru pada seluruh perangkat Arduino yang terdaftar.
19.     return hasil query;
20. }
21.
22.     public function calibrateDeviceMonitoring($device_id,
23.         $suhu, $kelembapan) {
24.
25.     Insert query pada database nilai kalibrasi suhu dan
26.     kelembapan pada perangkat Arduino.
27. }
28.
29.     public function addDeviceSwitch($device_id,
30.         $switch_name, $switch_pin) {
31.
32.     Insert query pada database nama relai dan kode pin
33.     relai yang terdapat pada perangkat Arduino.
34.     return hasil query;
35. }
36.
37.     public function addDeviceThreshold($device_id,
38.         $dht_data, $logical_operator, $dht_data_value,
39.         $description) {
40.
41.     Insert query pada database nilai suhu atau kelembapan
42.     threshold dan deskripsi dari threshold tersebut pada
43.     perangkat Arduino.
44. }
45.
46.     public function getDeviceSwitchThreshold(
47.         $device_switch_id) {
```

```

48.
49.     Select join query pada database daftar relay yang
50.     terdaftar pada perangkat Arduino yang dipilih dengan
51.     daftar threshold pada perangkat Arduino tersebut.
52.     return hasil query;
53. }
54. }

```

Kode 5. 7 Pseudocode Model “Device_model”

Kode 5.9 adalah potongan *pseudocode* dari implementasi model Device_model. Setiap fungsi yang ada pada *model* ini berfungsi untuk mengambil atau mengubah data yang ada pada tabel yang berhubungan dengan perangkat Arduino. Secara total terdapat 41 fungsi pada *model* ini dimana masing-masing fungsi memiliki *query* SQL yang berbeda-beda. Fungsi dimulai dengan menentukan parameter yang dibutuhkan untuk melakukan *query* SQL. Setelah itu *String query* dibuat menyesuaikan tujuan dari tiap fungsi. Kemudian *query* dijalankan dengan menggunakan *method* query() dengan parameter *String query* dan *array* yang berisi parameter *query* yang diinginkan. Hasil dari *query* adalah sebuah *array*.

```

1.  class User_model extends CI_Model {
2.      public function addUser($username, $password,
3.          $admin) {
4.
5.          Insert query pada database akun pada aplikasi web
6.          yang ingin ditambahkan beserta access level akun
7.          tersebut.
8.      }
9.
10.     public function deleteUser($username) {
11.         Delete query pada database akun pada aplikasi web
12.         pada username yang disebutkan.
13.     }
14.
15.     public function getUser($username) {
16.         Select query pada database seluruh info akun pada
17.         username yang disebutkan.
18.         return hasil query;
19.     }
20. }

```

```

21. public function changePassword($username,
22.     $password) {
23.
24.     Update query pada database password akun aplikasi
25.     web pada username yang disebutkan.
26. }
27.
28. public function logUser($username) {
29.     Insert query pada database timestamp terkini
30.     akun yang berhasil login pada aplikasi web.
31. }
32.
33. }

```

Kode 5. 8 Pseudocode Model “User_model”

Kode 5.10 adalah *pseudocode* dari *model* User_model. Pada model tersebut berisi fungsi-fungsi untuk mengambil dan memanipulasi data yang ada pada tabel “user” dan “user_log” yang terdapat pada basis data. Total terdapat delapan fungsi pada model tersebut untuk mengambil atau memanipulasi. Beberapa tujuan dari fungsi yang ada pada *model* ini adalah antara lain untuk menambah dan menghapus pengguna, mengambil data pengguna, mengubah *password* pengguna, mencatat waktu *login* pengguna, serta mengambil daftar catatan *login* pengguna.

5.3.3.2 Implementasi *Controller*

Controller yang diimplementasikan pada aplikasi web memiliki tujuannya masing-masing sesuai dengan yang dijelaskan pada tahap sebelumnya. Setiap *controller* terhubung dengan *model* yang dibutuhkan. *Controller* yang bertujuan untuk memproses data yang berhubungan dengan perangkat Arduino akan menggunakan *model* Device_model dan *controller* yang bertujuan untuk memproses data yang berhubungan dengan pengguna akan menggunakan *model* “User_model”.

```

1. class AdminDevice extends CI_Controller {
2.     public function view() {
3.         if(admin.telahLogin()) {
4.             Tampilkan halaman web pengelolaan perangkat
5.             Arduino menggunakan komponen view

```

```

6.     "devicemanagement".
7.     } else {
8.         redirect(halaman utama);
9.     }
10.    }
11.
12.    public function addDevice() {
13.        $this->adminCheckDevice();
14.        $key = $this->$generateRandKey();
15.        Masukkan info perangkat Arduino terbaru ke database
16.        berdasarkan data dari POST dan $key menggunakan fungsi
17.        pada "Device_model".
18.    }
19.
20.    public function deleteDevice() {
21.        $this->adminCheckDevice();
22.        Hapus perangkat Arduino dari database berdasarkan
23.        $device_id menggunakan fungsi pada "Device_model".
24.    }
25.
26.    public function editDevice() {
27.        $this->adminCheckDevice();
28.        Ubah info perangkat Arduino pada database
29.        berdasarkan data dari POST menggunakan fungsi
30.        pada "Device_model".
31.    }
32.
33.    public function newKeyDevice() {
34.        $this->adminCheckDevice();
35.        $key = $this->$generateRandKey();
36.        Perbarui key pada perangkat Arduino berdasarkan
37.        $device_id dan $key menggunakan fungsi pada
38.        "Device_model".
39.    }
40.
41. }

```

Kode 5. 9 Pseudocode Controller AdminDevice

Kode 5.11 adalah *pseudocode* dari *controller* AdminDevice yang berfungsi untuk menerima dan merespons *request* pengguna yang berhubungan dengan pengelolaan perangkat Arduino. Fungsi dari *controller* ini hanya bisa diakses oleh pengguna yang memiliki level akses 1 atau admin. Ketika fungsi *view()* dari *controller* tersebut diakses, maka aplikasi

web akan menampilkan halaman web dari komponen *view* *devicemanagement* yang berisi daftar perangkat Arduino yang terdaftar pada aplikasi web. Beberapa fungsi lain yang terdapat pada *controller* ini seperti *addDevice()*, *editDevice()*, dan *deleteDevice()* berfungsi untuk menambah, mengubah, atau menghapus perangkat Arduino yang terdaftar pada aplikasi web. Setiap perangkat Arduino yang didaftarkan memiliki *key* dengan tujuan untuk autentikasi data yang diterima oleh aplikasi web sesuai dengan nama perangkat Arduino data tersebut berasal. *Key* yang terdapat pada setiap perangkat Arduino dibuat dengan fungsi *generateRandKey()* dimana *key* dibuat dengan melakukan *hashing* dengan algoritma md5 pada nilai acak dan mengambil 15 karakter pertama dan mengubah karakter tersebut menjadi kapital. Fungsi *newKeyDevice()* menggunakan fungsi *generateRandKey()* untuk memperbarui *key* dari perangkat Arduino. Setiap fungsi yang mengubah tabel basis data pada *controller* ini menjalankan fungsi *adminCheckDevice()* terlebih dahulu dimana fungsi tersebut memeriksa *session login* serta memeriksa apakah fungsi tersebut diakses melalui halaman yang dibuat oleh komponen *view* “*devicemanagement*”.

```

1. class AdminUser extends CI_Controller {
2.     public function view() {
3.         if(admin.telahLogin()) {
4.             Tampilkan halaman web pengelolaan pengguna
5.             menggunakan komponen view “usermanagement”.
6.         } else {
7.             redirect(halaman utama);
8.         }
9.     }
10.
11.     public function addUser() {
12.         $this->adminCheckUser();
13.         Masukkan pengguna aplikasi web terbaru ke database
14.         berdasarkan $username, $password, dan $acclvl
15.         menggunakan fungsi pada “User_model”.
16.     }
17.
18.     public function deleteUser() {
19.         $this->adminCheckUser();
20.

```

```

21.     Hapus pengguna aplikasi web berdasarkan $username
22.     menggunakan fungsi pada "User_model".
23.   }
24.
25. }

```

Kode 5. 10 Pseudocode Controller AdminUser

Kode 5.12 adalah *pseudocode* dari *controller* AdminUser yang berfungsi menerima dan merespons *request* pengguna untuk melakukan pengolahan pengguna pada aplikasi web. Fungsi *controller* ini hanya mampu diakses oleh admin atau pengguna dengan akses level 1. Fungsi *view()* pada *controller* ini akan menampilkan halaman web dengan menggunakan komponen *view* *usermanagement*. Data yang ditampilkan pada halaman web tersebut adalah catatan *login* semua pengguna dan daftar pengguna pada aplikasi web. Fungsi dari *addUser()* dan *deleteUser()* bertujuan untuk menambah dan menghapus pengguna pada aplikasi web menggunakan fungsi dari *model* "User_model". Fungsi *adminCheckUser()* akan dijalankan terlebih dahulu ketika pengguna mengakses fungsi *addUser()* atau *deleteUser()* dimana fungsi tersebut memeriksa *session login* dan sumber referensi fungsi tersebut diakses.

