

**PEMBANGUNAN PERANGKAT LUNAK
UNTUK SISTEM PENYIRAMAN TANAMAN OTOMATIS
BERBASIS *WEBSITE***

(STUDI KASUS PENYIRAMAN TANAMAN RUMAH)

TUGAS AKHIR

Disusun sebagai salah satu syarat untuk kelulusan Program Strata 1,
di Program Studi Teknik Informatika, Universitas Pasundan Bandung

oleh :

Rd Suga Dewantha Garsela
NRP : 12.304.0207



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PASUNDAN BANDUNG
JANUARI 2019**

LEMBAR PENGESAHAN
LAPORAN TUGAS AKHIR

Telah diujikan dan dipertahankan dalam Sidang Sarjana Program Studi Teknik Informatika Universitas Pasundan Bandung, pada hari dan tanggal sidang sesuai berta acara sidang, tugas akhir dari :

Nama : Rd Suga Dewantha Garsela
Nrp : 12.304.0207

Dengan judul :

**“PEMBANGUNAN PERANGKAT LUNAK UNTUK SISTEM PENYIRAMAN TANAMAN
OTOMATIS BERBASIS *WEBSITE*”**

Bandung, 04 Januari 2019

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,

(Dr. Ayi Purbasari, ST.MT.,)

(Wanda Gusdya, ST.MT.,)

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

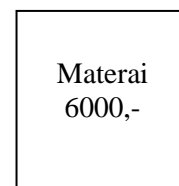
Saya menyatakan dengan sesungguhnya bahwa :

1. Tugas akhir ini adalah benar-benar asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Universitas Pasundan Bandung maupun di Perguruan Tinggi lainnya
2. Tugas akhir ini merupakan gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain kecuali arahan dari tim Dosen Pembimbing
3. Dalam tugas akhir ini tidak terdapat karya atau pendapat orang lain, kecuali bagian-bagian tertentu dalam penulisan laporan Tugas Akhir yang saya kutip dari hasil karya orang lain telah dituliskan dalam sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan karya ilmiah, serta disebutkan dalam Daftar Pustaka pada tugas akhir ini
4. Kakas, perangkat lunak, dan alat bantu kerja lainnya yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab saya, bukan tanggung jawab Universitas Pasundan Bandung

Apabila di kemudian hari ditemukan seluruh atau sebagian laporan tugas akhir ini bukan hasil karya saya sendiri atau adanya plagiasi dalam bagian-bagian tertentu, saya bersedia menerima sanksi akademik, termasuk pencabutan gelar akademik yang saya sandang sesuai dengan norma yang berlaku di Universitas Pasundan, serta perundang-undangan lainnya

Bandung, 04 Januari 2019

Yang membuat pernyataan,



(Rd Suga Dewantha Garsela)

NRP. 12.304.0207

ABSTRAK

Perkembangan teknologi informasi saat ini sangat cepat sehingga memberikan dampak pada globalisasi, persaingan bisnis, tuntutan pekerjaan dan tuntutan gaya hidup menjadi semakin meningkat. Salah satunya yaitu menggunakan model *Internet of Things (IoT)* yang menghasilkan peluang bisnis baru dengan menghubungkan benda-benda fisik dengan berbagai sensor.

Penelitian dan pengembangan dapat dilakukan dengan berbagai macam cara untuk mengoprasikan teknologi yang ada saat ini, salah satunya menggunakan sensor, komputer atau bahkan *smartphone* diperlukan media komunikasi dan antarmuka aplikasi. Media komunikasi yang dapat digunakan untuk menghubungkan perangkat tersebut dengan perangkat lunak yang dipasang pada komputer atau *smartphone* dapat melalui media komunikasi TPC (*Twisted Pair Cable*), PLC (*Power Line Communication*), RF (*Radio Frequency*), atau *Ethernet/IP (Internet Protocol)*.

Tugas akhir ini telah mengembangkan penelitian sebelumnya dengan membangun perangkat lunak untuk sistem penyiraman tanaman otomatis berbasis *website*. Hasil penelitian ini terdiri dari perangkat lunak dan perangkat keras, pada perangkat lunak berbasis *website* menggunakan bahasa pemograman php dengan memanfaatkan *framework* codeigniter dan perangkat keras dibangun menggunakan mikrokontroler Wemos D1 dengan menggunakan satu sensor yaitu sensor kelembaban (*soil moisture sensor*). Perangkat penyiraman ini diberi nama *Splash 'Em All*, perangkat lunak dan perangkat keras dapat saling terhubung menggunakan *WiFi*. *Splash 'Em All* dapat menangani penyiraman otomatis sesuai jadwal dan dapat menampilkan informasi status kelembaban.

Kata kunci: Penyiraman Tanaman Otomatis, Smart Watering Systems, MVC, Website, Wemos, Arduino, Web Services.

ABSTRACT

The development of information technology nowadays is grown fast so as to provide an increase in globalization, business competition, job improvement, and increased lifestyle recovery. One of them uses the Internet of Things (IoT) model that produces new business opportunities by connecting physical objects with various sensors.

Research and development can be done in various ways to operate existing technologies, one of which is using sensors, computers or even smartphones required communication media and application interfaces. Communication media that can be used to connect this device with software installed on a computer or smartphone can be via TPC (Twisted Pair Cable) communication media, PLC (Power Channel Communication), RF (Radio Frequency), or Ethernet / IP (Internet Protocol).

This research has developed previous research by building software for a web-based automated plant watering system. The results of this research consisted of software and hardware, on site-based software using the PHP programming language by using codeIgniter framework and hardware built using the Wemos D1 microcontroller with a single sensor that is the humidity sensor (soil moisture sensor). This watering device is named Splash 'Em All, software and hardware can be connected to each other using WiFi. Splash 'Em All can do automatic watering by schedule and can display moisture status information.

Key word: Penyiraman Tanaman Otomatis, Smart Watering Systems, MVC, Website, Wemos, Arduino, Web Services.

KATA PENGANTAR



Assalamu'alaikum Warahmatullah Wabarakatuh

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena berkat rahmat-Nya penulis bisa menyelesaikan Laporan Tugas Akhir dengan judul “Pembangunan Perangkat Lunak Untuk Sistem Penyiraman Tanaman Otomatis Berbasis Website (Studi Kasus Penyiraman Tanaman Rumah)”. Tugas Akhir merupakan salah satu dari syarat kelulusan mahasiswa Jurusan Teknik Informatika Fakultas Teknik Universitas Pasundan Bandung untuk mendapatkan gelar Strata 1 (S1).

Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu sehingga Laporan Tugas Akhir ini dapat diselesaikan tepat pada waktunya. Untuk itu penulis mengucapkan terimakasih kepada:

1. Allah SWT., atas anugerah yang tidak terbatas, sehingga dapat terselesaikannya laporan Tugas Akhir ini.
2. Nabi Muhammad SAW., semoga kemuliaan serta rahmatnya senantiasa di limpahkan kepada beliau beserta umatnya.
3. Teristimewa, ibunda tercinta ibu Diana Widianingsih, S.SOS., yang selalu berdoa dan memberikan bantuan baik dalam segi moral maupun moril yang tak terhingga demi keberhasilan penulis.
4. Kedua kaka penulis, Rd Lalitha Dewanthi Utami dan Rd Radita Dewangga Utama yang selalu memberikan dukungan dan do'a.
5. Ibu Dr. Ayi Purbasari, ST.,MT., Ketua Program Studi Teknik Informatika, Fakultas Teknik Universitas Pasundan Bandung.
6. Ibu Dr. Ayi Purbasari, ST.,MT., sebagai pembimbing utama yang telah membimbing selama pengerjaan tugas akhir.
7. Bapak Wanda Gusdya, ST.,MT., sebagai pembimbing pendamping yang telah membimbing selama pengerjaan tugas akhir.
8. Jihaan Nabilah Ulayya yang selalu menjadi support sistem dengan memberikan dukungan dan doa.
9. Yoga Ishandiary, Rizky Ashari, Indra Ridwanillah yang selalu menjadi support sistem dengan menemani dan memberikan dukungan tanpa henti.
10. Seluruh keluarga HMBN yang selalu memberikan motivasi dan dukungannya, terutama Dzulfikar Lutfhi R dan Rizki Setia Praja.
11. Teman – teman RPL Azam Hizbul, Rainata Triyadi, Deyani Hanifa, Erwins Wiradhika, Ilham Alamsyah yang selalu memberikan dukungan.

12. Ricko Septian, Muhammad Angga, Arif Rahman, Ahdiyaka Robby sebagai Starter Gang yang selalu memberikan dukungan sejak awal kuliah.
13. Seluruh keluarga Himpunan Teknik Informatika Universitas Pasundan Bandung periode 2014/2015 yang selalu memberikan dukungan.
14. Seluruh keluarga di Teknik Informatika Universitas Pasundan Bandung yang selalu memberikan dukungan.
15. Segenap dosen di Teknik Informatika Universitas Pasundan Bandung yang telah memberikan ilmu.
16. Para sahabat yang selalu memberikan semangat.

Penulis menyadari Laporan Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi sempurnanya Laporan Tugas Akhir ini. Semoga Laporan Tugas Akhir ini memberikan manfaat untuk pengembangan wawasan dan peningkatan ilmu pengetahuan bagi kita semua.

Bandung, 04 Januari 2019

Penulis

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT.....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR ISTILAH.....	viii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN.....	xi
DAFTAR SIMBOL.....	xii
BAB 1 PENDAHULUAN.....	1-1
1.1. Latar Belakang.....	1-1
1.2 Identifikasi Masalah.....	1-2
1.3 Tujuan Tugas Akhir.....	1-2
1.4 Lingkup Tugas Akhir.....	1-3
1.5 Metodologi Tugas Akhir.....	1-3
1.6 Sistematika Penulisan Tugas Akhir.....	1-5
BAB 2 LANDASAN TEORI.....	2-1
2.1. Perangkat Lunak.....	2-1
2.1.1. Website.....	2-1
2.1.2. Web Server.....	2-1
2.1.3. Sistem.....	2-2
2.1.4. CodeIgniter Framework.....	2-2
2.1.5. Model View Controller (MVC).....	2-3
2.1.6. PHP.....	2-4
2.1.6.1. Kelebihan PHP.....	2-5
2.1.6.2. Kekurangan PHP.....	2-5
2.1.7. MySQL.....	2-5

2.2.	Arduino.....	2-5
2.2.1.	Spesifikasi Arduino	2-6
2.2.2.	Wemos D1 ESP8266.....	2-6
2.2.3.	Water Pump.....	2-7
2.2.4.	Sensor Kelembaban Tanah (<i>Soil Moisture Sensor</i>).....	2-7
2.2.5.	Relay.....	2-8
2.3	Web Service	2-8
2.4	Internet of Things (IoT).....	2-9
2.5	Model Proses Pembangunan Perangkat Lunak (<i>Waterfall</i>)	2-10
2.5.1	Kelebihan <i>Waterfall</i>	2-11
2.5.2	Kekurangan <i>Waterfall</i>	2-11
2.6	Penelitian Terdahulu	2-11
BAB 3	SKEMA PENELITIAN	3-1
3.1.	Alur Penelitian.....	3-1
3.2.	Analisis Masalah dan Solusi Tugas Akhir.....	3-3
3.2.1.	Analisis Masalah	3-4
3.2.2.	Analisis Relevansi Solusi	3-4
3.3.	Kerangka Berpikir Teoritis	3-5
3.3.1.	Penggunaan Konsep	3-5
3.3.2.	Peta Analisis	3-8
3.4.	Profile Tempat dan Objek Penelitian.....	3-9
BAB 4	ANALISIS DAN PERANCANGAN	4-1
4.1.	Analisis Perangkat Lunak.....	4-1
4.1.1.	Model Bisnis.....	4-1
4.1.1.1.	Activity Diagram.....	4-1
4.1.1.1	User Requirement	4-3
4.1.2	Model Use Case.....	4-4
4.1.2.1	Kebutuhan Perangkat Lunak.....	4-4
4.1.2.1.1	Kebutuhan Fungsionalitas	4-4
4.1.2.1.2	Kebutuhan Non Fungsionalitas	4-5

4.1.2.2	Use Case Diagram.....	4-5
4.1.2.2.1	Definisi Aktor.....	4-6
4.1.2.2.2	Definisi Use Case.....	4-6
4.1.2.3	Skenario Use Case	4-7
4.1.2.4	Prototype Perangkat Lunak.....	4-9
4.1.2.5	Sequence Diagram	4-12
4.1.2.6	Diagram Kelas Analisis	4-14
4.2	Perancangan Perangkat Lunak.....	4-14
4.2.1	Perancangan Kelas	4-14
4.2.2	Perancangan Data.....	4-18
4.2.3	Perancangan <i>Algoritma</i> dan <i>Query</i>	4-19
4.2.3.1	Algoritma dan query dari fungsi meyiram tanaman terjadwal.....	4-19
BAB 5	IMPLEMENTASI DAN PENGUJIAN	5-1
5.1.	Implementasi	5-1
5.1.1	Implementasi Perangkat Lunak	5-1
5.1.1.1	Kebutuhan implementasi.....	5-1
5.1.1.2	Struktur program dan kode program	5-2
5.1.2	Implementasi rancangan antarmuka	5-3
5.1.2.1	Implementasi rancangan antarmuka perangkat lunak	5-3
5.2.	Pengujian	5-7
5.2.1.	Rencana Pengujian	5-7
5.2.1.1.	Metode Pengujian	5-7
5.2.1.2.	Fungsionalitas aplikasi	5-7
5.1.2.2	Skenario pengujian.....	5-8
5.1.2.3	Pengujian perangkat lunak	5-9
5.1.2.4	Pengujian perangkat lunak	5-10
BAB 6	KESIMPULAN DAN SARAN	6-1
6.1.	Kesimpulan.....	6-1
6.2	Saran.....	6-1

DAFTAR ISTILAH

No	Nama Istilah	Deskripsi
1.	Windows	Sistem Operasi
2.	PHP	Bahasa Pemograman
3.	<i>Requirement</i>	Kebutuhan
4.	C++	Bahasa Pemograman
5.	<i>Database</i>	Basis Data
6.	Smart Phone	Ponsel Cerdas
7.	Web Service	Layanan Web
8.	<i>Platform</i>	Serambi
9.	Microkontroller	Pengendali mikro / mikroprosesor
10.	Arduino	Mikrokontroler
11.	Wemos	Mikrokontroler
12.	<i>Interface</i>	Antar Muka
13.	Water Fall	Model Pembangunan
14.	Arduino IDE	Kakas Pembangunan
15.	JSON	Data
16.	<i>Query</i>	Perintah atau bahasa pemrograman untuk transaksi database
17.	ESP8266	Modul wifi mikrokontroler
18.	Sensor Humidity	Sensor kelembaban
19.	Relay	Saklar
20.	Brushless water pump	Pompa air tanpa penyaring

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	2-11
Tabel 3.1 Kerangka Tugas Akhir	3-1
Tabel 3.2 Analisis Penggunaan Konsep.....	3-6
Tabel 3.3 Penjelasan Analisis Penggunaan Konsep.....	3-7
Tabel 3.4 Peta Analisis	3-8
Tabel 3.5 Langkah Analisis	3-8
Tabel 4.1 User Requirement	4-4
Tabel 4.2 Kebutuhan Fungsionalitas.....	4-4
Tabel 4.3 Kebutuhan Non Fungsionalitas.....	4-5
Tabel 4.4 Definisi Aktor	4-6
Tabel 4.5 Definisi Use Case.....	4-6
Tabel 4.6 Skenario use case Penyiraman Tanaman	4-7
Tabel 4.7 Skenario use case Pengecekan Suhu Tanaman.	4-8
Tabel 4.8 Skenario use case Pengelolaan Jadwal Penyiraman	4-8
Tabel 4.9 Skenario use case Menyiram tanaman terjadwal	4-9
Tabel 4.10 Tabel Rancangan Operasi Kelas	4-15
Tabel 4.11 Atribut kelas.....	4-16
Tabel 4.12 Perancangan Data.....	4-18
Tabel 4.13 Algoritma dari fungsi menyiram tanaman terjadwal	4-19
Tabel 5.1 Spesifikasi Perangkat Keras untuk perangkat lunak berbasis website.....	5-1
Tabel 5.2 Perangkat lunak untuk pendukung <i>website</i>	5-2
Tabel 5.3 Struktur folder perangkat lunak penyiraman tanaman otomatis	5-2
Tabel 5.4 Tabel fungsionalitas perangkat lunak	5-8
Tabel 5.5 Tabel skenario pengujian	5-9
Tabel 5.6 Tabel Skenario Pengujian Perangkat Lunak	5-10