```

1.  class Api extends CI_Controller {
2.    public function dhtswitch() {
3.      if (!empty($_POST)) {
4.        Ambil $device_id dari perangkat Arduino berdasarkan
5.        key yang didapat dari $_POST menggunakan fungsi
6.        pada "Device_model".
7.        if (!empty($device_id)) {
8.          Masukkan data suhu dan kelembapan terbaru
9.          perangkat Arduino berdasarkan $device_id
10.         menggunakan fungsi pada "Device_model".
11.
12.         Ambil data relai, threshold, dan update interval
13.         perangkat Arduino menggunakan fungsi pada
14.         "Device_model" berdasarkan $device_id.
15.
16.         Kalkulasi status tiap relai sesuai dengan
17.         keadaan threshold dari perangkat Arduino
18.
19.         Mengirim response dengan format JSON yang berisi

```

```

20.         nama perangkat Arduino, update interval, dan
21.         status setiap relai.
22.
23.     } else {
24.         Response JSON "Device key tidak ditemukan".
25.     }
26. } else {
27.     Response JSON "Parameter tidak sesuai".
28. }
29. }
30. }

```

Kode 5. 11 *Pseudocode Controller Api*

Kode 5.13 adalah *pseudocode controller* Api yang berfungsi untuk menerima dan merespons *request* dari perangkat Arduino. Terdapat satu fungsi utama pada *controller* ini, yaitu *dhtswitch()*. Fungsi *dhtswitch()* adalah fungsi yang diakses oleh perangkat Arduino untuk mengirim data suhu dan kelembapan serta untuk menerima konfigurasi dari aplikasi web. Ketika fungsi *dhtswitch()* diakses oleh perangkat Arduino, maka yang pertama dilakukan adalah memastikan terdapat tiga parameter yang dikirim yaitu *key*, data suhu, dan data kelembapan. Parameter *key* berfungsi untuk melakukan autentikasi dan identifikasi perangkat Arduino sedangkan parameter data suhu dan kelembapan adalah data yang ingin disimpan dari perangkat Arduino tersebut. Jika semua parameter memiliki nilai, maka selanjutnya adalah memastikan bahwa perangkat Arduino terdaftar pada aplikasi web dengan menggunakan *key*. Jika ditemukan nama perangkat Arduino dengan *key* yang sama dengan yang dikirim maka data suhu dan kelembapan akan disimpan sesuai dengan nama perangkat Arduino dari *key* tersebut.

Setelah menyimpan data suhu dan kelembapan maka selanjutnya adalah menyiapkan data yang ingin dikirim pada perangkat Arduino sebagai respons. Data utama yang diperlukan untuk konfigurasi perangkat adalah *delay* tiap pengiriman data suhu dan kelembapan yang disebut *update interval* dan data konfigurasi relai. Untuk data *update interval* didapatkan langsung dengan mengakses fungsi pada model

Device_model. Untuk data konfigurasi relai, diperlukan pengaturan terlebih dahulu agar perangkat Arduino mampu membaca konfigurasi dengan mudah. Data konfigurasi relai yang dikirimkan ke perangkat Arduino hanyalah kode pin dari relai dan status dari relai tersebut apakah tersambung atau tidak. Status dari tiap relai ditentukan dengan mengalkulasi tiap *threshold* dari perangkat Arduino yang telah ditentukan apakah relai harus tersambung atau tidak. Data suhu dan kelembapan yang didapat sebelumnya dibandingkan dengan *threshold* yang telah ditentukan. Jika kondisi terpenuhi maka relai tersebut berstatus “1” yang berarti hidup, jika tidak akan berstatus “0” yang berarti mati. Setelah semua data telah siap maka respons akan dikirim dengan format JSON.

```

1. class Dashboard extends CI_Controller {
2.     public function index() {
3.         sessionCheck();
4.         $data = $this->allDeviceLatestMonDataWithWarning();
5.         Tampilkan halaman utama menggunakan komponen view
6.         "dashboard" dengan bantuan $data.
7.     }
8.     public function allDeviceLatestMonData() {
9.         sessionCheck();
10.        Perbarui data suhu dan kelembapan yang ditampilkan
11.        pada halaman utama dengan bantuan fungsi
12.        allDeviceLatestMonDataWithWarning().
13.    }
14. }

```

Kode 5. 12 PseudoCode Controller Dashboard

Kode 5.14 adalah *pseudocode controller* Dashboard untuk menampilkan halaman utama aplikasi web. Pada *controller* ini terdapat fungsi yaitu `index()` yang bertujuan untuk menampilkan halaman web dengan menggunakan komponen *view* `dashboard` dan juga terdapat fungsi `allDeviceLatestMonData()` untuk memperbarui data yang ditampilkan pada halaman utama. Kedua fungsi tersebut menggunakan fungsi `allDeviceLatestMonDataWithWarning()` untuk mendapatkan data suhu dan kelembapan terbaru serta memeriksa apakah data suhu dan kelembapan tersebut telah

melewati *threshold* yang telah ditentukan yang nantinya juga akan ditampilkan pada halaman utama. Semua pengguna yang telah *login* dapat mengakses *controller* ini diperiksa menggunakan fungsi `sessionCheck()`. Data yang ditampilkan pada halaman utama adalah daftar perangkat Arduino yang terdaftar serta data suhu dan kelembapan terbaru yang didapat dari tiap perangkat Arduino tersebut.

```

1.  class Device extends CI_Controller {
2.      public function view($device_id) {
3.          sessionCheck();
4.          Tampilkan halaman informasi suhu dan kelembapan dari
5.          perangkat Arduino yang dipilih menggunakan $device_id
6.          menggunakan komponen view "device".
7.      }
8.
9.      public function downloadCSV($device_id) {
10.         $this->adminCheckDevice($device_id);
11.         Unduh data suhu dan kelembapan pada perangkat
12.         Arduino yang dipilih dengan format CSV.
13.     }
14.
15.     public function deleteMonitoringDataDevice($device_id) {
16.         $this->adminCheckDevice($device_id);
17.         Hapus seluruh data suhu dan kelembapan pada
18.         perangkat Arduino yang dipilih menggunakan $device_id.
19.     }
20.
21.     public function calibrateMonitoringDeviceData() {
22.         $this->adminCheckDevice($device_id);
23.         Kalibrasi suhu dan kelembapan yang didapat dari
24.         perangkat Arduino berdasarkan device_id.
25.     }
26.
27.     public function deviceLatestMonitoringData() {
28.         sessionCheck();
29.         Perbarui data suhu dan kelembapan yang ditampilkan
30.         pada halaman web.
31.     }
32.
33.     public function deviceDateRangeMonitoringData() {
34.         sessionCheck();
35.         Tampilkan informasi suhu dan kelembapan perangkat
36.         Arduino pada halaman dengan rentang waktu tertentu.

```