DAFTAR GAMBAR

Gambar 1.1 Metedologi Tugas Akhir	1-3
Gambar 2.1 <i>Application Flow Chart CodeIgniter Framework</i> [AND11].....	2-2
Gambar 2.2 <i>Model-View-Controller</i> [AND11].....	2-3
Gambar 2.3. Wemos D1 ESP8266.....	2-6
Gambar 2.4 . DC 12v <i>brushless submersible water pump</i>	2-7
Gambar 2.5 . <i>Soil Moisture Sensor</i>	2-7
Gambar 2.6 . <i>Relay</i>	2-8
Gambar 2.7 Komponen <i>Web Service</i> [KRE01].....	2-9
Gambar 2.8 Model Proses Waterfall.....	2-10
Gambar 3.1 Diagram <i>fishbone</i>	3-3
Gambar 3.2 Gambaran umum sistem penyiraman otomatis berbasis website.....	3-5
Gambar 4.1 Activity Diagram Penyiraman Tanaman.....	4-1
Gambar 4.2 Activity Diagram Mengecek Suhu Tanaman	4-2
Gambar 4.3 Activity Diagram Kelola Jadwal Penyiraman.....	4-2
Gambar 4.4 Activity Diagram Menyiram Menurut Jadwal	4-3
Gambar 4.5 Use Case Diagram.....	4-6
Gambar 4.6 Rancangan <i>Prototype Login</i>	4-10
Gambar 4.7 Rancangan <i>Prototype Dashboard Admin</i>	4-10
Gambar 4.8 Rancangan <i>Prototype Penyiraman Tanaman</i>	4-11
Gambar 4.9 Rancangan <i>Prototype Pengelolaan Jadwal Penyiraman</i>	4-11
Gambar 4.10 <i>Sequence Diagram</i> Penyiraman Tanaman	4-12
Gambar 4.11 <i>Sequence Diagram</i> Pengecekan Suhu Tanaman	4-12
Gambar 4.12 <i>Sequence Diagram</i> Pengelolaan Jadwal Penyiraman.....	4-13
Gambar 4.13 <i>Sequence Diagram</i> Menyiram Tanaman Terjadwal.....	4-13
Gambar 4.14 Diagram Kelas Analisis.....	4-14
Gambar 4.15 Diagram Kelas Perancangan	4-17
Gambar 4.16 Relasi Kelas Model	4-18
Gambar 5.1 Antarmuka <i>login</i>	5-3
Gambar 5.2 Antarmuka <i>Dashboard</i>	5-4
Gambar 5.3 Antarmuka penyiraman tanaman	5-5
Gambar 5.4 Antarmuka pengelolaan jadwal.....	5-5
Gambar 5.5 Antarmuka <i>insert</i> jadwal	5-6
Gambar 5.6 Antarmuka <i>update</i> jadwal	5-7
Gambar A.1 fungsi pemanggilan kelas <i>view</i>	1
Gambar A.2 fungsi <i>viewJadwal</i>	2
Gambar A.3 fungsi <i>insertJadwal</i>	2
Gambar A.4 fungsi <i>formUpdateJadwal</i>	2
Gambar A.5 fungsi <i>updateJadwal</i>	3
Gambar A.6 fungsi <i>deleteJadwal</i>	3
Gambar A.7 fungsi <i>logged_in_check</i>	3
Gambar A.8 fungsi <i>deleteJadwal</i>	4
Gambar A.9 konfigurasi WiFi pada Wemos	4
Gambar A.10 konfigurasi <i>void setup</i> pada Wemos.....	5
Gambar A.11 konfigurasi <i>void loop</i> pada Wemos	5
Gambar A.12 konfigurasi <i>void loop</i> pada Wemos	6
Gambar A.13 Skema rangkaian	7
Gambar A.14 perangkat keras penyiraman tanaman otomatis tampak luar.....	7
Gambar A.15 perangkat keras penyiraman tanaman otomatis tampak dalam	8

DAFTAR LAMPIRAN

LAMPIRAN A.1 Penulisan kode program bahasa pemrograman PHP.....	A-1
LAMPIRAN A.1.1 Fungsi yang terdapat pada <i>controller</i> utama masterController.....	A-1
LAMPIRAN A.1.2 Fungsi yang terdapat pada <i>controller</i> utama Auth.....	A-3
LAMPIRAN A.1.3 Penulisan kode program pada Arduino IDE.....	A-4
LAMPIRAN A.2 Skema Wemos D1 ESP8266.....	A-6
LAMPIRAN A.3 Gambar hasil akhir perangkat keras penyiraman tanaman otomatis.....	A-7

DAFTAR SIMBOL

Flowchart



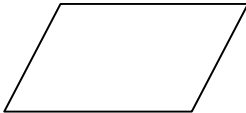
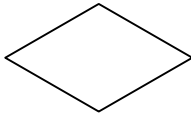




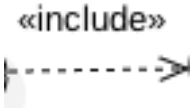

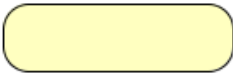
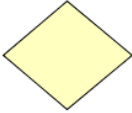


No	Simbol	Nama	Keterangan
1		<i>Start/End</i>	Simbol untuk permulaan (<i>start</i>) atau akhir (<i>stop</i>) dari suatu kegiatan.
2		<i>Process</i>	Simbol yang menunjukkan pengolahan yang dilakukan oleh prototype.
3		<i>Data</i>	Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.
4		<i>Decision</i>	Simbol pemilihan proses berdasarkan kondisi yang ada.
5		<i>Off-page reference</i>	Simbol untuk keluar – masuk atau penyambungan proses pada lembar / halaman yang berbeda.





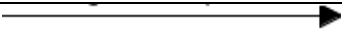

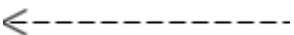
Diagram Use Case

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Simbol yang menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
3		<i>Association</i>	Simbol yang menghubungkan antara objek satu dengan objek lainnya.
4		<i>Include</i>	Simbol yang menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .

Activity Diagram

No	Simbol	Nama	Keterangan
1		<i>Start state</i>	Start state dengan tegas menunjukkan dimulainya suatu workflow pada sebuah <i>activity diagram</i> .
2		<i>Activity</i>	<i>Activity</i> menggambarkan sebuah pekerjaan atau tugas dalam workflow.
3		<i>Decision</i>	<i>Decision</i> adalah suatu titik atau point pada <i>activity diagram</i> yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.
4		<i>End state</i>	<i>End state</i> menggambarkan akhir atau terminal dari pada sebuah <i>activity diagram</i> .
5		<i>State Transition</i>	<i>State transition</i> menunjukkan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya.

Sequence Diagram

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Simbol yang menggambarkan pengguna yang sedang berinteraksi dengan sistem.
2		<i>Boundary</i>	Simbol yang menggambarkan sebuah penggambaran dari <i>form</i> .
3		<i>Controller</i>	Simbol yang menggambarkan penghubung antara <i>boundary</i> dengan tabel atau <i>entity</i> .
4		<i>Entity</i>	Simbol yang menggambarkan hubungan kegiatan yang akan dilakukan.
5		<i>Object Message</i>	Simbol yang menggambarkan pengiriman pesan
6		<i>Message to Self</i>	Simbol yang menggambarkan pengiriman pesan pada dirinya sendiri.
7		<i>Return Message</i>	Simbol yang menggambarkan pengiriman pesan kembali.

BAB 1

PENDAHULUAN

Bab ini membahas mengenai latar belakang masalah, identifikasi masalah, lingkup masalah, maksud dan tujuan, metode penelitian dan sistematika penulisan laporan tugas akhir.

1.1. Latar Belakang

Perkembangan teknologi informasi yang sangat cepat telah memberikan dampak pada globalisasi, persaingan bisnis, tuntutan pekerjaan, dan tuntutan gaya hidup menjadi semakin meningkat. Salah satunya yaitu menggunakan model *Internet of Things* (IoT) menghasilkan peluang bisnis baru dengan menghubungkan benda-benda fisik dengan banyak sensor. Penelitian IoT terutama difokuskan pada teknologi dan model bisnis yang relatif belum diselidiki dan perlu dikembangkan [SUG17]. Fungsi utama dari *Internet of Things* adalah sebagai sarana yang memudahkan untuk pengawasan dan pengendalian barang fisik maka konsep *Internet of Things* ini sangat memungkinkan untuk digunakan hampir pada seluruh kegiatan sehari-hari, mulai dari penggunaan perorangan, perkantoran, rumah sakit, pariwisata, industri, transportasi, pertanian dan peternakan, sampai ke pemerintahan [RAT17].

Terdapat berbagai macam cara mengoperasikan teknologi rumah pintar seperti menggunakan sensor, menggunakan komputer atau bahkan *smartphone* diperlukan media komunikasi dan antarmuka aplikasi. Media komunikasi yang biasa digunakan untuk dapat menghubungkan perangkat rumah pintar dengan perangkat lunak yang dipasang pada komputer atau *smartphone* dapat melalui media komunikasi TPC (*Twisted Pair Cable*), PLC (*Power Line Communication*), RF (*Radio Frequency*), atau *Ethernet/IP* (*Internet Protocol*) [WIR17].

Penelitian dan pengembangan dapat dilakukan dengan memanfaatkan berbagai macam teknologi yang ada saat ini, salah satunya dengan memanfaatkan teknologi mikrokontroler ATmega328 yang terpasang pada papan Arduino. Arduino sendiri merupakan pengendali mikro single-board yang bersifat *open-source* dan dapat digunakan sebagai *platform* perangkat keras terbuka. Perangkat ini ditujukan bagi siapa saja yang akan membuat purwarupa peralatan elektronik interaktif berdasarkan perangkat keras dan perangkat lunak yang fleksibel dan mudah digunakan [WIR17].

Sebelumnya telah dilakukan penelitian perangkat lunak penyiraman tanaman otomatis terdiri dari perangkat keras yang dibangun dengan memanfaatkan *Arduino* dan perangkat lunak yang dapat dioperasikan melalui *smartphone android* dengan memanfaatkan sebuah *web service* sebagai jembatan agar dapat saling terhubung antara perangkat keras dan perangkat lunak. Namun pada komponen tambahan perangkat keras terdapat beberapa kekurangan, seperti pompa air manual yang memanfaatkan kran elektrik, kurang luasnya jangkauan siram, dan tidak menggunakan *casing* penutup *Arduino*.

Penulis bermaksud untuk mengembangkan perangkat lunak untuk sistem penyiraman tanaman otomatis dengan mengangkat topik Tugas Akhir yang berjudul “Pembangunan Perangkat Lunak Untuk Sistem Penyiraman Tanaman Otomatis Berbasis Website”. Dengan memanfaatkan perangkat keras mikrokontroler *Wemos D1 WiFi ESP8622*, memperluas jangkauan siram dengan menggunakan dua *relay* dan pompa air menggunakan *brushless 12v DC water pump* untuk dua lokasi penyiraman, megemas perangkat penyiraman dengan *project box* dan perangkat lunak menggunakan *web-based* diharapkan dapat memudahkan pemilik tanaman dalam melakukan penyiraman tanaman.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan, terdapat beberapa poin dari permasalahan yang akan dibahas, yaitu:

1. Bagaimana mengembangkan perangkat penyiraman yang dapat dikontrol melalui *WiFi*?
2. Bagaimana penyiraman dapat menjangkau dua lokasi yang berbeda?
3. Bagaimana membangun perangkat penyiraman yang dapat melakukan penyiraman otomatis sesuai jadwal?
4. Bagaimana membangun perangkat lunak yang dapat mengelola penjadwalan?
5. Bagaimana membangun perangkat lunak yang dapat menampilkan status kelembaban?
6. Bagaimana membangun perangkat lunak yang dapat terhubung dengan perangkat keras mikrokontroller?

1.3 Tujuan Tugas Akhir

Adapun tujuan yang hendak dicapai dalam pengerjaan tugas akhir ini, yaitu:

1. Memanfaatkan mikrokontroler *Wemos D1* sebagai perangkat penyiraman tanaman yang dapat dikendalikan melalui *WiFi*.
2. Mengembangkan sistem penyiraman otomatis yang telah ada dengan memperluas jangkauan penyiraman menggunakan *relay* tambahan.
3. Membangun perangkat lunak yang dapat terkoneksi dengan perangkat keras melalui *WiFi*.

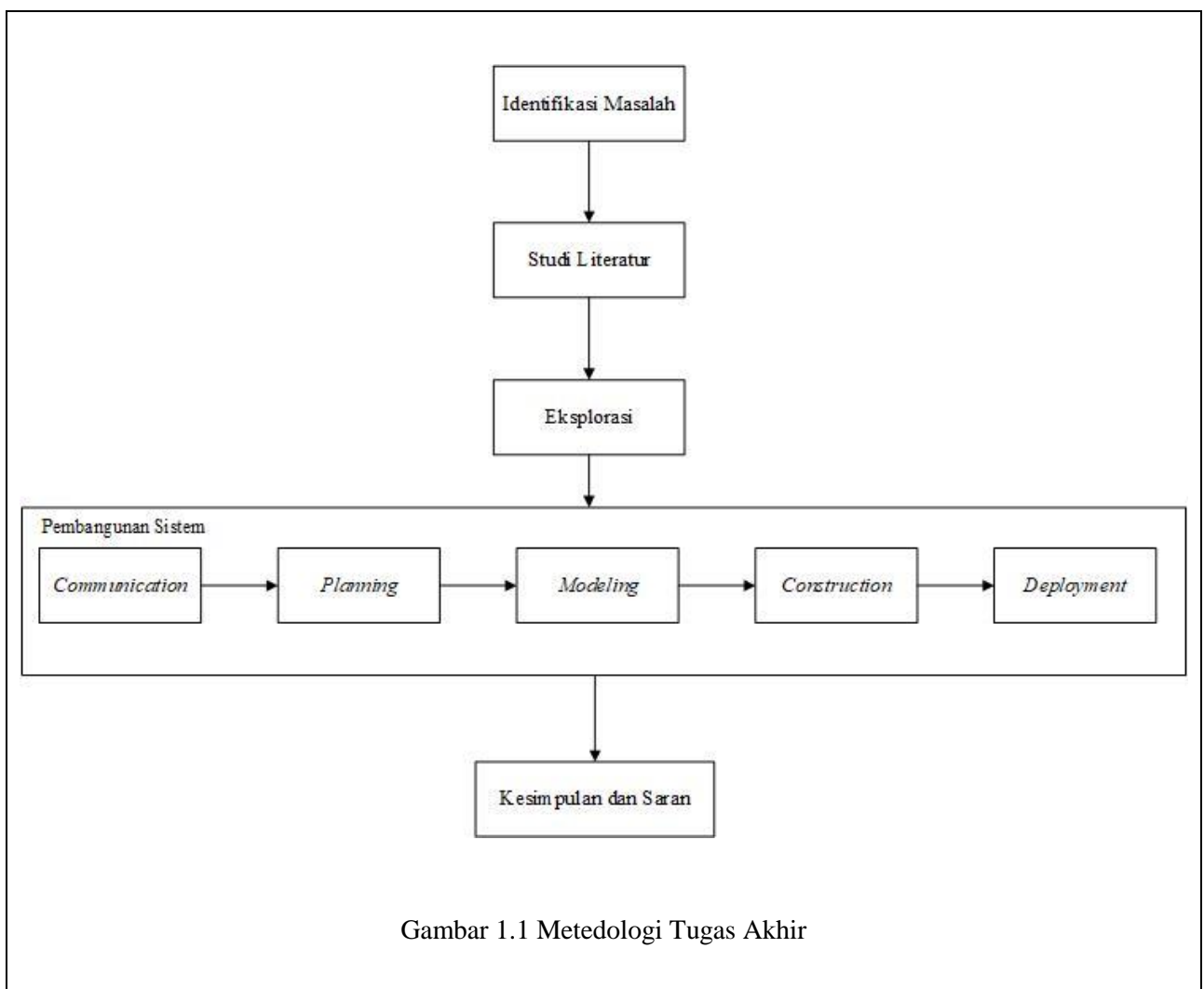
1.4 Lingkup Tugas Akhir

Agar pembahasan Tugas Akhir ini lebih terarah dan tidak keluar dari ruang lingkupnya, maka dibawah ini akan dijabarkan lingkup dari Tugas Akhir ini:

1. Membangun perangkat lunak berbasis *website* yang dapat mengontrol penyiraman tanaman yang telah terhubung dengan mikrokontroler,
2. Perangkat lunak dan perangkat keras saling terhubung secara lokal,
3. Sumber air yang digunakan telah terisi air,
4. Mengoprasikan dua *relay* untuk pompa air,
5. Menggunakan satu sensor,
6. Dikelola oleh satu pengguna.

1.5 Metodologi Tugas Akhir

Metodologi Berikut ini ini merupakan metodologi pengerjaan tugas akhir. Metodologi ini merupakan langkah-langkah yang digunakan untuk melakukan penulisan tugas akhir. Gambaran metodologi yang digunakan pada tugas akhir ini akan dipaparkan pada Gambar 1.1



Gambar 1.1 Metodologi Tugas Akhir

Berikut ini merupakan penjelasan mengenai Gambar 1.1 Metodologi Tugas Akhir :

1. Identifikasi masalah
Mengkaji permasalahan yang akan dibahas pada Tugas Akhir.
2. Studi Literatur
Mengumpulkan informasi dan mempelajari materi serta sumber-sumber data yang berhubungan dan terkait dengan Tugas Akhir.
3. Eksplorasi
Melakukan eksplorasi kakas yang akan digunakan untuk pembangunan Tugas Akhir.
4. Pembangunan Perangkat Lunak
Melakukan pembangunan perangkat lunak berdasarkan metodologi *Waterfall*. Dalam metode tersebut terdapat beberapa tahapan diantaranya:
 - a. **Communication (Project Initiation & Requirements Gathering)**
Komunikasi dengan *costumer* sangat diperlukan demi memahami dan mencapai tujuan yang dimaksud terkait kebutuhan – kebutuhan *costumer* tersebut guna mengetahui lingkup dan batasan perangkat lunak yang akan dibangun.
 - b. **Planning (Estimating, Scheduling, Tracking)**
Tahapan selanjutnya adalah tahap perencanaan yang menjelaskan estimasi modul yang akan dikerjakan, risiko-risiko yang kemungkinan dapat terjadi, sumber daya yang diperlukan, produk yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan.
 - c. **Modeling (Analysis & Design)**
Pada tahap ini dilakukan perancangan dan permodelan perangkat lunak yang akan dibangun yang berfokus pada perancangan struktur data, aritektur perangkat lunak, *interface prototype*.
 - d. **Construction (Code & Test)**
Pada tahap *Construction* akan dilakukan implementasi *prototype* yang sudah dibuat sebelumnya kedalam koding. Setelah proses koding selesai, dilakukan pengujian terhadap perangkat lunak yang sudah dibuat.
 - e. **Deployment (Delivery, Support, Feedback)**
Jika semua tahapan sudah selesai dilakukan selanjutnya adalah tahap *Deployment*. *Deployment* merupakan tahapan implementasi perangkat lunak kepada *customer*, pemeliharaan perangkat lunak secara berkala, dan pengembangan perangkat lunak.
5. Kesimpulan dan Saran
Membuat kesimpulan berdasarkan hasil dari pembangunan Tugas Akhir dan saran untuk penelitian selanjutnya.