```

37.     }
38.
39.     public function deviceInsightMonitoringData() {
40.         sessionCheck();
41.         Perbarui Insight seperti rata-rata, maksimum, dan
42.         minimum suhu dan kelembapan perangkat Arduino yang
43.         ditampilkan pada halaman web.
44.     }
45. }

```

Kode 5. 13 *Pseudocode Controller Device*

Kode 5.15 adalah *pseudocode controller Device* yang berfungsi menerima dan merespons *request* pengguna untuk mengelola data suhu dan kelembapan dari perangkat Arduino yang dipilih oleh pengguna. Pada fungsi *view()* halaman web akan ditampilkan data suhu dan kelembapan dengan bantuan komponen *view device*. Terdapat beberapa fungsi lain yang dapat diakses pada *controller* ini yaitu melakukan kalibrasi data suhu dan kelembapan dengan fungsi *calibrateMonitoringDevice Data()*, mengunduh data suhu dan kelembapan dengan format CSV dengan fungsi *downloadCSV()*, serta menghapus data suhu dan kelembapan dengan fungsi *deleteMonitoringData Device()*. Untuk fungsi *view()* dapat diakses oleh semua pengguna yang telah *login* sedangkan tiga fungsi yang sebelumnya sudah dijelaskan hanya dapat diakses oleh admin atau pengguna dengan akses level 1. Halaman web yang ditampilkan oleh *view()* memerlukan bantuan fungsi *deviceLatestMonitoringData()* dan *deviceInsightMonitoringData()* untuk menampilkan data terbaru secara otomatis yang didapat dari perangkat Arduino. Fungsi *deviceDateRangeMonitoringData()* digunakan oleh pengguna jika ingin menampilkan data suhu dan kelembapan pada rentang jangka waktu tertentu. Fungsi *deviceLatestMonitoringDataWithWarning()* digunakan pada fungsi *view()* dan *deviceInsightMonitoringData()* untuk menampilkan data suhu dan kelembapan terbaru dan menampilkan pemberitahuan jika data terbaru melebihi *threshold* yang telah ditentukan.

```

1. class DeviceThreshold extends CI_Controller {
2.     public function view($device_id) {
3.         if(admin.telahLogin()) {
4.             Tampilkan halaman web pengelolaan threshold dan
5.             relai perangkat Arduino yang dipilih dengan
6.             $device_id menggunakan komponen view
7.             "devicethreshold".
8.         } else {
9.             Tampilkan halaman 404 not found.
10.        }
11.    }
12.
13.    public function newDeviceThreshold() {
14.        $this->adminCheckDeviceThreshold($device_id);
15.        Masukkan info threshold perangkat Arduino terbaru
16.        berdasarkan $device_id ke database menggunakan fungsi
17.        pada "Device_model".
18.    }
19.
20.    public function editDeviceThreshold() {
21.        $this->adminCheckDeviceThreshold($device_id);
22.        Ubah info threshold perangkat Arduino pada database
23.        berdasarkan $threshold_id menggunakan fungsi pada
24.        "Device_model".
25.    }
26.
27.    public function deleteDeviceThreshold() {
28.        $this->adminCheckDeviceThreshold($device_id);
29.        Hapus threshold pada perangkat Arduino berdasarkan
30.        $threshold_id menggunakan fungsi pada "Device_model".
31.    }
32.
33.    public function newSwitchDevice() {
34.        $this->adminCheckDeviceThreshold($device_id);
35.        Masukkan info relai terbaru yang terhubung pada
36.        perangkat Arduino berdasarkan $device_id menggunakan
37.        fungsi pada "Device_model".
38.    }
39.
40.    public function editSwitchDevice() {
41.        $this->adminCheckDeviceThreshold($device_id);
42.        Ubah info relai yang terhubung pada perangkat Arduino
43.        berdasarkan $switch_id menggunakan fungsi pada
44.        "Device_model".
45.    }

```

```

46.
47. public function deleteSwitchDevice() {
48.     $this->adminCheckDeviceThreshold($device_id);
49.     Hapus info relai yang terhubung dengan perangkat
50.     Arduino berdasarkan $switch_id menggunakan fungsi
51.     pada "Device_model".
52. }
53.
54. public function enableSwitchThresholdCheck() {
55.     $this->adminCheckDeviceThreshold($device_id);
56.     Mengaktifkan atau menonaktifkan pemeriksaan threshold
57.     pada relai yang terhubung dengan perangkat Arduino.
58. }
59. }

```

Kode 5. 14 *Pseudocode Controller DeviceThreshold*

Kode 5.16 adalah *psseudocode* dari *controller* DeviceThreshold yang berfungsi merespons *request* yang berhubungan dengan pengelolaan *threshold* dan relai pada perangkat Arduino. Fungsi *view()* pada *controller* akan menampilkan halaman web yang berisi daftar *threshold* serta relai yang terdaftar pada perangkat Arduino yang dipilih. Pengguna dapat menambah, mengubah, atau menghapus *threshold* dengan menggunakan fungsi *newDeviceThreshold()*, *editDeviceThreshold()*, dan *deleteDeviceThreshold()*. Sedangkan untuk relai dapat menggunakan fungsi *newSwitchDevice()*, *editSwitchDevice()*, dan *deleteSwitch Device()*. Dari tiap relai yang ada, pemeriksaan *threshold* dapat dilakukan untuk menentukan kapan relai tersambung atau tidak. Pemeriksaan *threshold* dari tiap relai dapat diaktifkan atau dinonaktifkan dengan menggunakan fungsi *enableSwitch ThresholdCheck()*. Semua fungsi yang terdapat pada *controller* ini hanya dapat diakses oleh admin atau pengguna dengan akses level 1.

```

1. class UserSession extends CI_Controller {
2.
3.     public function login() {
4.         if(sudah login) {
5.             redirect(halaman utama);
6.         } else {
7.             Tampilkan halaman login menggunakan komponen view

```

```

8.         "login".
9.     }
10. }
11.
12. public function logout() {
13.     Hapus session pengguna.
14.     redirect(halaman login);
15. }
16.
17. public function validate() {
18.     if($uname == username pada database && $passhash ==
19.         hash password pada $uname) {
20.
21.         Buat session pengguna berdasarkan user yang login.
22.         redirect(halaman utama);
23.     } else {
24.         redirect(halaman login);
25.     }
26. }
27. }

```

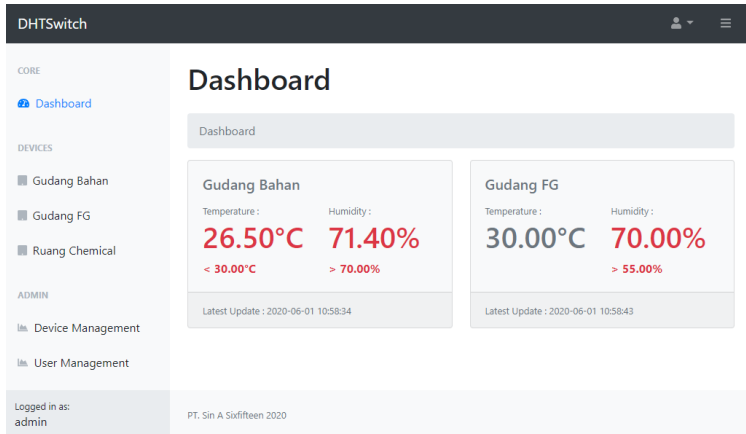
Kode 5.15 *Pseudocode Controller* UserSession

Kode 5.17 adalah *pseudocode controller* UserSession yang berfungsi mengatur autentikasi dan otorisasi pengguna yang mengakses aplikasi web. Pengguna aplikasi web akan diarahkan pada fungsi login() jika pengguna belum melakukan login. Pada fungsi login(), aplikasi web akan menampilkan halaman login dengan bantuan komponen view login. Fungsi validate() akan dijalankan ketika pengguna mengirim form username dan password yang diisi pada halaman login. Fungsi tersebut pertama-tama akan memastikan apakah fungsi diakses dari halaman login aplikasi web. Setelah itu username dan password akan dicocokkan dengan yang ada pada basis data. Jika username dan password cocok, maka session akan dibuat yang berisi username dan level akun tersebut dan kemudian akan dicatat waktu login pengguna. Terdapat fungsi lain pada controller ini yaitu fungsi logout() dimana session akan dihapus dan pengguna akan diarahkan ke halaman login serta fungsi changepass() untuk mengubah password dari akun pengguna.

5.3.3.3 Implementasi View

View adalah komponen yang akan dilihat oleh pengguna pada aplikasi web. Setiap *view* berpasangan dengan *controller* untuk memudahkan pengguna dalam mengakses fungsi pada *controller* tersebut terkecuali *controller* Api.

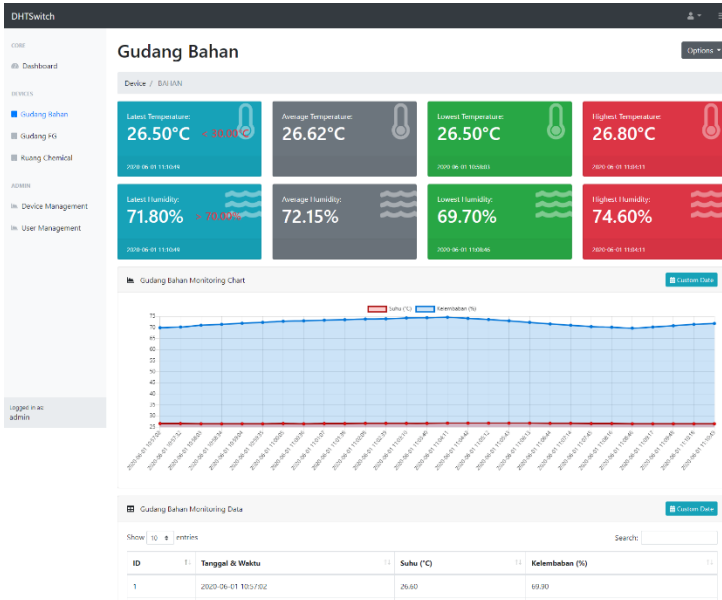
Komponen *view* dashboard berisi kode untuk menampilkan halaman utama aplikasi web kepada pengguna. Pada halaman ini terdapat info suhu dan kelembapan terbaru dari tiap perangkat Arduino yang ditampilkan dalam bentuk *card*. Jumlah *card* yang ditampilkan menyesuaikan dengan jumlah perangkat Arduino yang memiliki data suhu dan kelembapan. Tampilan halaman utama web dapat dilihat pada Gambar 5.3.



Gambar 5.3 User Interface View dashboard

Komponen *view* device berfungsi sebagai tampilan pengguna untuk menampilkan informasi suhu dan kelembapan pada perangkat Arduino yang dipilih. Informasi pada halaman web dibagi menjadi tiga bagian. Bagian pertama menampilkan informasi singkat dari data suhu dan kelembapan seperti data terakhir, data maksimal, data minimal, dan data rata-rata dari suhu dan kelembapan. Bagian kedua adalah grafik suhu dan kelembapan terhadap waktu. Sedangkan bagian ketiga adalah tabel suhu dan kelembapan secara detail. Grafik dan tabel dapat

diatur rentang tanggal yang ingin ditampilkan sesuai pengguna. Selain menampilkan data suhu dan kelembapan, terdapat fungsi lain yang dapat digunakan oleh pengguna pada halaman web ini. Tombol “options” akan menampilkan fungsi seperti melakukan kalibrasi data, mengunduh data, menghapus data, serta mengarahkan ke pengaturan *threshold*. Tampilan halaman web komponen *view device* dapat dilihat pada Gambar 5.4.



Gambar 5.4 User Interface View device

Komponen *view* devicemanagement berisi kode untuk menampilkan tampilan pengguna untuk mengelola perangkat Arduino. Daftar dari perangkat Arduino yang terdaftar akan ditampilkan dengan detail informasi tiap perangkat. Semua fungsi yang dapat dilakukan pada halaman web ini seperti menambah, mengubah, atau menghapus perangkat Arduino, memperbarui *key*, dan mengubah *update interval* perangkat Arduino di respons oleh fungsi yang terdapat pada controller AdminDevice. Tampilan halaman web komponen *view devicemanagement* dapat dilihat pada Gambar 5.5

The screenshot shows the 'Device Management' page in the DHTSwitch application. The left sidebar contains a navigation menu with categories: CORE (Dashboard), DEVICES (Gudang Bahan, Gudang FG, Ruang Chemical), and ADMIN (Device Management, User Management). The main content area is titled 'Device Management' and shows a breadcrumb 'Admin / Device'. Below this is a 'Device List' section containing a table with the following data:

Device ID	Device Name	Board Name	Key	Update Interval (ms)	Action
BAHAN	Gudang Bahan	Wemos D1 Mini	9A6CA0AA4454BC3A	30000	New Key Edit Delete Device
CHEM	Ruang Chemical	Wemos D1 Mini	ADD78FF9D5C5D57	30000	New Key Edit Delete Device
FG	Gudang FG	Wemos D1 Mini	ZCEE12E86A84423	30000	New Key Edit Delete Device

Below the table is a green 'New Device' button. At the bottom of the page, it shows 'Logged in as: admin' and 'PT. Sin A Sulfitan 2020'.

Gambar 5.5 User Interface View devicemanagement

Komponen *view* deviceThreshold berisi kode yang berfungsi sebagai tampilan pengguna untuk melakukan konfigurasi *threshold* dan relai pada perangkat Arduino. Daftar *threshold* dan relai perangkat Arduino yang didaftarkan pada aplikasi web akan ditampilkan pada *view* ini. Pengguna dapat menambah, mengubah, dan menghapus *threshold* dan relai yang didaftarkan. Dari tiap relai, pengguna dapat mengaktifkan *threshold* yang perlu dipenuhi agar relai tersambung. Ketika tombol “Threshold Check” diklik maka tabel yang berisi daftar *threshold* akan muncul. Pengguna dapat mengaktifkan *threshold* mana saja yang perlu dipenuhi agar relai menjadi aktif. Gambar 5.6 adalah tampilan dari komponen *view* deviceThreshold.

The screenshot shows the 'Gudang Bahan Threshold Management' page in the DHTSwitch application. The page has a dark header with the application name and user profile. A sidebar on the left contains navigation links under 'CORE', 'DEVICES', and 'ADMIN'. The main content area is titled 'Gudang Bahan Threshold Management' and shows a breadcrumb trail 'Device / BAHAN / Threshold'. Below this, there's a sub-section 'Gudang Bahan Threshold' containing a table with the following data:

ID	Threshold Check	Description	Options
1	Suhu < 30.00		Edit Delete
2	Suhu > 35.00		Edit Delete
3	Kelembapan < 50.00		Edit Delete
4	Kelembapan > 70.00		Edit Delete

Below the table is a green 'New Threshold' button. Underneath, there's another section 'Gudang Bahan Switch' with a table:

ID	Switch Name	Switch Pin	Options
1	Kipas	D1	Threshold Check Edit Delete

The bottom left corner shows 'Logged in as: admin'.

Gambar 5.6 User Interface View deviceThreshold

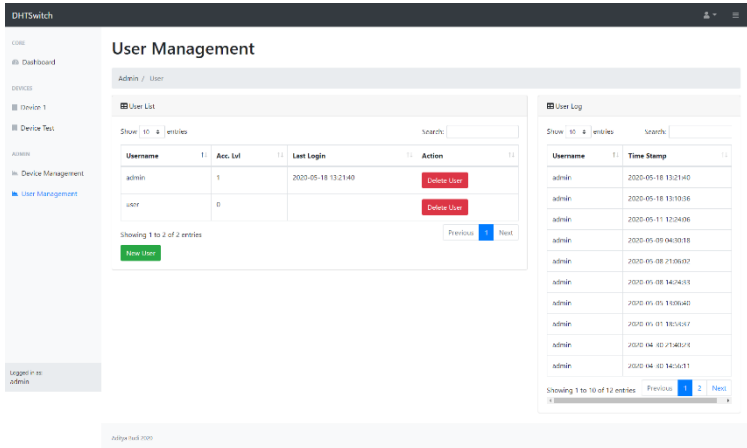
The screenshot shows the login page of the DHTSwitch application. It features a central white login form on a dark gray background. The form has the title 'DHTSwitch' and contains the following fields:

- Username: Enter username
- Password: Enter password
- A blue 'Login' button.

The footer of the page shows the text 'July 16, 2020'.

Gambar 5.7 User Interface View login

Komponen *view* login adalah tampilan pengguna untuk menampilkan halaman *login* dari aplikasi web. Pada halaman web ini pengguna dapat mengisi *username* dan *password* untuk melakukan autentikasi pada aplikasi web. Tampilan pengguna ini hanya akan ditampilkan jika pengguna belum melakukan autentikasi pada aplikasi web. Gambar 5.7 adalah tampilan pengguna dari komponen *view* login.



Gambar 5. 8 *User Interface View* usermanagement

Komponen *view* usermanagement berfungsi sebagai tampilan pengguna untuk melakukan pengelolaan pengguna pada aplikasi web. Pada tampilan komponen *view* ini terdapat tabel daftar pengguna yang dapat mengakses aplikasi web dan tabel catatan waktu *login* dari seluruh pengguna. Fungsi yang dapat digunakan oleh pengguna pada tampilan adalah fungsi untuk menambah dan menghapus pengguna. Semua fungsi yang terdapat pada tampilan ini direspons oleh fungsi pada *controller* AdminUser. Gambar 5.8 adalah tampilan pengguna dari komponen *view* usermanagement.

5.3.3.4 Implementasi Routing

Routing digunakan pada aplikasi web untuk melakukan kustomisasi URL dalam mengakses fungsi pada *controller*. *Routing* diimplementasikan sesuai dengan rancangan pada tahap sebelumnya.