1.6 Sistematika Penulisan Tugas Akhir

Adapun dalam pengerjaan Tugas Akhir ini dibagi dalam beberapa bab, yang dijelaskan sebagai berikut:

BAB 1 PENDAHULUAN

Pada bab ini menjelaskan tentang latar belakang, lingkup dari Tugas Akhir, tujuan Tugas Akhir, metodologi Tugas Akhir, serta sistematika penulisan laporan Tugas Akhir.

BAB 2 LANDASAN TEORI

Pada bab landasan teori berisi perluasan dari kerangka pemikiran. Didalamnya dikemukakan definisi-definisi, teori-teori, konsep-konsep yang diperlukan sebagai alat untuk menganalisis yang termaktub di buku-buku teks, ataupun makalah-makalah di jurnal-jurnal ilmiah yang terkait.

BAB 3 SKEMA PENELITIAN

Pada bab ini menjelaskan mengenai hasil analisis dari metode-metode dan studi literatur terkait perangkat lunak penyiraman tanaman otomatis.

BAB 4 ANALISIS DAN PERANCANGAN

Pada bab ini menjelaskan tentang tahap tahap perancangan berdasarkan kebutuhan dari analisis yang telah didefinisikan pada bab sebelumnya.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Pada bab ini menjelaskan tahap implementasi berdasarkan hasil analisis dan hasil rancangan yang telah dibuat pada tahap perancangan.

BAB 6 KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari keseluruhan bahasan tentang jawaban terhadap pertanyaan atau kebutuhan yang dikemukakan sebelumnya, dan saran berupa pemantapan terhadap kesimpulan yang telah dibuat, dengan demikian memantapkan hubungan antara masalah, analisis, pengembangan, dan kesimpulan.

BAB 2

LANDASAN TEORI

Pada bab ini membahas mengenai teori yang mendasari penulisan dari tugas akhir ini. Teori umum yang digunakan meliputi teori pengembangan perangkat lunak penyiraman tanaman dan beberapa teori lainnya yang mendasari penyusunan tugas akhir ini.

2.1. Perangkat Lunak

Perangkat lunak atau *software* adalah perintah (program komputer) yang dieksekusi memberikan fungsi dan petunjuk kerja seperti yang diinginkan. Struktur data yang memungkinkan program memanipulasi informasi secara proporsional dan dokumen yang menggambarkan operasi dan kegunaan program.

Software memiliki dua peran, disatu sisi berfungsi sebagai sebuah produk dan disisi lain sebagai pengontrol pembuatan sebuah produk. Sebagai produk, *software* mengantarkan potensi perhitungan yang dibangun oleh *software* komputer. *Software* merupakan *transformer* informasi yang memproduksi, mengatur, memperoleh, memodifikasi, menampilkan atau memancarkan informasi, ini dapat sesederhana bit tunggal atau sekompleks sebuah simulasi multimedia. Sedangkan peran sebagai pengontrol yang dipakai untuk mengantarkan produk, *software* berlaku sebagai dasar untuk kontrol komputer (sistem operasi), komunikasi informasi (jaringan), dan penciptaan serta kontrol dari program-program lain (peranti dan lingkungan *software*) [JUW11].

2.1.1. Website

Web merupakan fasilitas *hypertext* untuk menampilkan data berupa teks, gambar, bunyi, animasi, dan data multimedia lainnya, yang mana data tersebut saling berhubungan satu sama lain. Dalam sebuah *website* terdapat suatu halaman yang dikenal dengan sebutan *home page*. *Home page* adalah sebuah halaman yang pertama kali dilihat ketika seseorang mengunjungi *website*. Dari *home page*, pengunjung dapat mengklik hyperlink untuk pindah ke halaman lain yang terdapat dalam *website* tersebut [HEN14].

2.1.2. Web Server

Web server adalah suatu perangkat lunak yang mengatur halaman *web* dan membuat halaman – halaman *web* tersebut dapat diakses di klien, yaitu melalui jaringan lokal atau melalui jaringan *Internet*. Ada banyak *web server* yang tersedia diantaranya *Apache*, *ISS (Internet Information Service)*, dan *IPlanet's Enterprise server* [TRI09].

2.1.3. Sistem

Sistem bisa ditafsirkan sebagai kesatuan elemen yang memiliki keterkaitan. Beberapa elemen dapat digabung menjadi satu unit, kelompok atau komponen sistem dengan fungsi tertentu. Komponen sistem ini bisa dilihat, dianggap, atau memang dirancang untuk berfungsi mandiri sebagai *modul* sistem (lepas dari sistem tapi masih berkaitan dengan sistem pada mana modul ini menginduk) [HEN14].

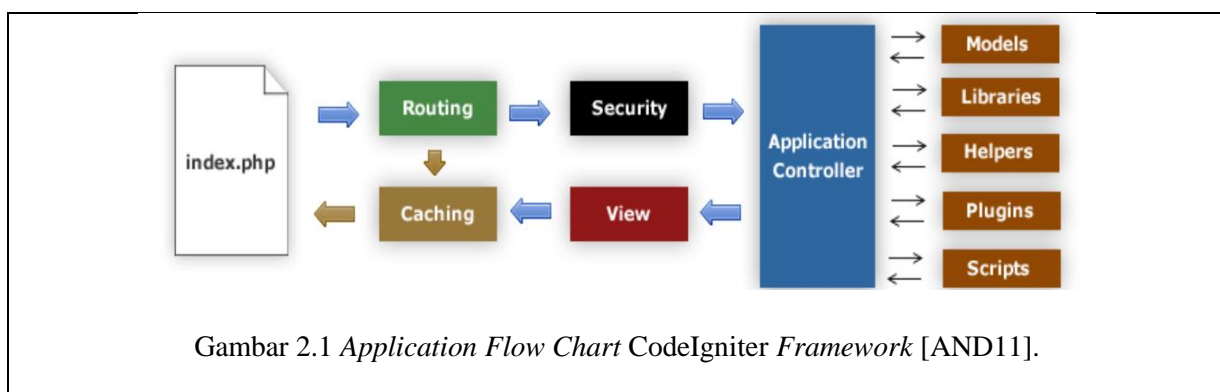
2.1.4. CodeIgniter Framework

CodeIgniter adalah *framework* aplikasi web yang *open source* untuk bahasa pemrograman PHP. CodeIgniter memiliki banyak fitur yang membuatnya berbeda dengan *framework* lainnya. Tidak seperti beberapa *framework* PHP lainnya, dokumentasi untuk *framework* ini sangat lengkap, yang mencakup seluruh aspek dalam *framework*. CodeIgniter juga mampu berjalan pada lingkungan *shared hosting* karena memiliki ukuran yang sangat kecil, namun memiliki kinerja yang sangat luar biasa.

Dari sisi pemrograman, CodeIgniter kompatibel dengan PHP4 dan PHP5, sehingga akan berjalan dengan baik pada *web host* yang banyak dipakai pada saat ini. CodeIgniter menggunakan pola desain *Model-View-Controller* (MVC), yang merupakan cara untuk mengatur aplikasi web ke dalam tiga bagian yang berbeda, yaitu *Model* –lapisan abstraksi database, *Views* –file-file template tampilan depan, dan *Controller* –logika bisnis dari aplikasi. Pada intinya, CodeIgniter juga membuat menggunakan ekstensif dari pola desain *Singleton*. Maksudnya adalah cara untuk me-*load class* sehingga jika *class* itu dipanggil dalam beberapa kali, kejadian yang sama pada *class* tersebut akan digunakan kembali. Hal ini sangat berguna dalam koneksi database, karena kita hanya ingin menggunakan satu koneksi setiap kali *class* itu digunakan.

CodeIgniter dikembangkan oleh Rick Ellis, dengan versi awal yang dirilis pada tanggal 28 Februari 2006. Dari tahun itulah hingga sekarang, telah muncul banyak versi CodeIgniter yang terus berkembang dengan penambahan fitur baru dari versi sebelumnya. Untuk versi terbaru dari CodeIgniter adalah versi 2.0.3 [AND11].

Proses data yang mengalir pada sistem menggunakan CodeIgniter *Framework* diilustrasikan pada gambar *Application Flow Chart CodeIgniter Framework* yang akan dipaparkan pada Gambar 2.1



Gambar 2.1 *Application Flow Chart CodeIgniter Framework* [AND11].

Keterangan:

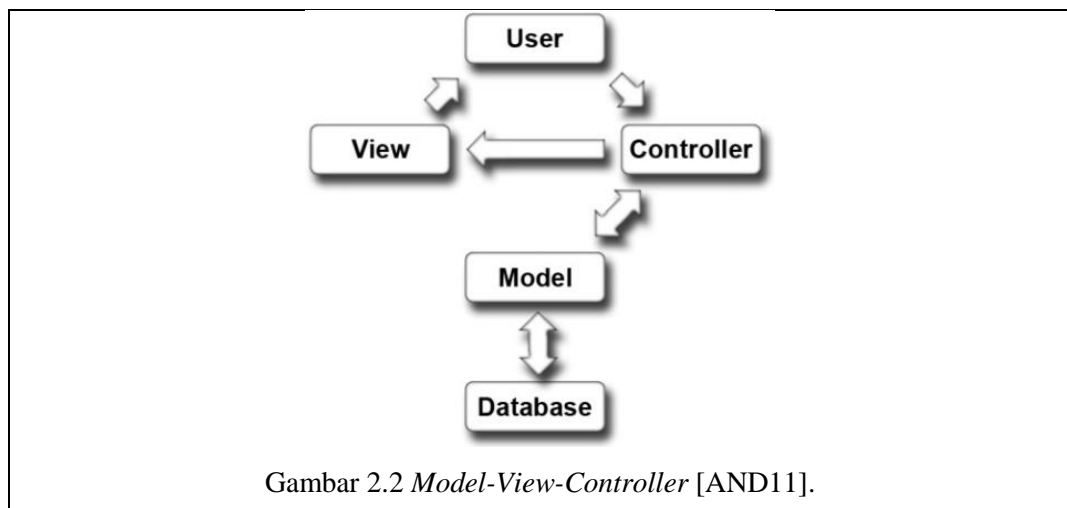
1. ***index.php*** berfungsi sebagai *front controller*, menginisialisasi *base resource* untuk menjalankan CodeIgniter.
2. ***Router*** berfungsi untuk memeriksa *HTTP request* untuk menentukan apa yang harus dilakukan dengannya.
3. Jika ***Cache*** aktif, maka hasilnya akan langsung dikirimkan ke *browser* dengan mengabaikan aliran data normal.
4. ***Security***. Sebelum *controller* dimuat, *HTTP request* dan data yang dikirimkan *user* akan difilter untuk keamanan.
5. ***Controller*** memuat *model*, *core libraries*, *plugins*, *helpers*, dan semua *resource* yang diperlukan untuk memproses *request*.

Akhirnya ***View*** yang dihasilkan akan dikirim ke *browser*. Jika *cache* aktif, maka *view* akan disimpan sebagai *cache* dahulu, sehingga pada *request* berikutnya langsung ditampilkan.

2.1.5. Model View Controller (MVC)

Model-View-Controller (MVC) pertama kali diperkenalkan peneliti Xerox PARC yang bekerja pada bahasa pemrograman *Smalltalk* di akhir tahun 1970-an dan awal 1980-an. *Smalltalk* adalah bahasa pemrograman yang berorientasi objek, bertipe dinamis, dan reflektif. *Smalltalk* pertama kali digunakan dalam pembelajaran edukasi, dan hal lain ini berbeda dari data *mainframe* dan struktur kontrol dalam program *Smalltalk* yang terlibat pada *Windowed User Interfaces*, konsep pemrograman objek, dan kemampuan untuk memonitor dan memodifikasi struktur dan perilakunya sendiri.

Singkatnya, *Model-View-Controller* (MVC), adalah pola desain pengembangan perangkat lunak. MVC adalah sebuah pendekatan untuk memisahkan aplikasi menjadi tiga segmen, yaitu *Models*, *Views*, dan *Controller*. MVC menginstruksikan aplikasi dengan cara tersebut untuk mempromosikan penggunaan kembali dari kode program [AND11]. Gambaran *Model-View-Controller* akan dipaparkan pada Gambar 2.2



Gambar 2.2 *Model-View-Controller* [AND11].

Berdasarkan gambar diatas dapat disimpulkan bahwa ketika *user* melakukan *request*, maka *request* tersebut akan ditangani oleh *Controller*, kemudian *Controller* akan memanggil *Model* jika diperlukan operasi database. Hasil *query* oleh *Model* kemudian akan dikembalikan ke *Controller*. Selanjutnya *Controller* akan memanggil *View* yang tepat dan menyatukannya dengan hasil *query* dari *Model*. Hasil akhir dari operasi ini akan ditampilkan ke *browser*.

1. *Model*

Model merupakan jenis data yang dapat digunakan oleh aplikasi. Beberapa contoh data yang bisa digunakan adalah *database*, *RSS Feed*, *API calls*, dan setiap tindakan lainnya yang melibatkan pengambilan (*retrieving*), pengembalian (*returning*), memperbarui (*updating*), menghapus data (*removing*)

2. *View*

File-file yang ditempatkan pada bagian ini bertanggung jawab untuk menunjukkan data kami pada para pengunjung situs kita, atau pengguna dari aplikasi kita. Tidak ada logika pemrograman, tidak ada *query insert* atau *update* yang harus dijalankan disini, meskipun akses data bisa terjadi pada file-file ini. File-file disini hanya untuk menunjukkan hasil dari dua bagian lainnya. Jadi kita mengambil data dalam *model*, dan menampilkan dalam *view*.

3. *Controller*

Controller adalah logika bisnis dari aplikasi. File-file yang ada di sini akan melayani sebagai perantara antara *Model* dan *Views*. *Controller* akan merespon permintaan HTTP dan menghasilkan halaman *web*. *Controller* adalah inti dari aplikasi karena bagian menentukan bagaimana HTTP harus ditangani.

2.1.6. PHP

PHP adalah sebuah bahasa pemrograman yang didesain agar dapat disisipkan dengan mudah ke halaman HTML. PHP memberikan solusi sangat mudah (karena gratis digunakan) dan dapat berjalan di berbagai jenis *platform*. Pada awalnya memang PHP berjalan di sistem UNIX dan variannya, namun kini dapat berjalan dengan lancar di lingkungan sistem operasi Windows. Suatu nilai tambah yang luar biasa karena proses pengembangan program berbasis *web* dapat dilakukan lintas sistem operasi.

Dengan luasnya cakupan sistem operasi yang mampu menjalankan PHP dan ditambah begitu lengkapnya *function* yang dimilikinya (tersedia lebih dari 400 *function* di PHP yang sangat berguna) tidak heran jika PHP semakin menjadi tren di kalangan *programmer web*.

Penemu bahasa pemrograman ini adalah Rasmus Lerdorf, yang bermula dari keinginan sederhana Lerdorf untuk mempunyai alat bantu dalam memonitor pengunjung yang melihat situs *web* pribadinya. Inilah sebabnya pada awal pengembangannya, PHP merupakan singkatannya dari *Personal Home Page tools*, sebelum akhirnya menjadi *Hypertext Preprocessor* [AND11].

2.1.6.1. Kelebihan PHP

Diantara maraknya pemrograman *server web* ini, adalah ASP yang berkembang ASP .NET, JSP, CFML, dan PHP. Jika dibandingkan di antara 3 terbesar pemrograman *server web* diatas. Terdapat kelebihan dari PHP itu sendiri, yaitu:

1. PHP merupakan sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi seperti *Visual Basic* dan sebagainya,
2. PHP dapat berjalan pada *web server* yang dirilis oleh Microsoft, seperti IIS atau PWS juga pada Apache yang bersifat *open source*,
3. Karena sifatnya yang *open source*, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak milis-milis dan *developer* yang siap membantu pengembangannya,
4. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami,
5. PHP dapat berjalan pada 3 *operating system*, yaitu Linux, Unix, dan Windows, dan juga dapat berjalan secara *runtime* pada saat *console*.

2.1.6.2. Kekurangan PHP

Seperti pemrograman aplikasi *server web* lainnya, PHP pun memiliki kelemahan, di antaranya:

1. Tidak ideal untuk pengembangan skala besar,
2. Tidak memiliki sistem pemrograman berorientasi objek yang sesungguhnya (sampai versi 4). Namun pada versi PHP 5 sudah dilengkapi pemrograman berorientasi objek yang sesungguhnya.

2.1.7. MySQL

MySQL adalah sistem manajemen *database* yang bersifat *open source*. MySQL adalah pasangan serasi dari PHP. MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di Swedia.

MySQL merupakan sistem manajemen *database* yang bersifat relasional. Artinya data-data yang dikelola dalam *database* akan menjadi lebih cepat.

MySQL dapat digunakan untuk mengelola *database* mulai dari yang kecil sampai dengan yang sangat besar. MySQL juga dapat menjalankan perintah-perintah *Structured Query Language* (SQL) untuk mengelola *database-database* relasional yang ada di dalamnya [AND11].

2.2. Arduino

Arduino dikatakan sebagai sebuah *platform* dari *physical computing* yang bersifat *open source*. Arduino juga merupakan kombinasi dari *hardware*, bahasa pemrograman dan ***Integrated Development Environment* (IDE)** yang canggih. IDE adalah sebuah *software* yang sangat berperan untuk menulis

program, meng-*compile* menjadi kode biner dan mengunggah ke dalam *memory* Mikrokontroler [SAR17].