```

1. $route['default_controller'] = 'Dashboard';
2. $route['404_override'] = '';
3. $route['translate_uri_dashes'] = FALSE;
4.
5. $route['login'] = 'UserSession/login';
6. $route['logout'] = 'UserSession/logout';
7.

```

```

8. $route['device/(:any)'] = 'Device/view/$1';
9. $route['device/(:any)/threshold'] =
   'DeviceThreshold/view/$1';
10.
11. $route['admin/device'] = 'AdminDevice/view';
12. $route['admin/user'] = 'AdminUser/view';

```

Kode 5. 16 Implementasi *Routing* pada Aplikasi Web

Kode 5.18 adalah implementasi dari *routing* pada aplikasi web. *Controller* Dashboard digunakan menjadi *controller default* aplikasi web sehingga ketika pengguna mengakses URL aplikasi web tanpa segmen maka *controller* tersebut yang akan merespons *request* dari pengguna. Selain itu *routing* yang diimplementasikan adalah *routing* terkait autentikasi pengguna. Untuk melakukan *login* atau *logout*, pengguna tidak memerlukan untuk menuliskan *controller* *UserSession* terlebih dahulu pada URL sehingga langsung pada menulis alamat aplikasi web dengan tambahan segmen *login* atau *logout*. Fungsi *view()* pada *controller* *Device* dan *DeviceThreshold* diarahkan dengan menggunakan *routing* dimana segmen pertama adalah “device”, segmen kedua adalah ID dari perangkat Arduino yang ingin diakses, dan segmen ketiga adalah “threshold” jika ingin mengakses konfigurasi relai. Untuk *controller* yang berfungsi untuk mengelola perangkat Arduino dan pengguna aplikasi web digunakan *routing* dimana segmen pertama adalah “admin” karena hanya pengguna dengan akses level “1” atau admin yang dapat mengakses *controller* tersebut.

5.4 *Testing* Sistem

Sistem akan diujicoba dengan metode *black box testing* berdasarkan desain *testing* yang telah dibuat pada tahap sebelumnya. *Testing* akan terdiri dari beberapa *test case* yang akan diuji pada sistem.

5.4.1 *Testing* Perangkat Keras

Testing pada perangkat keras mengacu pada *test case* yang telah dibuat pada tahap sebelumnya. Setiap *test case* memiliki kriteria ekspektasi yang harus dipenuhi.

Tabel 5. 4 Hasil *Test Case* Perangkat Keras

Test Case	Ekspektasi	Hasil
Terhubung dengan jaringan Wi-Fi	Perangkat Arduino terkoneksi dengan jaringan Wi-Fi yang telah ditentukan	OK
Pembacaan sensor DHT22	Data suhu dan kelembapan dapat terbaca oleh perangkat Arduino dari sensor DHT22 dan ditampilkan pada Serial.	OK
Mengirim data ke aplikasi web dengan protokol HTTP	HTTP <i>respons code</i> 200 dan mendapatkan data respons dari aplikasi web	OK
Mengirim data ke aplikasi web dengan protokol HTTPS	HTTP <i>respons code</i> 200 dan mendapatkan data respons dari aplikasi web	OK
Memperbarui konfigurasi relai dan <i>delay</i>	Konfigurasi relai dan <i>delay</i> menyesuaikan konfigurasi pada aplikasi web	OK

Tabel 5.4 menjelaskan hasil *test case* perangkat Arduino. Semua *test case* yang diuji coba memenuhi kriteria ekspektasi. Perangkat Arduino dapat terhubung dengan jaringan Wi-Fi dibuktikan dengan mendapatkan *IP Address* yang ditunjukkan pada Serial Monitor pada IDE Arduino. Data suhu dan kelembapan berhasil didapatkan dan ditunjukkan pada Serial Monitor. Pengiriman data menggunakan protokol HTTP maupun HTTPS dapat dilakukan dan mendapat *respons code* 200. Pengiriman data dilakukan dengan mempersiapkan terlebih dahulu perangkat lunak dan mengambil *fingerpint* untuk protokol HTTPS pada domain web perangkat lunak. Pengaturan relai dan *delay* mampu dilakukan berdasarkan data yang didapat dari respons web ketika mengirim data suhu dan kelembapan. Nilai *delay* dimasukkan pada variabel `update_interval` dan relai dikonfigurasi satu-persatu.

Tabel 5. 5 Pengujian Jarak Akurasi Sensor DHT22

Jarak	Kondisi Lingkungan	Perangkat 1		Perangkat 2	
		Suhu (°C)	RH (%)	Suhu (°C)	RH (%)
4,4 m	Non-AC, ruangan tertutup	27,6	81.8	28,1	81,3
4,4 m	AC, ruangan tertutup	25,4	55,3	25,6	57,4
12 m	Luar ruangan, sore hari	29	81.3	29	80.1
30 m	Luar ruangan, sore hari	29,7	77,69	29,6	79,6

Tabel 5.5 adalah hasil pengujian jarak akurasi untuk sensor DHT22. Pengujian dilakukan pada beberapa jarak dan kondisi lingkungan. Uji akurasi dilakukan dengan menggunakan dua perangkat Arduino yang diletakkan pada kedua ujung setiap jarak dan dicatat nilai suhu dan kelembapannya. Sebelum dilakukan pengujian, perangkat Arduino 2 dikalibrasi terlebih dahulu mengikuti nilai perangkat Arduino 1 dengan asumsi nilai perangkat Arduino 1 akurat. Berdasarkan data yang terdapat pada Tabel 5.5, sensor DHT22 dapat menjaga akurasi sesuai dengan spesifikasi yaitu untuk suhu memiliki akurasi $\pm 0,5^{\circ}\text{C}$ dan untuk kelembapan memiliki akurasi $\pm 2\%$.