2.2.1. Spesifikasi Arduino

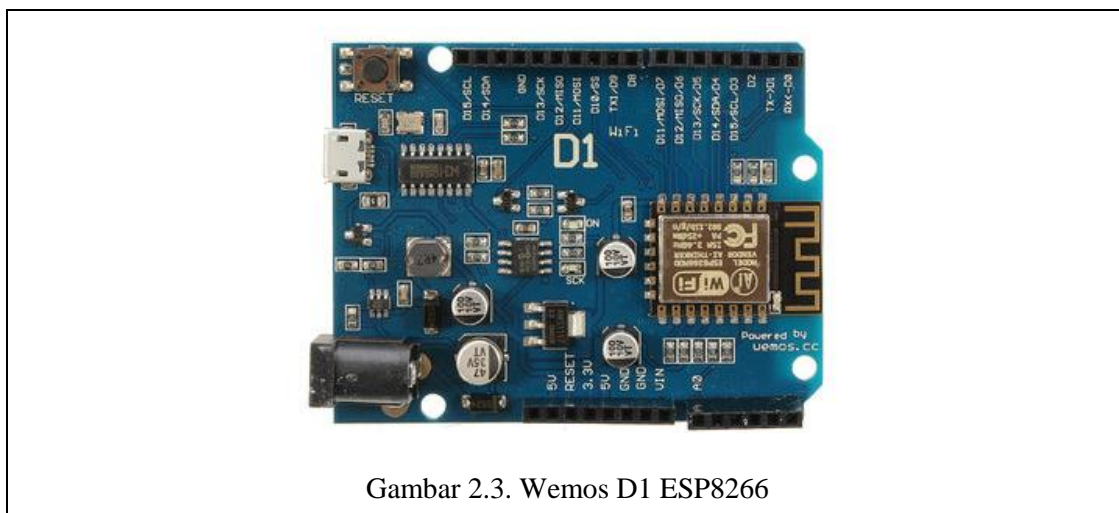
Arduino dikatakan sebagai sebuah *platform* dari *physical computing* yang bersifat *open source*. Arduino juga merupakan kombinasi dari *hardware*, bahasa pemrograman dan ***Integrated Development Environment (IDE)*** yang canggih. IDE adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan mengunggah ke dalam *memory* Mikrokontroler.

Ada banyak project dan alat-alat dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino, selain itu juga ada banyak modul-modul pendukung (sensor, tampilan, penggerak dan sebagainya) yang bisa disambungkan dengan Arduino. Arduino berevolusi menjadi sebuah platform karena ia menjadi pilihan dan acuan bagi banyak praktisi [SAR17].

Salah satu yang membuat Arduino menjadi pilihan banyak orang adalah karna sifatnya yang *open source*, baik *hardware* maupun *software*. Sifat *open source* tersebut menjadikan pengguna bebas men-*download* rangkaian elektronik Arduino, membeli komponen-komponennya dan merangkainya sendiri. Selain itu IDE Arduino pun dapat di-*download* dan diinstal secara gratis. Bahkan pada kemasan Arduino terdapat tulisan bahwa Arduino diperuntukan bagi seniman, perancang dan penemu.

2.2.2. Wemos D1 ESP8266

Wemos D1 adalah sebuah modul *WiFi* yang dikembangkan dari ESP8266. Modul ESP8266 memiliki output serial TTL yang dilengkapi dengan GPIO, yang dapat digunakan secara *stand alone* maupun dengan mikrokontroler tambahan untuk pengendaliannya. Adapun *output serial TTL (Transistor Transistor Logic)* adalah output yang keluarannya bernilai logika *Low* '0' dan *High* '1'. Seringkali dapat dikatakan logika '0' bernilai 0 volt dan logika 3.3 volt atau 5 volt (Vcc) [SUG16]. Gambaran Wemos D1 ESP8266 akan dipaparkan pada Gambar 2.3.



Gambar 2.3. Wemos D1 ESP8266

2.2.3. Water Pump

Mini Subersible Water Pump adalah motor pompa air celup yang berukuran kecil. Pompa air mini ini bisa digunakan untuk akuarium, kolam ikan, hidroponik, robotika atau proyek dalam pembuatan aplikasi yang berbasis mikrokontroller [IMA16].

Mini subersible water pump menggunakan motor DC *Brushless* dan bekerja dengan tegangan DC 12V 240L/hr, kelebihan mini water pump ini adalah tidak berisik dalam penggunaannya dan aman ketika bekerja di dalam air. Gambaran DC 12v *brushless submersible water pump* akan dipaparkan pada Gambar 2.4.



2.2.4. Sensor Kelembaban Tanah (*Soil Moisture Sensor*)

Sensor kelembaban tanah atau dalam istilah bahasa inggris *soil moisture sensor* adalah jenis sensor kelembaban yang mampu mendeteksi intensitas air di dalam tanah (*moisture*). Sensor ini berupa dua lempengan konduktor berbentuk pisau berbahan logam yang sangat sensitif terhadap muatan listrik dalam suatu media khususnya tanah. Kedua lempengan logam tersebut merupakan media yang akan menghantarkan tegangan analog berupa tegangan listrik yang nilainya relatif kecil berkisar antara 3,3-5 volt dan baru kemudian tegangan tersebut akan diubah menjadi tegangan digital untuk diproses lebih lanjut oleh *system* [SAR17]. Gambaran *Soil Moisture Sensor* akan dipaparkan pada Gambar 2.5

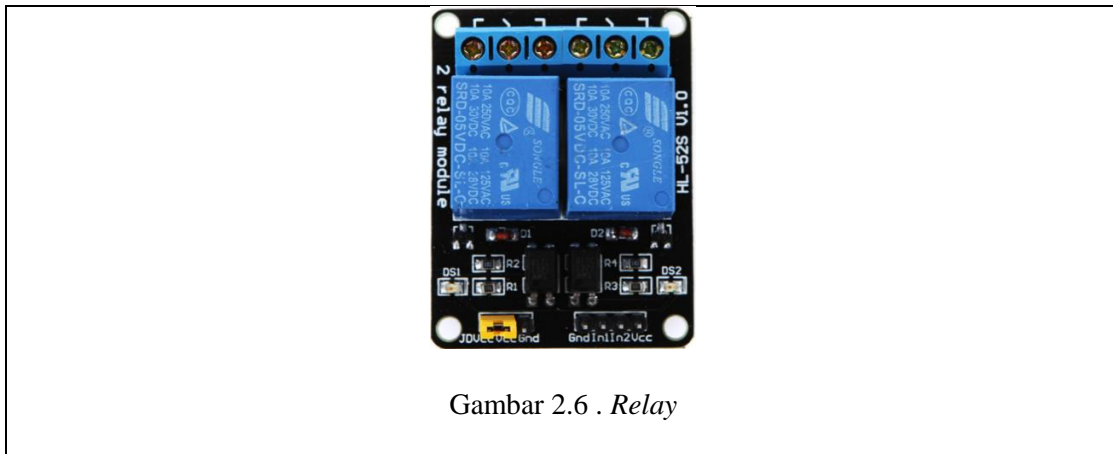


Sensor ini merupakan sensor yang ideal untuk memantau kadar air tanah untuk tanaman. Sensor ini menggunakan dua konduktor untuk memantau kadar air tanah untuk tanaman, kemudian membaca nilai resistansi untuk mendapatkan tingkat kelembaban. Lebih banyak air dalam tanah akan membuat tanah lebih mudah menghantarkan listrik (nilai resistansi lebih besar), sedangkan tanah kering akan mempersulit untuk menghantarkan listrik (nilai resistansi kurang). Sensor *soil moisture* dalam penerapannya membutuhkan daya sebesar 3.3 v atau 5 V dengan keluaran tegangan sebesar 9 – 4.2 V. Sensor ini mampu membaca kadar air yang memiliki 3 kondisi yaitu:

1. 0 – 300 : tanah kering / udara bebas
2. 300 – 800 : tanah lembab
3. 800 – 1023 : tanah basah / di dalam air

2.2.5. Relay

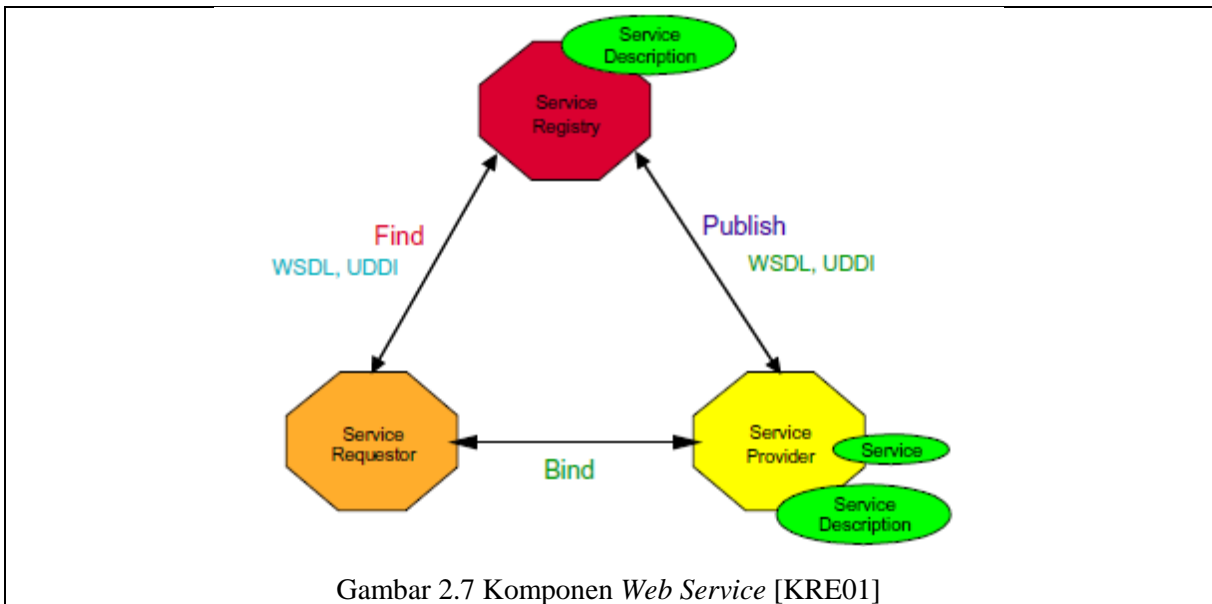
Relay adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. Secara prinsip, *relay* merupakan tuas saklar dengan lilitan kawat pada batang besi (*solenoid*) di dekatnya. Ketika *solenoid* dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet yang terjadi pada *solenoid* sehingga kontak saklar akan menutup. Pada saat arus dihentikan, gaya magnet akan hilang, tuas akan kembali ke posisi semula dan kontak saklar kembali terbuka [SAR17]. Gambaran *Relay* akan dipaparkan pada Gambar 2.6



Gambar 2.6 . Relay

2.3 Web Service

Web service adalah sebuah antarmuka yang mendeskripsikan sekumpulan operasi yang dapat diakses dalam sebuah jaringan melalui pesan XML. Web service dibangun dari tiga komponen utama, yaitu *service provider*, *service registry*, dan *service requestor*. Komponen - komponen tersebut saling berinteraksi sehingga membentuk *web service* yang berupa deskripsi dan implementasi layanan. Terdapat tiga macam operasi yang memungkinkan komponen – komponen tersebut untuk dapat saling berinteraksi yaitu *publish*, *find* dan *bind* [KRE01]. Gambaran Komponen Web Service memperlihatkan keterkaitan antara peran, operasi dan artifak dari *web service* dapat dilihat pada gambar 2.7 di bawah:



Peranan arsitektur dalam web service [KRE01]:

1. *Service provider*

Dari sudut pandang bisnis, komponen ini adalah pihak yang memiliki layanan. Dari sudut pandang arsitektural, komponen ini adalah platform yang menyediakan akses layanan.

2. *Service requestor*

Dari sudut pandang bisnis, komponen ini adalah pihak yang membutuhkan fungsi–fungsi tertentu untuk dipenuhi. Dari sudut pandang arsitektural, komponen ini adalah aplikasi yang mencari dan berinteraksi dengan layanan.

3. *Service registry*

Komponen ini merupakan tempat di mana *service provider* mendaftarkan layanan dan tempat di mana *service requestor* mencari dan mendapatkan informasi layanan yang dibutuhkan.

4. *Publish*

Operasi ini dilakukan oleh *service provider* untuk mempublikasikan *service description* yang akan ditemukan dan diakses oleh *service requestor*.

5. *Find*

Operasi ini merupakan operasi yang dilakukan oleh *service requestor* untuk mendapatkan *service description* yang diinginkan. Terdapat dua alternatif yang dapat dilakukan oleh *service requestor*, yaitu mengambil *service description* secara langsung ke *service provider* atau dengan melakukan *query* ke *service registry*.

6. *Bind*

Operasi ini merupakan operasi yang dilakukan oleh *service requestor* untuk menginisiasi interaksi dengan layanan pada saat waktu dieksekusi.

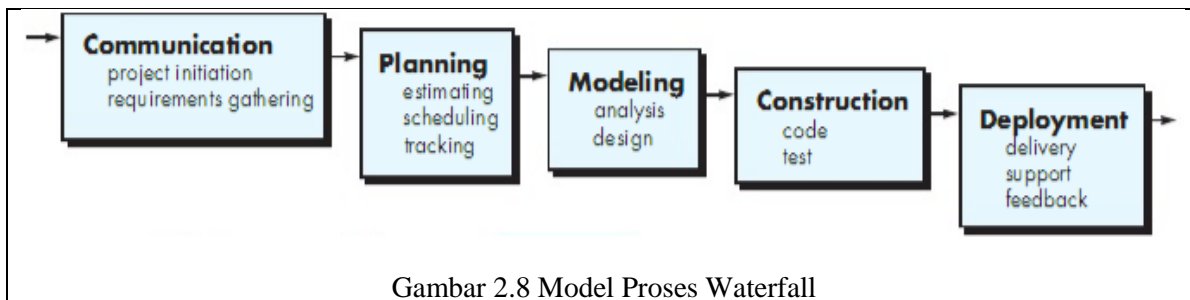
2.4 Internet of Things (IoT)

Internet of Things adalah sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas *internet* yang tersambung secara terus-menerus, berikut kemampuan *remote kontrol*,

berbagi data, dan sebagainya. Bahan pangan, elektronik, peralatan apa saja, koleksi, termasuk benda hidup, yang semuanya tersambung ke jaringan lokal dan *global* melalui sensor tertanam dan selalu 'ON' [SUG17].

2.5 Model Proses Pembangunan Perangkat Lunak (*Waterfall*)

Model proses yang digunakan pada pembangunan perangkat lunak tugas akhir ini adalah model proses *waterfall*. Model proses ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan diperkenalkan pertama kali oleh Winston Royce sekitar tahun 1970. Model proses ini sering disebut juga dengan *classic life cycle*. Model ini melakukan pendekatan secara sistematis dan berurutan dalam membangun perangkat lunak, dimulai dari mengumpulkan spesifikasi kebutuhan konsumen melalui komunikasi (*Communication*), setelah mengumpulkan spesifikasi kebutuhan berlanjut pada tahap perencanaan (*Planning*), pemodelan (*Modeling*), konstruksi (*Construction*) dan implementasi (*Deployment*) [PRE15]. Gambaran Model Proses *Waterfall* akan dipaparkan pada Gambar 2.4



Keterangan:

a. *Communication (Project Initiation & Requirements Gathering)*

Komunikasi dengan *costumer* sangat diperlukan demi memahami dan mencapai tujuan yang dimaksud terkait kebutuhan – kebutuhan *costumer* tersebut guna mengetahui lingkup dan batasan perangkat lunak yang akan dibangun.

b. *Planning (Estimating, Scheduling, Tracking)*

Tahapan selanjutnya adalah tahap perencanaan yang menjelaskan estimasi modul yang akan dikerjakan, risiko-risiko yang kemungkinan dapat terjadi, sumber daya yang diperlukan, produk yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan.

c. *Modeling (Analysis & Design)*

Pada tahap ini dilakukan perancangan dan permodelan perangkat lunak yang akan dibangun yang berfokus pada perancangan struktur data, aritektur perangkat lunak, *interface prototype*.

d. *Construction (Code & Test)*

Pada tahap *Construction* akan dilakukan implementasi *prototype* yang sudah dibuat sebelumnya kedalam koding. Setelah proses koding selesai, dilakukan pengujian terhadap perangkat lunak yang sudah dibuat.

e. *Deployment (Delivery, Support, Feedback)*

Jika semua tahapan sudah selesai dilakukan selanjutnya adalah tahap *Deployment*. *Deployment* merupakan tahapan implementasi perangkat lunak kepada *customer*, pemeliharaan perangkat lunak secara berkala, dan pengembangan perangkat lunak.

2.5.1 Kelebihan *Waterfall*

Model proses *waterfall* memiliki kelebihan dalam membangun dan mengembangkan suatu sistem atau perangkat lunak, yaitu [TUN14]:

1. Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
2. Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.

2.5.2 Kekurangan *Waterfall*

Dalam proses membangun dan mengembangkan suatu sistem, metode *waterfall* mempunyai beberapa kekurangan, antara lain [TUN14]:

1. Diperlukan manajemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
2. Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan.
3. Pelanggan sulit menyatakan kebutuhan secara eksplisit sehingga tidak dapat mengakomodasi ketidakpastian pada saat awal pengembangan.

2.6 Penelitian Terdahulu

Perangkat lunak yang dibangun merupakan pengembangan dari penelitian yang pernah ada. berikut dipaparkan beberapa penelitian yang pernah ada berkaitan dengan penyiraman tanaman, Daftar penelitian terdahulu dapat dilihat pada tabel 2.1

Tabel 2.1 Penelitian Terdahulu

No	Nama Peneliti	Judul penelitian	Hasil penelitian
1.	Wiradhika Erwins, 2017.	Pembangunan Perangkat Lunak Penyiraman Otomatis Berbasis Android	Menghasilkan sistem penyiraman tanaman otomatis berbasis android.
2.	Priyanto Sihno, 2013.	Purwarupa Sistem Penyiraman Tanaman Otomatis Berbasis Sensor Kelembaban Tanah Dan Arduino UNO	Menghasilkan sistem penyiraman tanaman otomatis berbasis sensor tanah dan arduino.
3.	Nuryadi Agus, 2015.	Prototipe Penyiraman Tanaman Otomatis Tanaman Cabai Berbasis Mikrokontroller ATMEGA16	Menghasilkan prototipe penyiraman tanaman cabai berbasis mikrokontroller ATMEGA16.
4.	Ratnawati, 2017.	Sistem Kendali Penyiram Tanaman Menggunakan <i>Propeller</i> Berbasis <i>Internet Of Things</i>	Menghasilkan sistem kendali penyiraman tanaman menggunakan <i>propeller</i> berbasis <i>Internet of Things</i> .
5.	Sari Weny Amanda, 2017.	Sistem Penyiraman Tanaman Otomatis Dengan Menggunakan Sensor Kelembaban Tanah Berbasis Arduino	Menghasilkan sistem penyiraman tanaman otomatis berbasis arduino.

Pada tabel dibawah ini menunjukkan kesimpulan dan perbandingan antara penelitian terdahulu dengan penelitian penyiraman tanaman otomatis berbasis *website*. perbedaan dari segi fitur dan teknologi yang digunakan akan dijelaskan pada tabel 2.2 berikut ini:

Tabel 2.2 fitur yang digunakan dalam penelitian

Judul penelitian	Menggunakan Web Server	Memanfaatkan dua sumber penyiraman	Menyiram tanaman secara otomatis	Menampilkan informasi kelembaban tanah	Menghubungkan perangkat lunak dengan perangkat keras
Pembangunan Perangkat Lunak Penyiraman Otomatis Berbasis Android	-	-	√	√	√
Purwarupa Sistem Penyiraman Tanaman Otomatis Berbasis Sensor Kelembaban Tanah Dan Arduino UNO	-	-	√	√	-
Prototipe Penyiraman Tanaman Otomatis Tanaman Cabai Berbasis Mikrokontroller ATMEGA16	-	-	√	-	-
Sistem Kendali Penyiram Tanaman Menggunakan <i>Propeller Of Things</i> Berbasis <i>Internet</i>	-	-	√	√	-
Sistem Penyiraman Tanaman Otomatis Dengan Menggunakan Sensor Kelembaban Tanah Berbasis Arduino	-	-	√	-	-
Pembangunan Perangkat Lunak Untuk Sistem Penyiraman Tanaman Berbasis Website	√	√	√	√	√

Perbandingan diatas menunjukkan bahwa penelitian pada pembangunan perangkat lunak penyiraman tanaman otomatis berbasis *website* ini memiliki ciri lain dari penelitian sebelumnya yang hanya memanfaatkan otomatisasi dalam penyiraman, akan tetapi pada penelitian pembangunan perangkat lunak untuk sistem penyiraman tanaman otomatis berbasis *website* ini memiliki dua fitur lain yaitu Menggunakan Web Server dan Memanfaatkan dua sumber penyiraman.

BAB 3
SKEMA PENELITIAN

Bab ini menjelaskan mengenai kerangka penulisan dari tugas akhir, rancangan penelitian, analisis relevansi solusi, analisis yang dilakukan terhadap kebutuhan dengan menggunakan pemodelan spesifikasi kebutuhan.

3.1. Alur Penelitian

Alur penelitian menjelaskan mengenai tahapan dan langkah - langkah pengerjaan Tugas Akhir. Dalam pembuatan Tugas Akhir yang dibuat, Alur penelitian dilandasi oleh tiga komponen, yaitu :

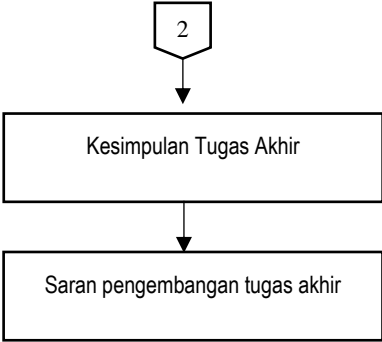
1. Tahap, adalah komponen yang berisi gambaran yang akan dikerjakan dalam Tugas Akhir.
2. Langkah, adalah komponen yang berisi hal-hal yang akan dikerjakan dalam Tugas Akhir yang disesuaikan dengan metode yang digunakan.
3. Literatur atau Referensi, adalah komponen yang berisi sumber pengetahuan yang dijadikan landasan dalam mengerjakan Tugas Akhir.

Berikut alur penelitian akan dipaparkan pada tabel 3.1

Tabel 3.1 Kerangka Tugas Akhir

Tahap & Hasil	Langkah Penelitian	Literatur & Referensi
<p>Tahap 1 : Studi Literatur</p> <p>Hasil: Teori yang digunakan guna mendukung dalam penelitian tugas akhir.</p> <p>Kontribusi: Berguna untuk tahap analisis dan perancangan perangkat lunak.</p>	<pre> graph TD A[Menganalisis penelitian terdahulu.] --> B[Menentukan topik tugas akhir.] B --> C[Mengumpulkan data yang relevan terkait tugas akhir dari buku, e-book, jurnal, dan hasil penelitian orang lain.] C --> D[Gagasan perangkat lunak yang akan dibangun] D --> E{1} </pre>	<ol style="list-style-type: none"> 1. Prototipe Penyiraman Tanaman Otomatis Tanaman Cabai Berbasis Mikrokontroler ATMEGA16 [NUR15]. 2. Software Engineering A Practicioner’s Approach Eighth Edition [PRE15]. 3. Purwarupa Sistem Penyiraman Tanaman Otomatis Berbasis Sensor Kelembaban Tanah Dan Arduino UNO [PRI13]. 4. Sistem Penyiraman Tanaman Otomatis Dengan Menggunakan Sensor Kelembaban Tanah Berbasis Arduino [SAR17]. 5. Sistem Kendali Penyiram Tanaman Menggunakan Propeller Berbasis Internet Of Things [RAT17]. 6. Pembangunan Perangkat Lunak Penyiraman Tanaman Secara Otomatis Berbasis Android [WIR17].

<p>Tahap 2 : Analisis dan perancangan</p> <p>Hasil : Perancangan perangkat lunak dan kebutuhan.</p> <p>Kontribusi : Berguna untuk tahap implementasi.</p>		<ol style="list-style-type: none"> 1. Software Engineering A Practicioner’s Approach Eighth Edition [PRE15]. 2. Pembangunan Perangkat Lunak Penyiraman Tanaman Secara Otomatis Berbasis Android [WIR17].
<p>Tahap 3 : Implementasi Perangkat Lunak</p> <p>Hasil : Perangkat Lunak.</p> <p>Kontribusi : Berguna untuk tahapan pengujian perangkat lunak.</p>		<ol style="list-style-type: none"> 1. Software Engineering A Practicioner’s Approach Eighth Edition [PRE15]. 2. Pembangunan Perangkat Lunak Penyiraman Tanaman Secara Otomatis Berbasis Android [WIR17].
<p>Tahap 4 : Pengujian Perangkat Lunak.</p> <p>Hasil : Perangkat lunak yang telah diuji.</p> <p>Kontribusi : -</p>		<ol style="list-style-type: none"> 1. Software Engineering A Practicioner’s Approach Eighth Edition [PRE15].

<p>Tahap 5 : Kesimpulan dan saran</p> <p>Hasil : Kesimpulan dan saran mengenai tugas akhir yang telah dilakukan.</p> <p>Kontribusi : -</p>		<p>1. Software Engineering A Practitioner's Approach Eighth Edition [PRE15].</p>
---	---	--

3.2. Analisis Masalah dan Solusi Tugas Akhir

Masalah dan solusi Tugas Akhir menjelaskan mengenai masalah serta solusi pada pembuatan perangkat lunak untuk sistem penyiraman berbasis *website*. Gambaran *Fishbone* akan dipaparkan pada Gambar 3.1

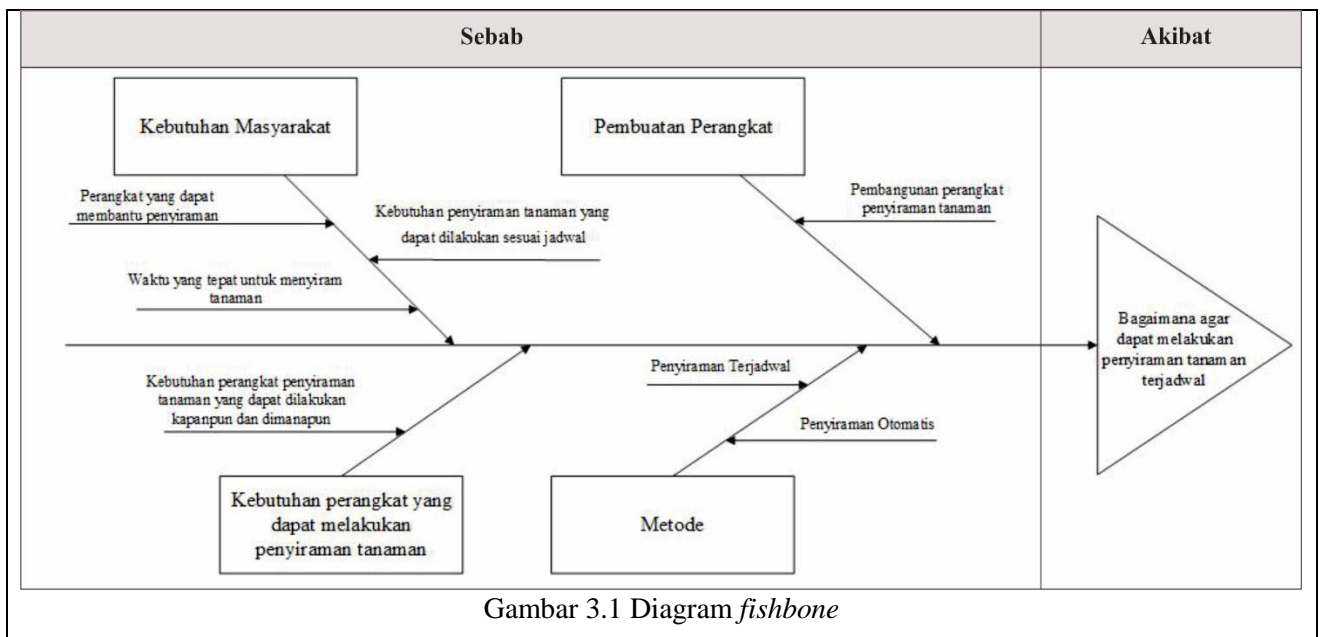


Diagram *fishbone* digunakan untuk menjabarkan faktor-faktor permasalahan yang terjadi. Dari faktor-faktor tersebut, penulis dapat menemukan akar dari masalah yang sedang dihadapi. Permasalahan yang penulis hadapi ketika menyusun tugas akhir ini adalah mengidentifikasi *bad smells code*, dari permasalahan tersebut penulis mencoba memetakan faktor-faktor serta pemicu apa saja yang terdapat pada permasalahan ini. Dari hasil pemetaan tersebut penulis dapat mendapatkan gambaran untuk menyelesaikan permasalahan yang ada dengan mempertimbangkan faktor-faktor dan pemicu yang terjadi yaitu bagaimana agar dapat melakukan penyiraman tanaman terjadwal.

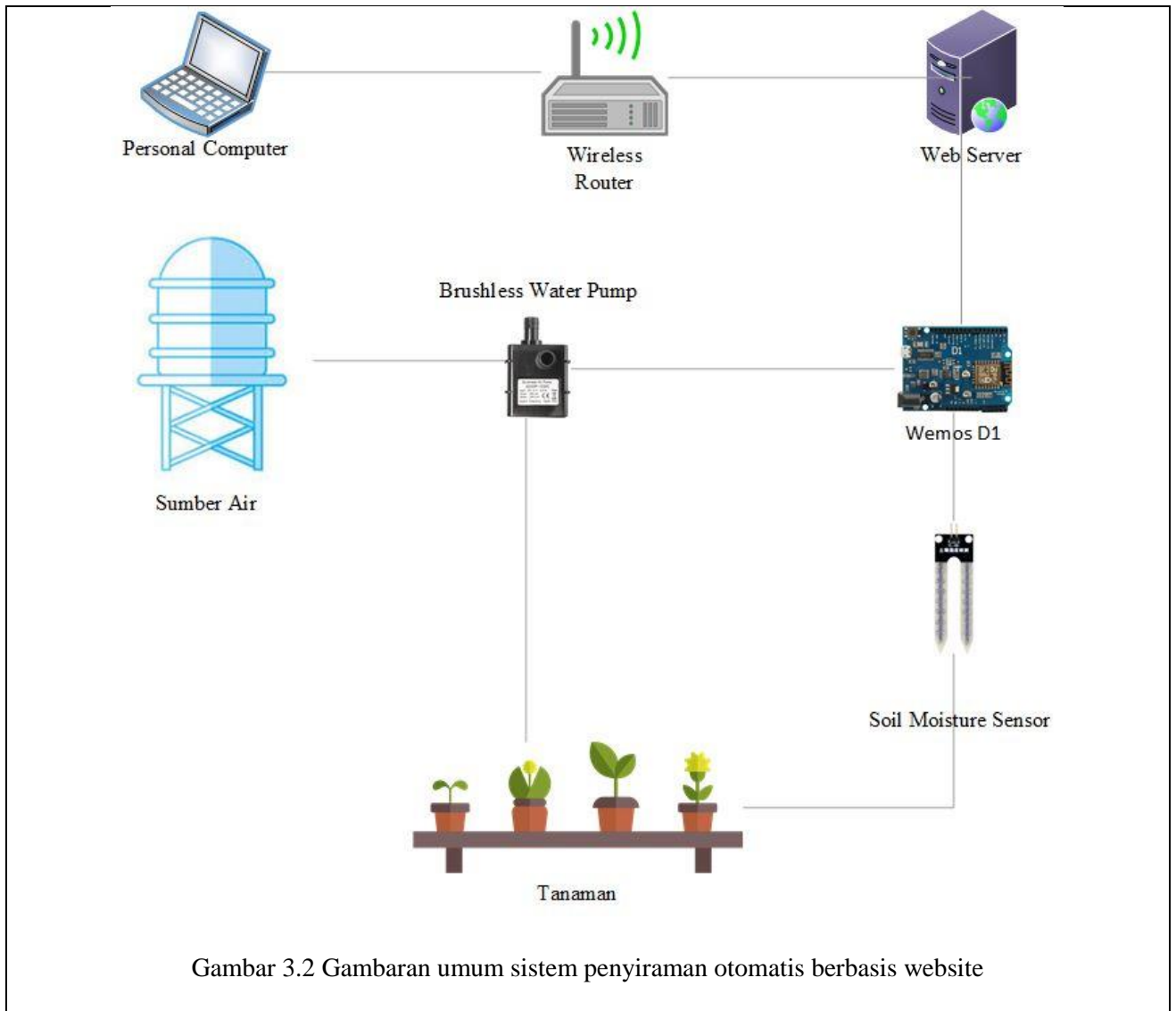
3.2.1. Analisis Masalah

Berdasarkan diagram pada gambar 3.1 diatas dikemukakan bahwa masalah yang ada dalam pembangunan perangkat lunak ini ada pada sebuah kebutuhan masyarakat, kebutuhan perangkat yang dapat melakukan penyiraman tanaman, pembuatan perangkat, dan metode, yang di mana pada waktu penyiraman yang tepat terdapat masalah yaitu masyarakat tidak memiliki waktu untuk melakukan penyiraman, kebutuhan perangkat penyiraman tanaman yang dapat melakukan penyiraman yaitu dapat dilakukan kapanpun dan di mana pun, pada sebuah pembuatan perangkat terdapat masalah pada pembangunan perangkat penyiraman tanaman, dan pada metode terdapat penyiraman terjadwal dan penyiraman otomatis.

3.2.2. Analisis Relevansi Solusi

Penyiraman yang dilakukan oleh pemilik tanaman pada umumnya dapat dilakukan tanpa harus menggunakan perangkat lunak yang diusulkan, akan tetapi dalam pelaksanaannya terdapat beberapa kendala yang ditemui seperti waktu luang untuk menyiram sulit ditemukan, kesibukan pemilik tanaman seperti bekerja di kantor, berpergian ke luar kota, dan kesibukan lainnya yang menyebabkan tidak memiliki waktu luang untuk menyiram sehingga dapat menyebabkan kematian pada tanaman. Dari kendala yang terdapat pada penyiraman diusulkan solusi untuk permasalahan tersebut, dengan mengusulkan perangkat lunak untuk sistem penyiraman tanaman otomatis yang berfungsi untuk membantu pemilik tanaman dalam melakukan penyiraman di waktu-waktu tertentu tanpa harus campur tangan langsung dengan alat siram atau tanaman.