Tabel 5. 6 Pengujian Pin Wemos D1 Mini untuk Relai

Kode Pin	Hasil Pengujian
D0	Normal
D1	Normal
D2	Normal
D3	Aktif sesaat pada saat <i>booting</i>
D4	Aktif sesaat pada saat <i>booting</i>
D5	Normal
D6	Normal
D8	Tidak dapat digunakan

Tabel 5.6 adalah hasil pengujian penggunaan pin digital pada Wemos D1 Mini untuk keperluan mengatur relai. Setiap pin diuji coba untuk mengatur relai dan diperiksa apakah terdapat kelemahan ketika menggunakan pin tersebut. Hasil dari Tabel 5.6 menunjukkan pin D0, D1, D2, D5, D6 dapat digunakan secara normal. Untuk pin D3, D4 pin akan menyala sesaat dikarenakan proses *booting* dari Wemos D1 Mini namun akan normal setelah proses tersebut. Sedangkan untuk pin D8 didapatkan bahwa pin tersebut tidak dapat digunakan karena relai selalu dalam keadaan tersambung ketika menggunakan pin tersebut terlepas dari status yang diberikan.

5.4.2 Testing Perangkat Lunak

Testing pada perangkat keras mengacu pada *test case* yang telah telah dibuat pada tahap sebelumnya. Setiap *test case* memiliki kriteria ekspektasi yang harus dipenuhi.

Tabel 5. 7 Hasil *Test Case* Perangkat Lunak

<i>Test Case</i>	Skenario	Ekspektasi	Hasil
<i>Login</i>	Pengguna mengetik <i>username</i> dan <i>password</i> dengan benar	Pengguna masuk ke halaman utama	OK
	Pengguna mengetik <i>username</i> dan atau <i>password</i> yang salah	Pengguna mendapat <i>warning username</i> atau <i>password</i> salah	OK
Menerima data dari perangkat Arduino	Parameter dan <i>key</i> sesuai	Data suhu dan kelembapan disimpan pada basis data dan memberi respons	OK

		konfigurasi perangkat.	
	Parameter dan atau <i>key</i> tidak sesuai	Memberi respons kesalahan parameter.	OK
Melihat <i>dashboard</i>	Pengguna yang sudah <i>login</i> mengakses halaman utama	Halaman utama ditampilkan	OK
Melihat data suhu dan kelembapan	Pengguna yang sudah <i>login</i> memilih perangkat Arduino pada <i>navigation bar</i>	Data suhu dan kelembapan dalam bentuk rangkuman, grafik, dan tabel ditampilkan	OK
Kalibrasi data suhu dan kelembapan dari perangkat Arduino	Pengguna level admin mengisi <i>form</i> kalibrasi suhu dan kelembapan perangkat Arduino	Data kalibrasi tersimpan pada basis data	OK
Menghapus data suhu dan kelembapan	Pengguna level admin menekan tombol hapus pada halaman data suhu dan kelembapan perangkat Arduino	Data suhu dan kelembapan pada perangkat Arduino yang dipilih terhapus dari basis data	OK
Mengunduh data suhu dan	Pengguna menekan tombol	Data suhu dan kelembapan	OK

kelembapan perangkat Arduino	unduh pada halaman data suhu dan kelembapan perangkat Arduino	dalam format CSV diunduh ke komputer pengguna.	
Mengelola relai perangkat Arduino	Pengguna level admin menambah, mengubah, dan menghapus relai pada perangkat Arduino	Konfigurasi relai yang dilakukan pengguna tersimpan di basis data	OK
Mengelola <i>threshold</i> perangkat Arduino	Pengguna level admin menambah, mengubah, dan menghapus <i>threshold</i> perangkat Arduino	Konfigurasi <i>threshold</i> tersimpan di basis data	OK
Mengelola daftar perangkat Arduino yang terdaftar	Pengguna level admin menambah, mengubah, dan menghapus perangkat Arduino dan konfigurasinya	Perangkat Arduino dan konfigurasinya tersimpan pada basis data.	OK
Mengelola daftar pengguna pada aplikasi web	Pengguna level admin menambah dan menghapus pengguna pada aplikasi web	Konfigurasi pengguna tersimpan pada basis data	OK

Tabel 5.7 menjelaskan hasil *test case* perangkat Arduino. Setiap *test case* diuji coba menggunakan skenario yang telah ditetapkan dan diperiksa apakah hasil dari skenario tersebut telah memenuhi ekspektasi yang ditetapkan. Hasil *testing*

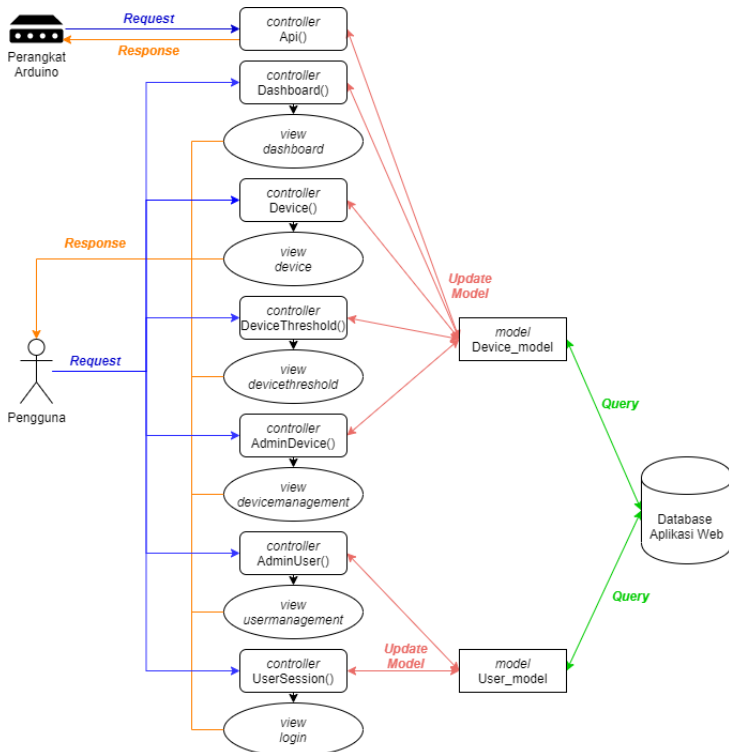
menunjukkan bahwa semua *test case* yang dijalankan memenuhi ekspektasi.

BAB VI HASIL DAN PEMBAHASAN

Pada bagian ini akan menjelaskan mengenai hasil dan pembahasan dari pengembangan sistem yang telah dibuat pada pengerjaan tugas akhir ini.

6.1 Arsitektur Sistem

Arsitektur dari sistem yang dikembangkan disesuaikan dengan kebutuhan yang telah disebutkan pada tahap perancangan. Arsitektur dikembangkan dengan menggunakan *design pattern* Model-View-Controller. *Design pattern* tersebut digunakan sesuai dengan *framework* yang digunakan pada aplikasi web yaitu CodeIgniter.