Perangkat lunak untuk sistem penyiraman tanaman otomatis merupakan sistem yang menggabungkan perangkat lunak (*software*) dengan perangkat keras (*hardware*) dengan tujuan sebagai alat bantu penyiraman yang dapat dioperasikan dari jarak jauh. Perangkat lunak terdiri dari beberapa komponen yaitu *web server* dan *web service* sedangkan pada perangkat keras terdapat papan sirkuit arduino. *Web server* berguna untuk memerintahkan perangkat keras melalui *web service*, lalu *web service* itu sendiri berguna untuk menghubungkan perangkat lunak dan perangkat keras, sedangkan perangkat keras berfungsi sebagai alat penggerak alat siram. Gambaran umum dari perangkat lunak penyiraman tanaman akan dipaparkan pada Gambar 3.2 di bawah:



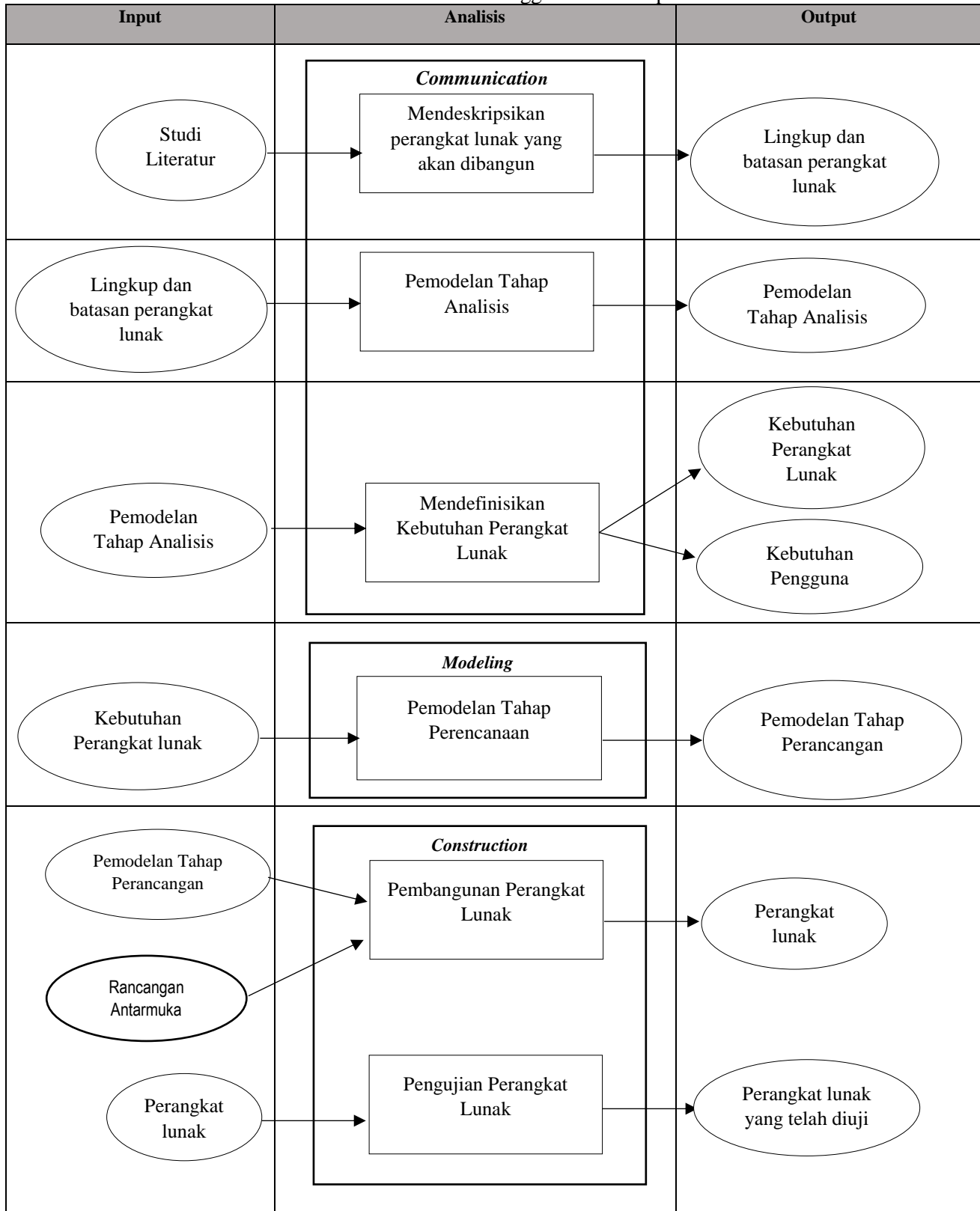
3.3. Kerangka Berpikir Teoritis

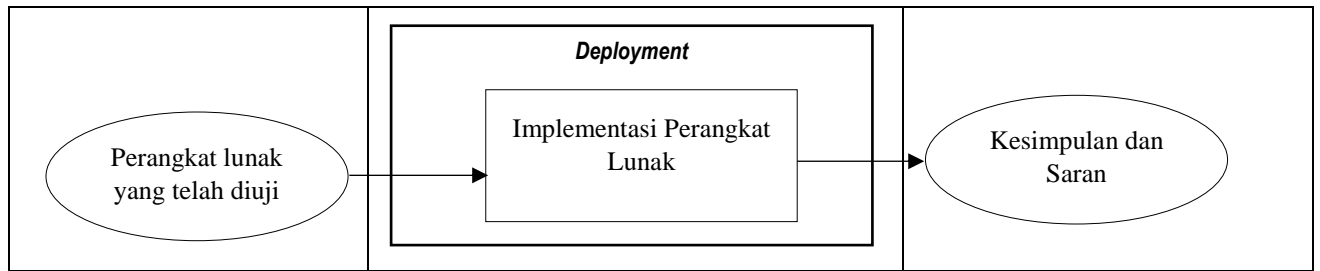
Tujuan dari tugas akhir ini adalah untuk membangun perangkat lunak untuk sistem penyiraman tanaman otomatis berbasis *website* sebagai alat bantu yang dapat menyiram tanaman otomatis. Adapun beberapa faktor yang mempengaruhi pembuatan perangkat lunak penyiraman tanaman otomatis adalah sebagai berikut :

3.3.1. Penggunaan Konsep

Penggunaan konsep menggambarkan tahapan-tahapan yang memunculkan input dan output yang dihasilkan dari sebuah proses analisis yang dilakukan. Berikut ini Analisis Penggunaan Konsep akan dipaparkan pada tabel 3.2

Tabel 3.2 Analisis Penggunaan Konsep





Berikut ini merupakan tabel Penjelasan Analisis Penggunaan Konsep dari skema analisis yang sudah dibuat akan dipaparkan pada tabel 3.3

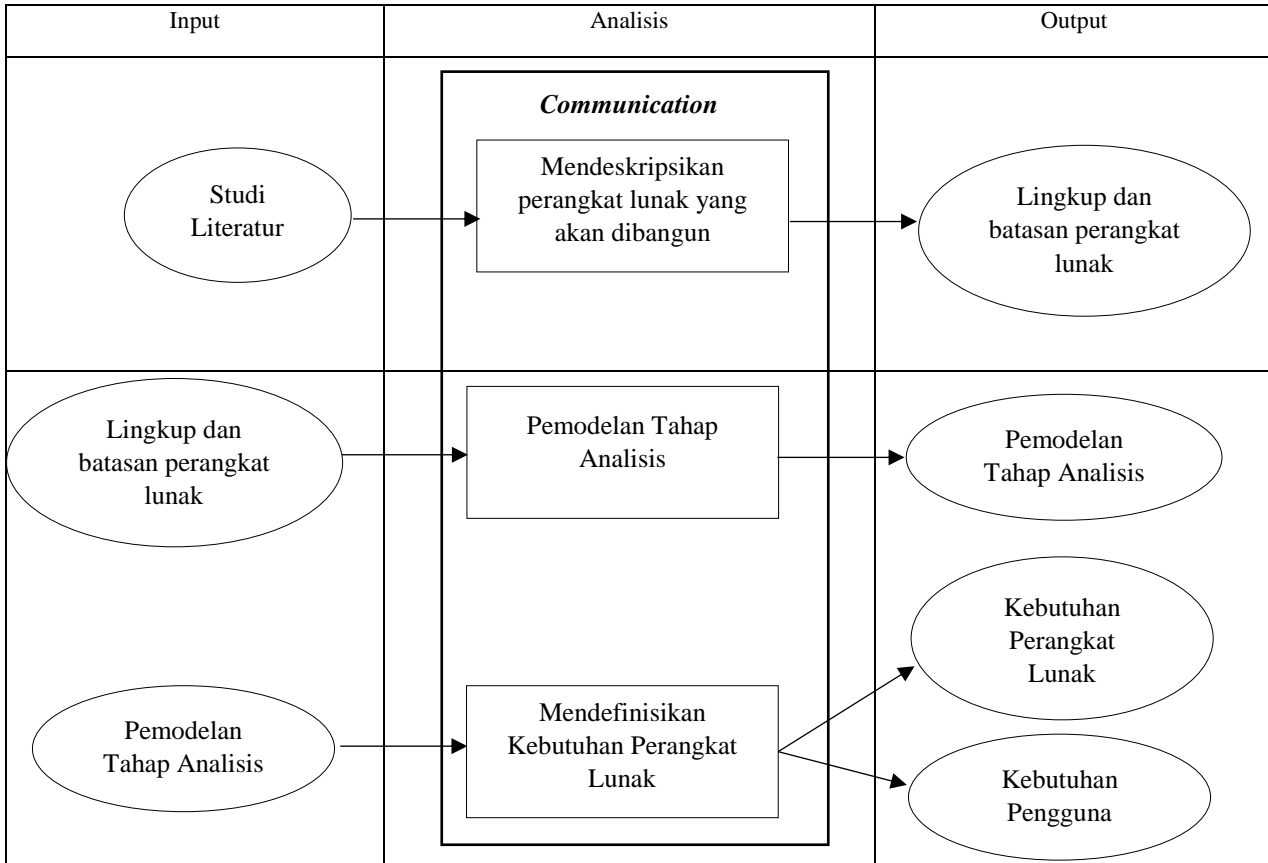
Tabel 3.3 Penjelasan Analisis Penggunaan Konsep

Input	Langkah Analisis	Hasil Analisis
a. Studi literatur	Mendeskripsikan perangkat lunak yang akan dibangun. Langkah ini merupakan langkah untuk mendeskripsikan seperti apa perangkat lunak yang akan dibangun.	a. Lingkup dan batasan perangkat lunak
a. Lingkup dan batasan perangkat lunak	Pemodelan tahap analisis. Langkah ini merupakan langkah untuk menentukan lingkup dan batasan perangkat lunak.	a. Pemodelan tahap analisis
a. Pemodelan tahap analisis	Mendefinisikan kebutuhan perangkat lunak. Langkah ini merupakan langkah untuk menentukan kebutuhan-kebutuhan untuk menunjang pembangunan perangkat lunak, diantaranya kebutuhan perangkat lunak dan kebutuhan pengguna.	a. Kebutuhan perangkat lunak b. Kebutuhan pengguna
a. Kebutuhan perangkat lunak	Pemodelan tahap perancangan. Langkah ini merupakan langkah untuk merepresentasikan rancangan arsitektur perangkat lunak yang dibangun mulai dari rancangan antarmuka hingga interaksi proses yang berjalan di dalam perangkat lunak berdasarkan rancangan tahap analisis yang telah dilakukan.	a. Pemodelan tahap perancangan b. Rancangan antarmuka
a. Pemodelan tahap perancangan b. Rancangan antarmuka	Pembangunan perangkat lunak. Langkah ini merupakan langkah untuk membangun perangkat lunak. Pembangunan perangkat lunak mengacu kepada rancangan dan analisis yang telah dilakukan sebelumnya, sehingga perangkat yang akan dibangun sesuai dengan kebutuhan.	a. Perangkat lunak
a. Perangkat lunak	Pengujian perangkat lunak. Langkah ini merupakan langkah pengujian perangkat lunak. Perangkat lunak diuji apakah telah berfungsi sesuai yang ditetapkan, pengujian bertujuan untuk memastikan bahwa perangkat lunak dapat digunakan tanpa adanya <i>error</i> .	a. Perangkat lunak yang telah diuji
a. Perangkat lunak yang telah diuji	Implementasi perangkat lunak. Tahap ini merupakan tahap implementasi perangkat lunak kepada pengguna untuk digunakan sesuai fungsi perangkat lunak. Pada tahap ini akan diangkat kesimpulan dan saran pengembangan perangkat lunak.	a. Kesimpulan dan saran

3.3.2. Peta Analisis

Peta analisis merepresentasikan proses-proses analisis yang dilakukan, input, dan output dari proses analisis yang terdapat pada konsep pembangunan perangkat lunak yang digunakan yaitu *Waterfall*. Peta Analisis dapat dilihat pada tabel 3.4 di bawah ini

Tabel 3.4 Peta Analisis



Berikut ini merupakan tabel penjelasan dari skema analisis yang sudah dibuat akan dipaparkan melalui Langkah Analisis pada tabel 3.5

Tabel 3.5 Langkah Analisis

Input	Langkah Analisis	Hasil Analisis
a. Studi Literatur	<p>Mendeskripsikan perangkat lunak yang akan dibangun. Langkah ini merupakan langkah untuk mendeskripsikan seperti apa perangkat lunak yang akan dibangun.</p>	a. Lingkup dan batasan perangkat lunak
b. Lingkup dan batasan perangkat lunak	<p>Pemodelan tahap analisis. Langkah ini merupakan langkah untuk menentukan lingkup dan batasan perangkat lunak.</p>	a. Pemodelan tahap analisis

Input	Langkah Analisis	Hasil Analisis
c. Pemodelan Tahap Analisis	Mendefinisikan kebutuhan perangkat lunak. Langkah ini merupakan langkah untuk menentukan kebutuhan-kebutuhan untuk menunjang pembangunan perangkat lunak, diantaranya kebutuhan perangkat lunak dan kebutuhan pengguna.	a. Kebutuhan Perangkat Lunak b. Kebutuhan Pengguna

3.4. Profile Tempat dan Objek Penelitian

Profil studi kasus meliputi tempat penelitian tugas akhir di sebuah rumah tinggal yang memiliki taman berskala kecil dengan tanaman berwadah pot. Pada umumnya pemilik tanaman yang tinggal di kompleks atau perumahan dan bekerja dari pagi hingga sore jika tidak memiliki pembantu tanaman yang dimilikinya sulit untuk dirawat dengan baik. Waktu terbaik untuk menyiram tanaman adalah pagi hari, karena akan memberi tanaman waktu untuk mengering sebelum malam hari, namun pada waktu yang tepat tersebut pemilik tanaman tidak dapat melakukan penyiraman. Membiarkan air mendiami tanaman semalaman dapat memicu pertumbuhan jamur. Demi tercapainya tujuan untuk merawat tanaman tersebut melakukan penyiraman di saat yang tepat dan menggunakan metode yang tepat akan menjaga tanaman tetap sehat

BAB 4 ANALISIS DAN PERANCANGAN

Bab ini menjelaskan mengenai penggambaran kebutuhan perangkat lunak, kemudian melakukan analisis terhadap kebutuhan tersebut hingga perancangan terhadap perangkat lunak yang sesuai dengan kebutuhan awal dan hasil dari analisis.

4.1. Analisis Perangkat Lunak

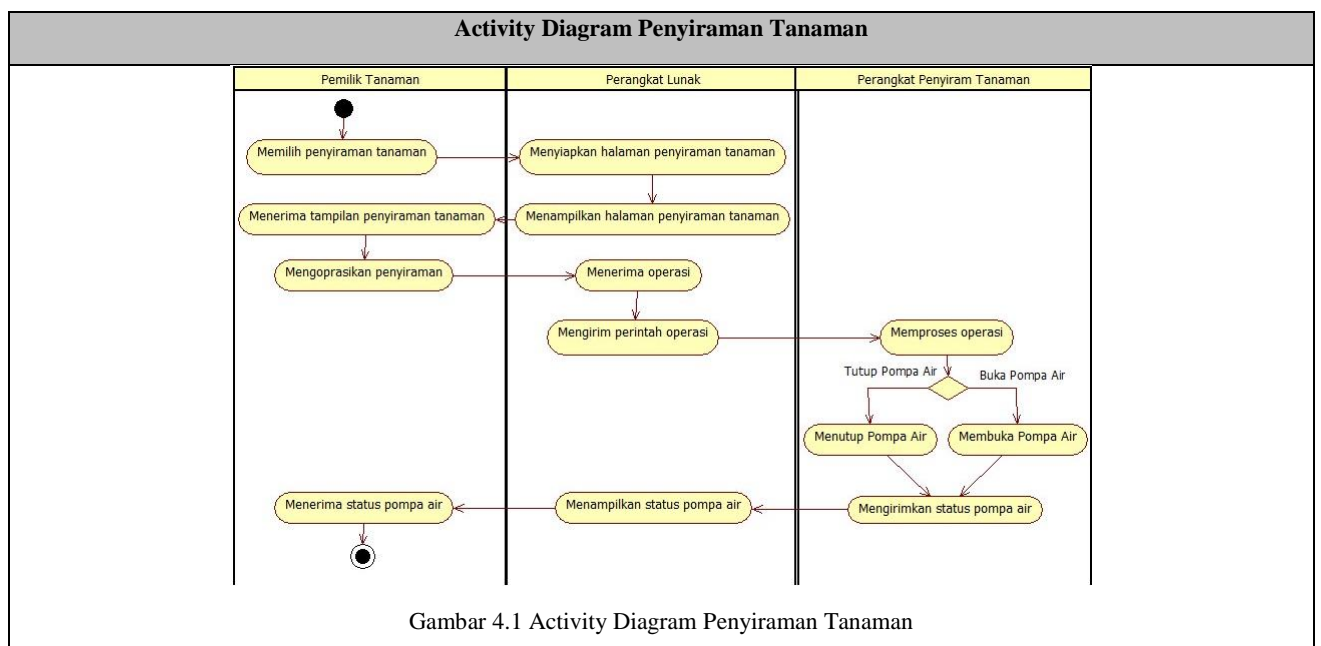
Berikut ini akan dijelaskan mengenai analisis dan perangkat lunak yang akan dibangun pada penelitian tugas akhir. Analisis memiliki tujuan untuk mendapatkan kebutuhan dari sistem kemudian memodelkannya dengan menggunakan diagram-diagram. Analisis ini berguna untuk tahap perancangan.

4.1.1. Model Bisnis

Model bisnis merupakan fase analisis untuk menentukan lingkup pembangunan perangkat lunak yang akan dibangun. Lingkup itu sendiri terdiri dari proses bisnis yang terkait dengan pembangunan perangkat lunak, aktivitas-aktivitas yang terjadi di dalam proses bisnis dengan aktor bisnis hingga menghasilkan kebutuhan perangkat lunak.

4.1.1.1. Activity Diagram

Activity diagram merepresentasikan alur aktivitas antara aktor dengan sistem yang bertujuan untuk menjelaskan rangkaian aktivitas, proses dan pengulangan yang terjadi. Berikut ini merupakan *activity diagram* penyiraman tanaman akan dipaparkan pada gambar 4.1



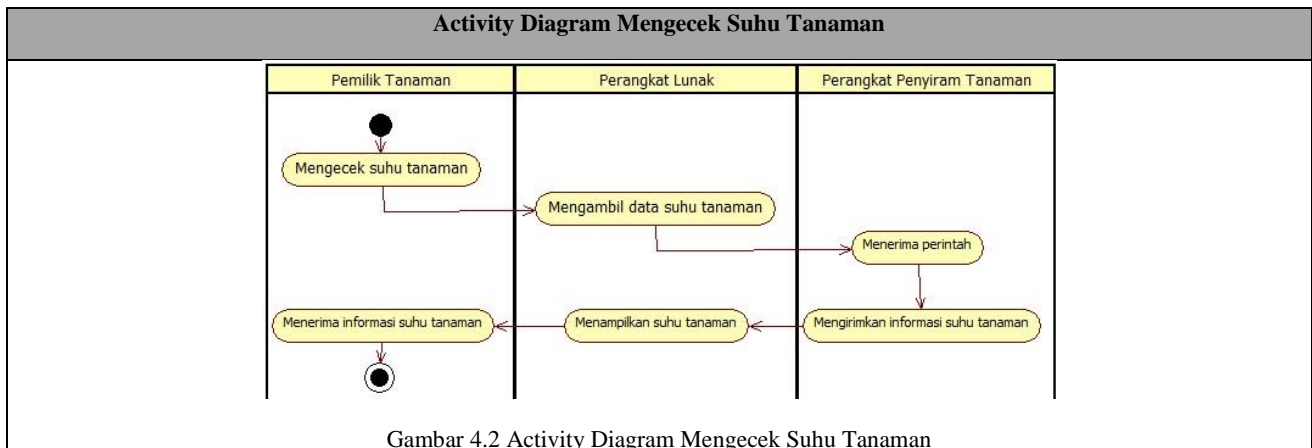
Gambar 4.1 Activity Diagram Penyiraman Tanaman

Deskripsi

Diagram aktivitas di atas menggambarkan alur untuk menyiram tanaman menggunakan perangkat lunak penyiraman tanaman otomatis, pemilik tanaman dapat melakukan penyiraman seolah-olah seperti membuka/menutup pompa air yang diganti menggunakan

tombol pada perangkat lunak. Aktivitas ini dimulai dari pemilik tanaman yang mengoperasikan penyiraman menggunakan perangkat lunak lalu sistem akan mengirim perintah operasi kepada perangkat penyiram tanaman kemudian pompa air akan dibuka atau ditutup sesuai perintah.

Berikut ini merupakan *activity diagram* mengecek suhu tanaman akan dipaparkan pada gambar 4.2

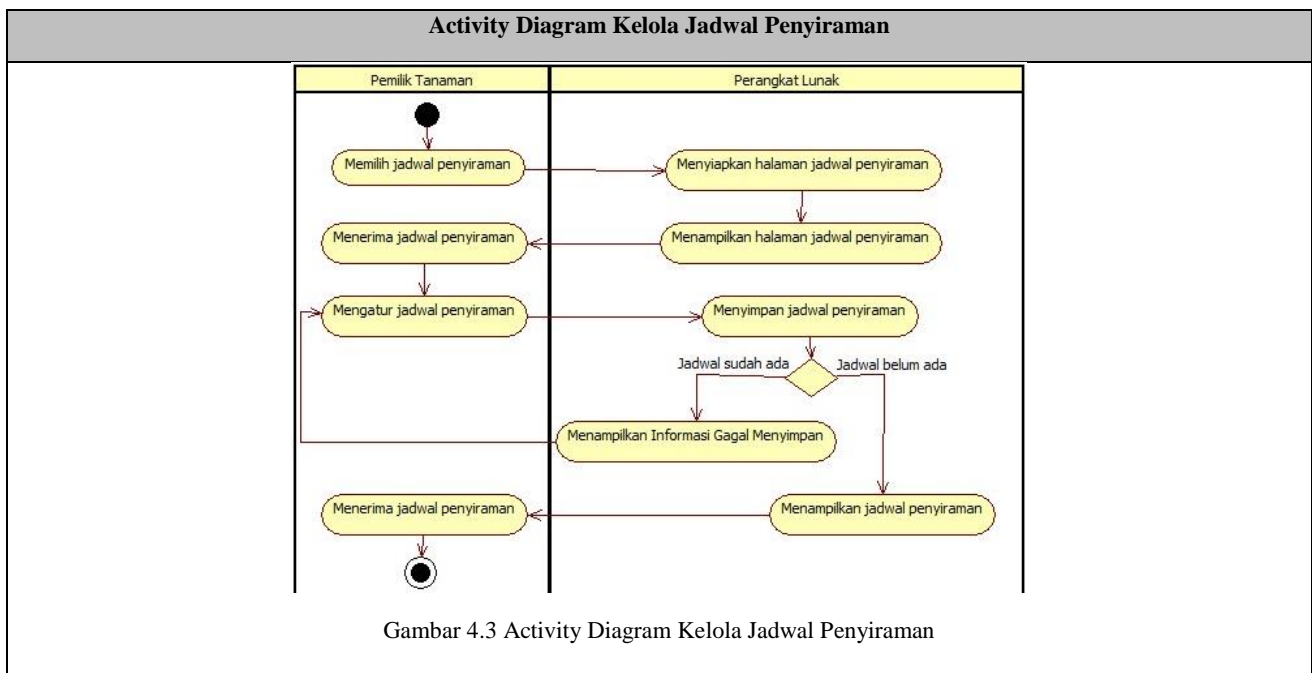


Gambar 4.2 Activity Diagram Mengecek Suhu Tanaman

Deskripsi

Diagram aktivitas di atas menggambarkan alur untuk mengecek suhu tanaman menggunakan perangkat lunak penyiraman tanaman otomatis, pemilik tanaman akan mendapatkan informasi suhu tanaman lembab/kering melalui sensor yang terdapat pada tanaman lalu dikirimkan kepada sistem.

Berikut ini merupakan *activity diagram* kelola jadwal penyiraman akan dipaparkan pada gambar 4.3



Gambar 4.3 Activity Diagram Kelola Jadwal Penyiraman

Deskripsi

Diagram aktivitas di atas menggambarkan alur untuk mengelola jadwal tanaman menggunakan perangkat lunak penyiraman tanaman otomatis, pemilik tanaman dapat menentukan jadwal penyiraman sesuai keinginan lalu penyiraman akan dilakukan oleh sistem sesuai

jadwal yang telah ditentukan. Aktivitas ini dimulai dari permintaan pemilik tanaman yang menyimpan jadwal pada perangkat lunak lalu setelah dijadwalkan penyiraman akan dilakukan oleh sistem sesuai jadwal.

Berikut ini merupakan *activity diagram* menyiram menurut jadwal akan dipaparkan pada gambar 4.4

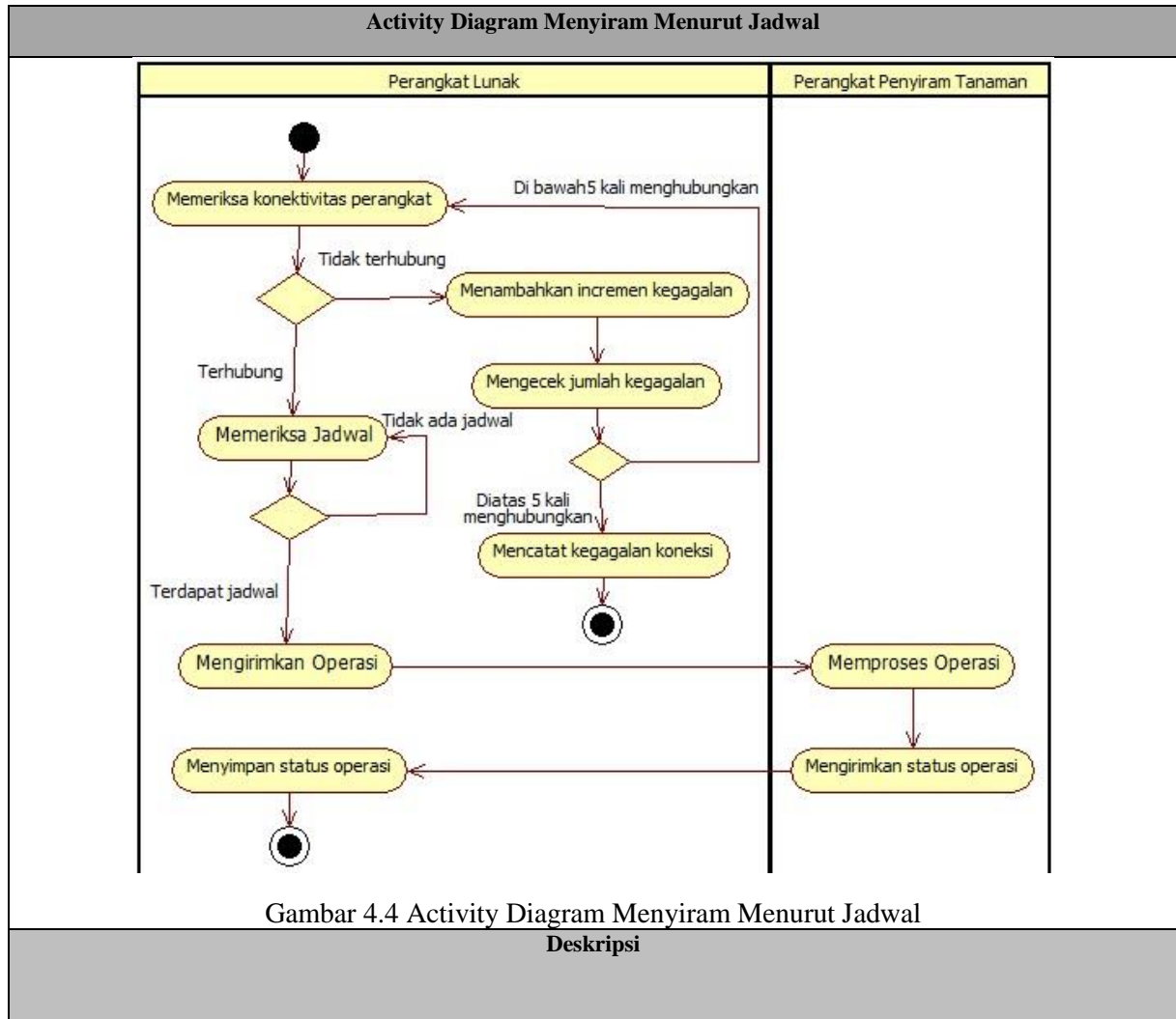


Diagram aktivitas di atas menggambarkan alur sistem untuk melakukan penyiraman tanaman sesuai jadwal yang telah dibuat oleh pemilik tanaman. Aktivitas ini dimulai dari sistem mengecek terlebih dahulu apakah terdapat perangkat penyiram tanaman yang sudah terhubung dengan perangkat lunak atau belum. Jika terdapat perangkat penyiraman tanaman yang terhubung maka sistem melakukan pengecekan jadwal penyiraman, apabila terdapat jadwal maka perangkat lunak akan memerintahkan perangkat penyiraman melakukan operasi sesuai perintah. Setelah operasi dilakukan, perangkat penyiram tanaman akan mengirimkan status kepada perangkat lunak.

4.1.1.1 User Requirement

User requirements merupakan langkah menentukan kebutuhan-kebutuhan untuk menunjang pembangunan perangkat lunak dari segi kebutuhan pengguna. Pada tahap sebelumnya sudah dijelaskan mengenai aktivitas-aktivitas antara pengguna dengan sistem, beberapa aktivitas tersebut dapat disimpulkan menjadi kebutuhan pengguna, selanjutnya akan dipaparkan pada tabel 4.1 di bawah ini:

Tabel 4.1 User Requirement

No.	Kode User Requirement	Aktivitas	Deskripsi
1.	UR-01	Mengoprasikan penyiraman.	Pengguna membutuhkan perangkat lunak untuk dapat mengoprasikan penyiraman tanaman.
2.	UR-02	Mengecek suhu tanaman.	Pengguna membutuhkan perangkat lunak untuk dapat mengetahui informasi kelembaban.
3.	UR-03	Mengatur jadwal penyiraman.	Pengguna membutuhkan perangkat lunak untuk dapat menjadwalkan penyiraman tanaman.

4.1.2 Model Use Case

Model *Use Case* merupakan fase analisis untuk menentukan kebutuhan perangkat lunak yang terdiri dari kebutuhan fungsionalitas dan kebutuhan non fungsionalitas dari perangkat lunak yang akan dibangun. Selain melakukan analisis kebutuhan-kebutuhan tersebut, pada fase ini juga menentukan skenario perangkat lunak, mengidentifikasi kelas yang digunakan beserta interaksi antar kelas dan mengidentifikasi proses yang terjadi di dalam interaksi antar kelas.

4.1.2.1 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak pada penelitian ini berfokus pada kebutuhan fungsionalitas dan kebutuhan non fungsionalitas. Berikut ini akan dijelaskan mengenai kebutuhan tersebut.

4.1.2.1.1 Kebutuhan Fungsionalitas

Kebutuhan fungsionalitas perangkat lunak terdiri dari beberapa fungsi utama perangkat lunak yang saling berhubungan dan mendukung satu sama lain. Berikut ini pada tabel 4.2 akan dijelaskan kebutuhan fungsionalitas dari perangkat lunak penyiraman tanaman:

Tabel 4.2 Kebutuhan Fungsionalitas

No	Kode Kebutuhan Fungsionalitas	Kode User Requirement	Kebutuhan Fungsionalitas	Deskripsi
1	KF-01	UR-01	Perangkat lunak harus dapat mengoprasikan penyiraman.	Perangkat lunak harus dapat menyiram tanaman, sehingga dibutuhkan fungsionalitas untuk mengoperasikan penyiraman tanaman.

No	Kode Kebutuhan Fungsionalitas	Kode User Requirement	Kebutuhan Fungsionalitas	Deskripsi
2	KF-02		Perangkat penyiraman tanaman dapat menyiram dari dua sumber air	Perangkat penyiram tanaman harus dapat menyiram dari dua sumber air, sehingga dapat memperluas jarak penyiraman tanaman.
3	KF-03	UR-02	Perangkat lunak harus dapat mengecek suhu tanaman.	Perangkat lunak harus dapat mengetahui informasi kelembaban pada tanaman, sehingga dibutuhkan fungsionalitas untuk mengecek suhu tanaman.
4	KF-04	UR-03	Perangkat lunak harus dapat mengatur jadwal penyiraman.	Perangkat lunak harus dapat mengatur jadwal penyiraman tanaman, sehingga dibutuhkan fungsionalitas untuk mengatur jadwal penyiraman tanaman.
5	KF-05		Perangkat lunak harus dapat menyiram tanaman sesuai jadwal yang telah ditentukan.	Perangkat lunak harus dapat menyiram tanaman sesuai jadwal yang telah ditentukan, sehingga dibutuhkan fungsionalitas untuk menyiram sesuai jadwal.

4.1.2.1.2 Kebutuhan Non Fungsionalitas

Kebutuhan non fungsionalitas perangkat lunak merupakan kebutuhan yang ditujukan untuk mendukung fungsi utama dari perangkat lunak. Berikut ini pada tabel 4.3 akan dipaparkan kebutuhan non fungsionalitas dari perangkat lunak penyiraman tanaman.

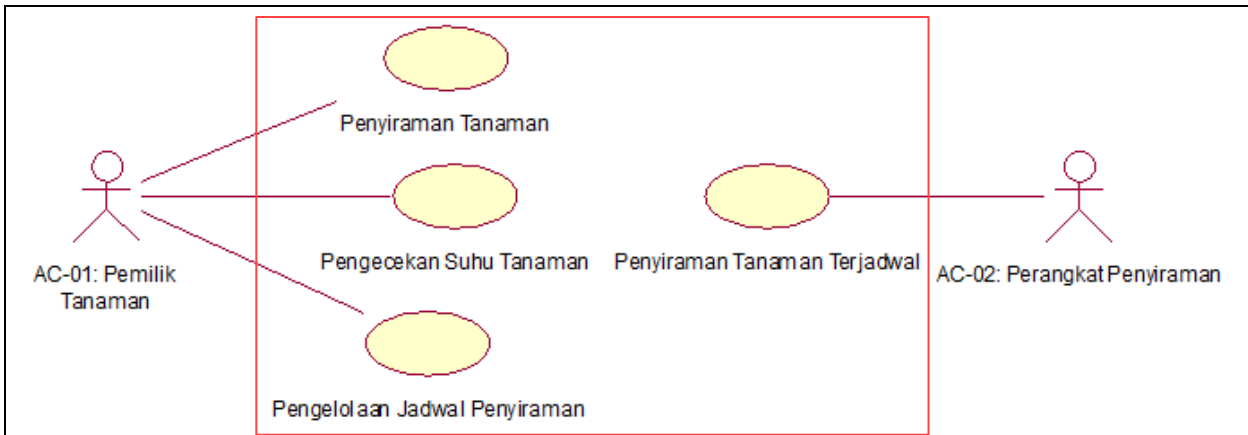
Tabel 4.3 Kebutuhan Non Fungsionalitas

No	Parameter	Keterangan
1	Availability	Perangkat lunak harus dapat digunakan selama 24 jam, kecuali tidak terdapatnya koneksi internet.
2	Usability	Perangkat lunak harus mudah digunakan.
3	Testability	Perangkat lunak telah diuji.