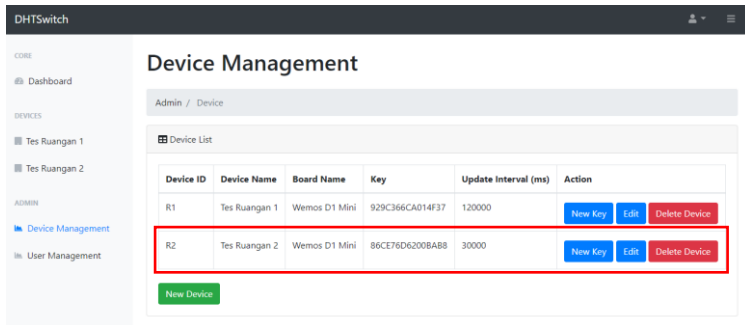


Gambar 6. 1 Arsitektur Sistem

Gambar 6.1 adalah ilustrasi dari arsitektur sistem yang dikembangkan. Secara keseluruhan terdapat 8 *controller* dengan fungsinya masing-masing. Setiap *controller* berfungsi untuk menerima request, mengolah, dan mengirimkan *response* dari *request*. Untuk *controller* yang menerima *request* dari pengguna, *controoler* tersebut memberikan *response* dengan bantuan *view* untuk memudahkan pengguna dalam mengakses aplikasi web. Sedangkan untuk *controller* Api() yang menerima *request* dari perangkat Arduino tidak memerlukan komponen *view* karena perangkat Arduino cukup menerima *response* dalam bentuk JSON. Setiap *controller* terhubung dengan *model* yang bertugas mengambil, mengubah, menghapus, atau memperbarui data yang ada pada database dengan cara melakukan *query*.

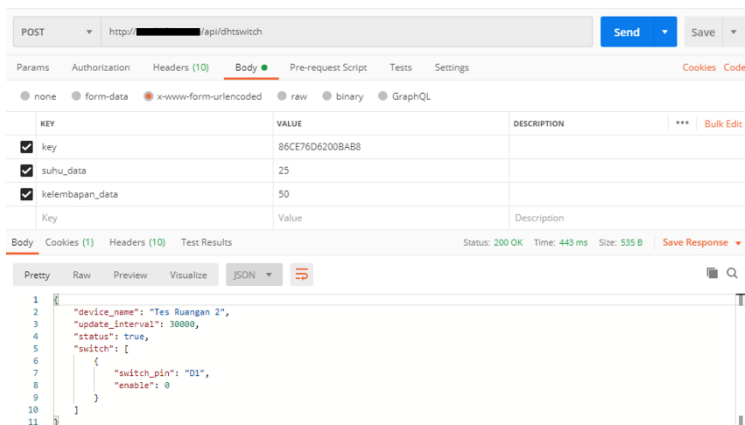
6.2 Komunikasi Perangkat Arduino

Perangkat Arduino berkomunikasi dengan aplikasi web menggunakan *controller* Api yang terdapat pada aplikasi web. Perangkat Arduino melakukan *request* pada *method* dhtswitch() pada *controller* dengan mengirimkan parameter *key*, suhu, dan kelembapan dengan menggunakan protokol HTTP. Respons dari *request* tersebut adalah sebuah paket data JSON yang berisi konfigurasi dari perangkat lunak seperti *update interval* dan konfigurasi relai. Perangkat Arduino membaca paket data JSON tersebut untuk diimplementasikan. Perangkat Arduino harus didaftarkan terlebih dahulu pada aplikasi web untuk dapat menyimpan data suhu dan kelembapan yang dikirim. *Key* dari perangkat Arduino yang didaftarkan berperan penting sebagai autentikasi perangkat. Gambar 6.2 adalah daftar perangkat Arduino yang telah didaftarkan oleh pengguna pada aplikasi web.

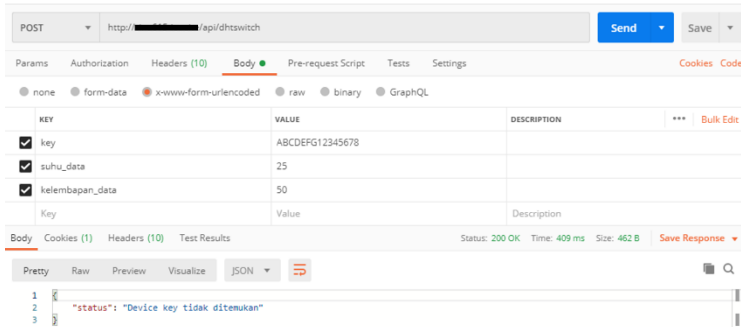


Gambar 6. 2 Daftar Perangkat Arduino yang Terdaftar

Pengujian apakah *method* `dhtswitch()` pada *controller* Api dapat bekerja sesuai dengan fungsinya dapat dilakukan dengan menggunakan *tools* Postman. Gambar 6.3 adalah hasil pengujian ketika mengirimkan *request* dengan parameter *key* yang telah terdaftar sedangkan Gambar 6.4 adalah hasil pengujian dengan parameter *key* yang belum terdaftar.



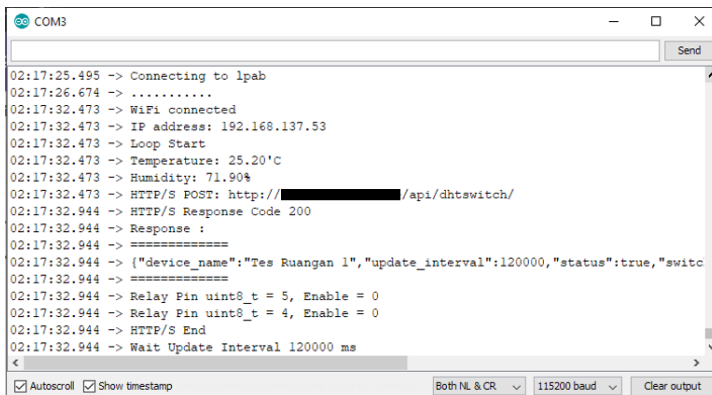
Gambar 6. 3 Update Data dengan Key Terdaftar



Gambar 6.4 Update Data dengan Key Tidak Terdaftar

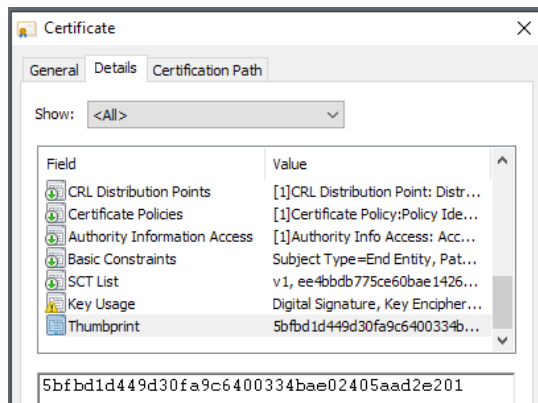
6.3 Program Perangkat Arduino

Program dari perangkat Arduino dikembangkan untuk mengirimkan data suhu dan kelembapan yang didapat dari sensor DHT22 ke aplikasi web dan mengatur relai sesuai dengan konfigurasi yang didapat dari aplikasi web. Pengiriman data ke aplikasi web dan melakukan konfigurasi dilakukan dalam satu kali fungsi loop() dimana perangkat Arduino mengirim POST request dan menerima respon konfigurasi hasil request tersebut sehingga konfigurasi pada perangkat Arduino tidak bisa diubah kecuali perangkat Arduino itu sendiri yang melakukan request pada aplikasi web. Gambar 6.5 adalah *Serial Monitor* dari program yang dijalankan pada perangkat Arduino dimulai dari terhubung dengan Wi-Fi hingga relai terkonfigurasi.



Gambar 6.5 Serial Monitor Perangkat Arduino

Perangkat Arduino dapat mengirimkan data suhu dan kelembapan kepada aplikasi web dengan menggunakan protokol HTTPS. Untuk dapat melakukan hal tersebut, perangkat Arduino harus mengetahui terlebih dahulu *fingerprint* atau juga biasa disebut *Thumbprint* dari protokol HTTPS pada aplikasi web. *Fingerprint* HTTPS memiliki panjang 40 karakter yang dapat didapatkan dengan cara melihat sertifikat HTTPS dari alamat web. Gambar 6.6 adalah contoh dari *fingerprint* domain web aplikasi web. *Fingerprint* setiap domain web berbeda-beda dan memiliki jangka waktu yang mana *fingerprint* tersebut akan berubah ketika jangka waktu tersebut habis.



Gambar 6.6 *Fingerprint* Domain Web

Setelah mendapatkan *fingerprint* dari domain web tempat aplikasi web berada, maka *fingerprint* tersebut perlu dimasukkan pada program perangkat Arduino agar mampu mengirim data melalui protokol HTTPS. Kode 6.1 adalah potongan program dari perangkat Arduino yang mana terdapat variabel *fingerprint* yang harus diisi sesuai dengan *fingerprint* yang didapatkan dari sertifikat domain aplikasi web. Perlu diingat bahwa Program perangkat Arduino perlu diperbarui dengan *fingerprint* yang terbaru ketika sertifikat HTTPS pada domain aplikasi web diperbarui.

```

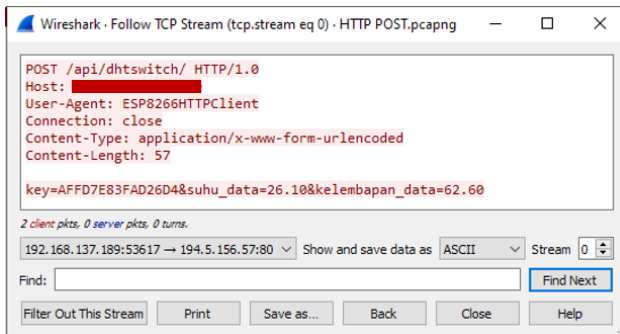
1. //Host Config=====
2. const bool https = true;
3. const char* url =
4. "https://[host aplikasi web]/api/dhtswitch/";
5. //Fingerprint HTTPS
6. const uint8_t fingerprint[20] = {0x5b, 0xfb, 0xd1, 0xd4,
   0x49, 0xd3, 0x0f, 0xa9, 0xc6, 0x40, 0x03, 0x34, 0xba,
   0xe0, 0x24, 0x05, 0xaa, 0xd2, 0xe2, 0x01};
7. //=====

```

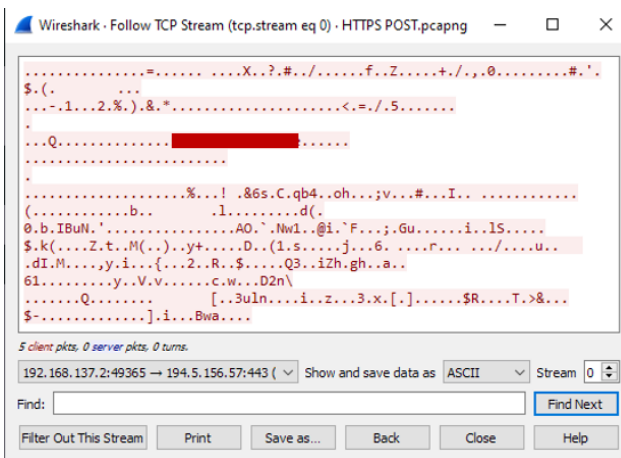
Kode 6. 1 Pengaturan *Host* Program Perangkat Arduino

Penggunaan protokol HTTPS untuk melakukan transmisi data antara perangkat Arduino dengan aplikasi web dilakukan dengan tujuan untuk mengaman data ketika proses transmisi karena dengan menggunakan protokol HTTPS maka data yang dikirim atau diterima terenkripsi. *Key* perangkat Arduino adalah data utama yang perlu dilindungi ketika melakukan transmisi data. Jika *key* tersebut diketahui oleh orang yang tidak bertanggung jawab, maka ada kemungkinan data suhu dan kelembapan yang disimpan pada aplikasi web bukan berasal dari perangkat Arduino. Untuk mengetahui apakah penggunaan protokol HTTPS telah berhasil untuk mengamankan transmisi data maka dilakukan percobaan *packet sniffing* atau membaca paket yang melintas pada suatu jaringan. *Packet sniffing* dilakukan dengan menggunakan *tools* WireShark pada jaringan Wi-Fi yang terhubung dengan perangkat Arduino.

Gambar 6.7 adalah hasil dari *packet sniffing* pada perangkat Arduino yang menggunakan protokol HTTP. Konten dari pengiriman menggunakan *method* POST dapat terlihat secara jelas. Sedangkan pada Gambar 6.8 adalah hasil dari *packet sniffing* pada perangkat Arduino yang menggunakan protokol HTTPS untuk melakukan transmisi data dengan aplikasi web. Konten pada transmisi tersebut berisi karakter acak yang merupakan hasil dari enkripsi menggunakan protokol HTTPS. Hal tersebut membuat transmisi menggunakan protokol HTTPS lebih aman dari menggunakan protokol HTTP karena tidak sembarang orang dapat membaca transmisi data ketika menggunakan protokol HTTPS.



Gambar 6. 7 Pengiriman Data Protokol HTTP



Gambar 6. 8 Pengiriman Data Protokol HTTPS

Halaman ini sengaja dikosongkan

BAB VII

KESIMPULAN DAN SARAN

Pada bagian ini akan menjelaskan mengenai kesimpulan dari pengerjaan tugas akhir ini dan saran untuk pengembangan sistem selanjutnya yang relevan.

7.1 Kesimpulan

Berikut adalah kesimpulan dari pengembangan sistem yang telah dilakukan pada pengerjaan tugas akhir ini.

1. Perangkat Arduino yang dikembangkan dapat mengetahui informasi suhu dan kelembapan ruangan dengan menggunakan sensor.
2. Perangkat Arduino mampu mengirim dan menerima informasi dari aplikasi web dengan menggunakan protokol HTTP.
3. Perangkat Arduino dapat menerapkan protokol HTTPS untuk melakukan komunikasi dengan aplikasi web dengan aman.
4. Sensor DHT22 mampu melaporkan data suhu dan kelembapan dengan akurat setelah dilakukan kalibrasi terlebih dahulu.
5. Tidak semua pin digital yang terdapat pada *board* Wemos D1 Mini dapat digunakan untuk mengatur relai.
6. Aplikasi web memudahkan pengguna dalam mengakses informasi yang dikirimkan oleh perangkat Arduino.
7. Aplikasi web memudahkan pengguna dalam mengatur konfigurasi perangkat Arduino.
8. Aplikasi web dapat menerima informasi yang dikirimkan oleh beberapa perangkat Arduino secara bersamaan.
9. Sistem yang dibuat telah memenuhi kebutuhan dari studi kasus.

7.2 Saran

Berikut adalah beberapa saran untuk pengembangan lebih lanjut untuk menyempurnakan pengembangan sistem.

1. Menggunakan WebSocket untuk melakukan komunikasi antara perangkat Arduino dengan aplikasi serta untuk menampilkan data pada aplikasi web.
2. Pengembangan aplikasi *mobile* untuk menampilkan info suhu dan kelembapan pada perangkat genggam.

DAFTAR PUSTAKA

- [1] N. C. Laksana, “Apa itu Industri 4.0 dan bagaimana Indonesia menyongsongnya,” *tek.id*, 18 Februari 2019. [Online]. Available: <https://www.tek.id/tek/apa-itu-industri-4-0-dan-bagaimana-indonesia-menyongsongnya-b1Xbl9d4L>. [Diakses 10 Oktober 27].
- [2] “About us,” PT. Sin A Sixfifteen, [Online]. Available: <http://sinasixfifteen.com/about/>. [Diakses 2019 Oktober 27].
- [3] M. Banzi dan M. Shiloh, *Getting Started with Arduino: The Open Source Electronics Prototyping Platform*, Maker Media, Inc., 2014.
- [4] S. K. Dewi, R. D. Nyoto dan E. D. Marindani, “Perancangan Prototipe Sistem Kontrol Suhu dan Kelembaban pada Gedung Walet dengan Mikrokontroler Berbasis Mobile,” *Jurnal Edukasi dan Penelitian Informatika*, vol. 4, no. 1, pp. 36-42, 2018.
- [5] M. F. Areed, “A keyless Entry System based on Arduino board with Wi-Fi technology,” *Measurement*, vol. 139, pp. 34-39, 2019.
- [6] M. Saleh dan M. Haryanti, “Rancang Bangun Sistem Keamanan Rumah Menggunakan Relay,” *Jurnal Teknologi Elektro*, vol. 8, no. 3, pp. 181-186, 2017.
- [7] G. Cooper, *Air-conditioning America: Engineers and the Controlled Environment*, JHU Press, 2002.
- [8] “MySQL 8.0 Reference Manual: What is MySQL?,” Oracle, [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. [Diakses 27 Oktober 2019].
- [9] “CodeIgniter at a Glance,” CodeIgniter, [Online]. Available: https://codeigniter.com/user_guide/overview/at_a_glance.html. [Diakses 27 Oktober 2019].

- [10] "DHT22 - SparkFun Electronics," [Online]. Available: <https://www.sparkfun.com/dataheets/Sensors/Temperature/DHT22.pdf>. [Diakses 27 Oktober 2019].
- [11] D. A. Martillano, R. G. Reyes, I. R. B. Miranda dan K. L. C. Diaz, "Android-Based Smart Power Outlet Switching Device Using ESP8266 Enabled WiFi Module," *Journal of Advances in Computer Networks*, vol. 6, no. 1, pp. 61-65, 2018.
- [12] S. K. Memon, F. K. Shaikh, N. A. Mahoto dan A. A. Memon, "IoT based smart garbage monitoring & collection system using WeMos & Ultrasonic sensors," *2nd International Conference on Computing, Mathematics and Engineering Technologies*, pp. 1-6, 2019.

BIODATA PENULIS



Penulis bernama Aditya Budi Laksono, lahir di Serang pada tanggal 16 Juli 1998, merupakan anak pertama dari dua bersaudara. Penulis telah menempuh jenjang Pendidikan formal di beberapa sekolah, yaitu: SD Mutiara Bunda Cilegon (2006 - 2011), SMP Negeri 2 Cilegon (2011 - 2014), dan SMA Negeri 1 Cilegon (2014 - 2016). Penulis melanjutkan pendidikan sarjana di Departemen Sistem Informasi Fakultas Teknologi Informasi dan Komunikasi (FTIK) Institut Teknologi Sepuluh Nopember (ITS) pada tahun 2016 yang terdaftar sebagai mahasiswa dengan NRP 05211640000003.

Selama menjadi mahasiswa, penulis aktif mengikuti organisasi kemahasiswaan Himpunan Mahasiswa Sistem Informasi ITS pada tahun kepengurusan 2017-2018 dan 2018-2019 serta kegiatan kepanitiaan seperti Information Systems Expo 2018 dan *Volunteer* ITS Expo 2016. Kegiatan lain yang pernah diikuti yang berkaitan dengan keprofesian adalah Indonesia Android Kejar tahun 2016.

Pada Juli hingga Agustus 2019, penulis melakukan kegiatan magang di PT. Sin A Sixfifteen sebagai *Software Developer* pada Departemen Teknologi Informasi. Penulis dapat dihubungi melalui email adittbudi@gmail.com