4.1.2.2 Use Case Diagram

Use Case adalah teknik untuk menangkap kebutuhan fungsional dari sistem. Use Case bekerja dengan menggambarkan interaksi antara pengguna dari sistem dan sistem itu sendiri, memberikan narasi

tentang bagaimana sistem yang digunakan. Berikut ini pada gambar 4.5 akan ditampilkan Use Case Diagram dari perangkat lunak yang akan dibangun.



Gambar 4.5 Use Case Diagram

Deskripsi

Diagram aktivitas di atas menggambarkan interaksi aktor dengan sistem penyiraman tanaman otomatis. Perangkat lunak penyiraman tanaman otomatis ini terdapat 4 fungsionalitas utama yang terdiri dari Penyiraman Tanaman untuk mengoperasikan perangkat penyiram, Pengelolaan Jadwal Penyiraman tanaman, Pengecekan Suhu Tanaman untuk mengetahui kelembaban tanaman, Pengaturan relay untuk memperluas jangkauan siram, peran Perangkat penyiraman yaitu melakukan Penyiraman Tanaman Terjadwal.

4.1.2.2.1 Definisi Aktor

Aktor merupakan seseorang yang berinteraksi dengan perangkat lunak. Berikut ini akan dipaparkan aktor yang terlibat dengan perangkat lunak penyiraman tanaman otomatis pada tabel 4.4 di bawah ini.

Tabel 4.4 Definisi Aktor

No	Aktor	Deskripsi
UA-01	Pemilik Tanaman	Pemilik tanaman merupakan aktor pengguna perangkat lunak penyiraman tanaman otomatis.
UA-02	Perangkat Penyiraman	Perangkat penyiraman merupakan aktor yang berperan sebagai penggerak perangkat penyiraman tanaman.

4.1.2.2.2 Definisi Use Case

Pada bagian ini akan dijelaskan hubungan antara kebutuhan fungsional dan diagram Use Case, yang akan direpresentasikan kedalam bentuk tabel definisi Use Case pada tabel 4.5 di bawah ini.

Tabel 4.5 Definisi Use Case

No Use Case	Use Case	Deskripsi	ID Requirement
UC-01	Penyiraman tanaman	Perangkat lunak dapat menyiram tanaman	KF-01
UC-02	Pengecekan suhu tanaman	Perangkat lunak dapat mengecek suhu tanaman	KF-03

No Use Case	Use Case	Deskripsi	ID Requirement
UC-03	Pengelolaan jadwal penyiraman	Perangkat lunak dapat mengatur jadwal penyiraman	KF-04
UC-04	Penyiraman tanaman terjadwal	Perangkat penyiraman tanaman dapat menyiram tanaman sesuai jadwal.	KF-05

4.1.2.3 Skenario Use Case

Skenario *use case* digunakan untuk mendetailkan setiap *use case* yang terdapat pada diagram *use case*. Skenario *use case* akan diberikan uraian nama aktor yang berhubungan dengan *use case*, tujuan dari *use case*, deskripsi global tentang *use case*, pra-kondisi yang harus dipenuhi pasca-kondisi yang diharapkan setelah berjalannya fungsional *use case*. Berikut ini adalah skenario dari *use case* perangkat lunak.

Tabel 4.6 Skenario use case Penyiraman Tanaman

Identifikasi	
No.	: UC-01
Nama	: Penyiraman Tanaman
Tujuan	: Untuk melakukan penyiraman tanaman KF-01
Tipe	: High Level
Aktor	: Pemilik Tanaman
Skenario Utama	
Kondisi Awal	: Perangkat lunak menampilkan halaman penyiraman tanaman.
Aksi Aktor	Reaksi Sistem
1. Menekan tombol penyiraman tanaman	2. Menampilkan halaman penyiraman tanaman
3. Mengubah status keran ON/OFF	4. Mengirim perintah buka/tutup keran kepada perangkat penyiraman
	5. Mengecek perintah
	6. Memeriksa konektivitas perangkat penyiraman
	7. Menerima perintah penyiraman
Kondisi Akhir: Perangkat lunak berhasil melakukan penyiraman tanaman.	
Skenario Alternatif	
Kondisi	: Perangkat lunak tidak terhubung dengan perangkat penyiraman.
Aksi Aktor	Reaksi Sistem
	1. Perangkat lunak menampilkan pesan perangkat penyiraman tidak terhubung.
	2. Menampilkan halaman penyiraman tanaman
Kondisi Akhir : Perangkat lunak menampilkan halaman penyiraman tanaman dan pemberitahuan perangkat penyiraman tidak terhubung.	

Tabel 4.7 Skenario use case Pengecekan Suhu Tanaman.

Identifikasi	
No.	: UC-02
Nama	: Pengecekan Suhu Tanaman.
Tujuan	: Menerima status suhu tanaman KF-03
Tipe	: High Level
Aktor	: Pemilik Tanaman
Skenario Utama	
Kondisi Awal	: Sensor kelembaban mendeteksi suhu tanaman dan perangkat lunak menampilkan dasbor admin
Aksi Aktor	Reaksi Sistem
	1. Mendeteksi kelembaban tanaman
	2. Mengirimkan status kelembaban tanaman
	3. Menampilkan status kelembaban tanaman
4. Menerima status kelembaban tanaman	
Kondisi Akhir: Perangkat lunak menampilkan status kelembaban tanaman	
Skenario Alternatif	
Kondisi	: Perangkat penyiraman tidak terhubung
Aksi Aktor	Reaksi Sistem
	1. Perangkat lunak menampilkan pesan perangkat penyiraman tidak terhubung.
	2. Menampilkan dasbor admin
Kondisi Akhir : Perangkat lunak menampilkan dasbor admin dan menampilkan pesan tidak terhubung	

Tabel 4.8 Skenario use case Pengelolaan Jadwal Penyiraman

Identifikasi	
No.	: UC-03
Nama	: Pengelolaan Jadwal Penyiraman
Tujuan	: Untuk mengelola jadwal penyiraman tanaman KF-04
Tipe	: High Level
Aktor	: Pemilik Tanaman
Skenario Utama	
Kondisi Awal	: Perangkat lunak menampilkan dasbor admin.
Aksi Aktor	Reaksi Sistem
1. Memilih menu pengelolaan jadwal penyiraman	2. Menampilkan halaman pengelolaan jadwal penyiraman
	3. Menampilkan jadwal penyiraman tanaman
4. Menekan tombol ubah jadwal	5. Menerima perintah ubah jadwal
	6. Menyesuaikan jadwal yang ditentukan
	7. Menyimpan jadwal penyiraman
	8. Menampilkan pesan jadwal berhasil disimpan
9. Menerima informasi jadwal disimpan	
Kondisi Akhir: Perangkat lunak menampilkan halaman pengelolaan jadwal dan menampilkan pemberitahuan jadwal berhasil disimpan	
Skenario Alternatif	

Kondisi	: Pengaturan jadwal gagal dilakukan.	
	Aksi Aktor	Reaksi Sistem
		1. Perangkat lunak menampilkan pesan jadwal gagal disimpan
		2. Menampilkan halaman pengelolaan jadwal penyiraman
Kondisi Akhir	: Perangkat lunak menampilkan halaman pengelolaan jadwal penyiraman dan menampilkan pemberitahuan jadwal gagal disimpan	

Tabel 4.9 Skenario use case Menyiram tanaman terjadwal.

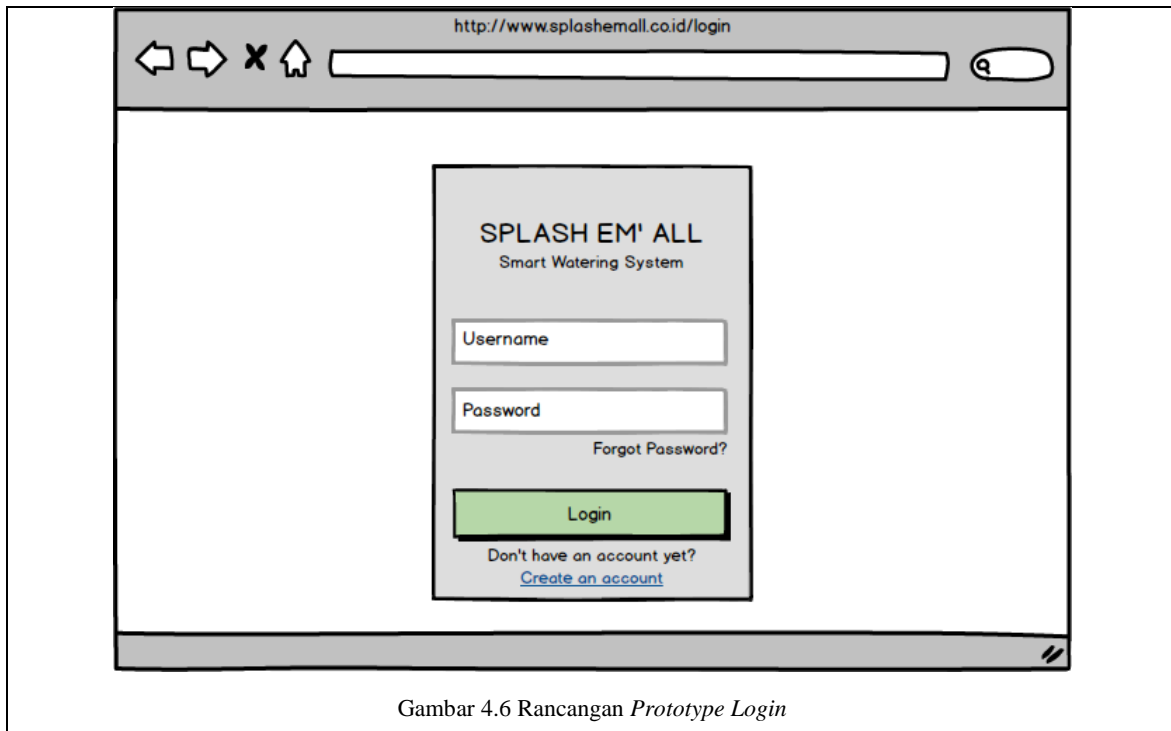
Identifikasi		
No.	: UC-04	
Nama	: Penyiraman Tanaman Terjadwal.	
Tujuan	: Menyiram tanaman sesuai jadwal KF-05	
Tipe	: High Level	
Aktor	: Perangkat penyiraman	
Skenario Utama		
Kondisi Awal	: Perangkat penyiraman memeriksa jadwal penyiraman.	
	Aksi Aktor	Reaksi Sistem
	1. Memeriksa jadwal penyiraman	
		2. Menerima jadwal penyiraman
		3. Menerima perintah penyiraman tanaman
	4. Melakukan penyiraman tanaman	
		5. Mengubah status kelembaban tanaman
		6. Menampilkan status kelembaban tanaman
Kondisi Akhir:	Perangkat lunak berhasil melakukan penyiraman tanaman	
Skenario Alternatif		
Kondisi	: Perangkat penyiraman tidak terhubung	
	Aksi Aktor	Reaksi Sistem
		1. Perangkat lunak menampilkan pesan perangkat penyiraman tidak terhubung.
		2. Menampilkan dashboard admin
Kondisi Akhir	: Perangkat lunak menampilkan dasbor admin dan menampilkan pesan tidak terhubung	

4.1.2.4 Prototype Perangkat Lunak

Prototype perangkat lunak menggambarkan gambaran awal *prototype* dari perangkat lunak yang akan dibangun berikut akan dipaparkan gambaran awal beserta penjelasannya :

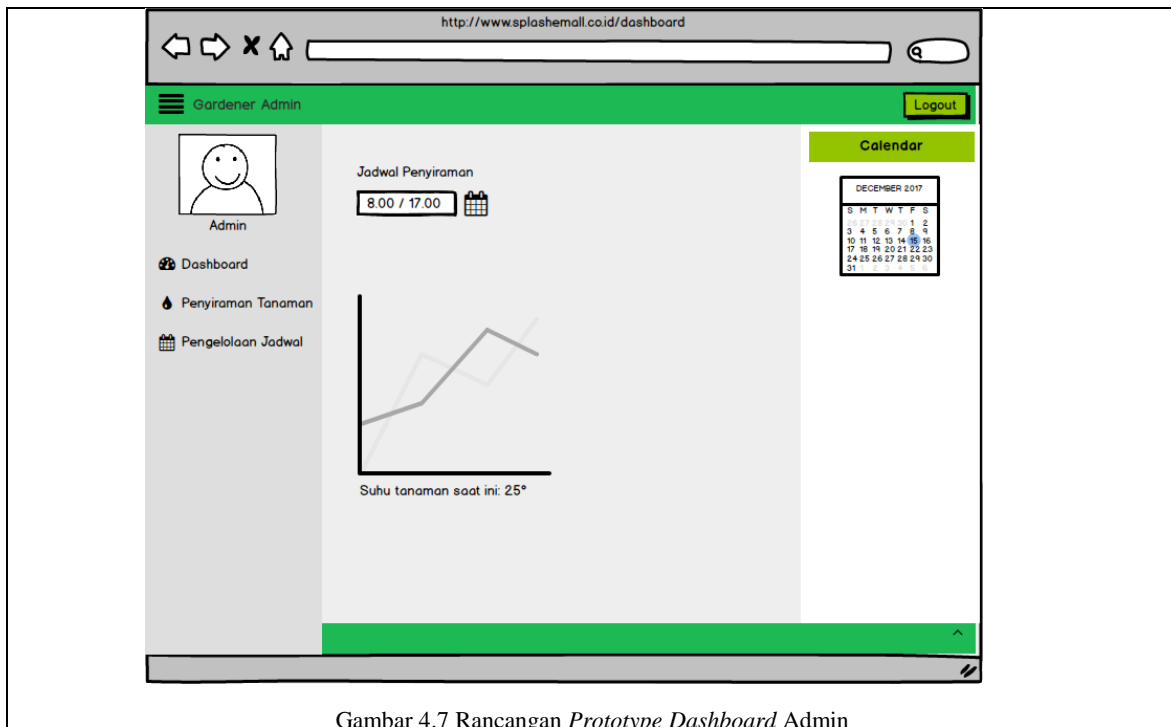
1. Perancangan Prototype Login

Berikut ini merupakan perancangan antarmuka untuk halaman *Login*, Pada antarmuka *Login* terdapat *form* untuk mengisi *username* dan *password*, kemudian terdapat tombol yang berfungsi untuk melakukan *login* dan tombol untuk masuk ke halaman pendaftaran akun. Gambaran *prototype login* akan dipaparkan pada gambar 4.6

Gambar 4.6 Rancangan *Prototype Login*

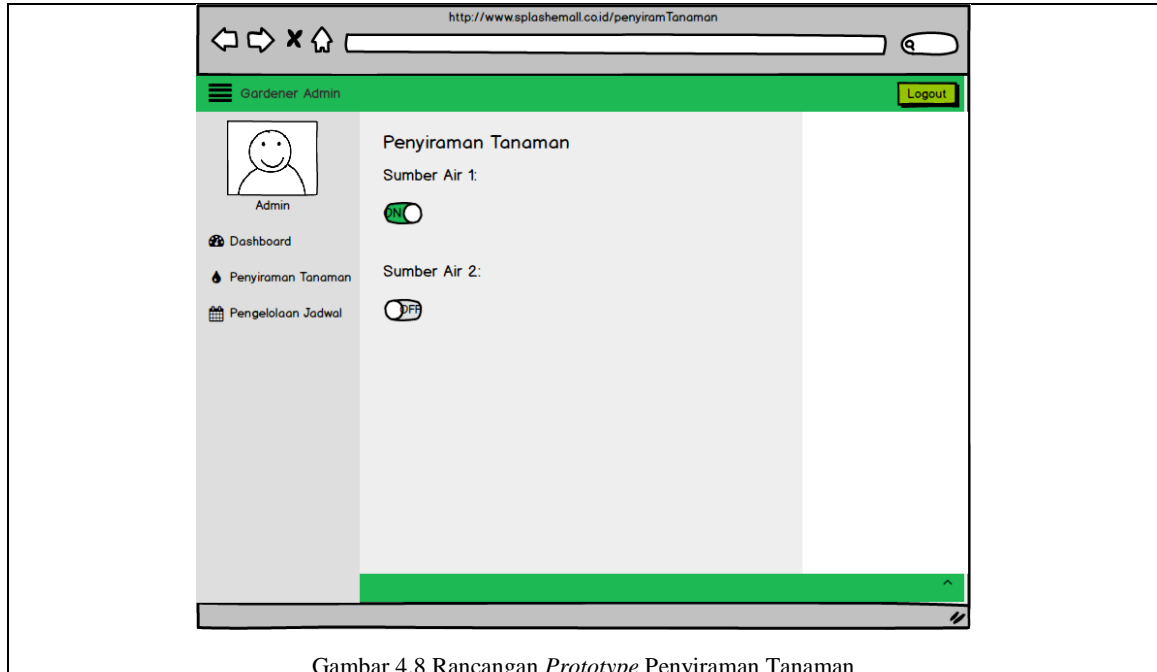
2. Perancangan *Prototype Dashboard Admin*

Berikut ini merupakan perancangan *prototype dashboard* untuk admin yang akan menggunakan perangkat lunak penyiraman tanaman otomatis, menampilkan suhu tanaman dan jadwal penyiraman yang telah ditentukan. Gambaran *prototype dashboard* admin akan dipaparkan pada gambar 4.7

Gambar 4.7 Rancangan *Prototype Dashboard Admin*

3. Perancangan Prototype Penyiraman Tanaman

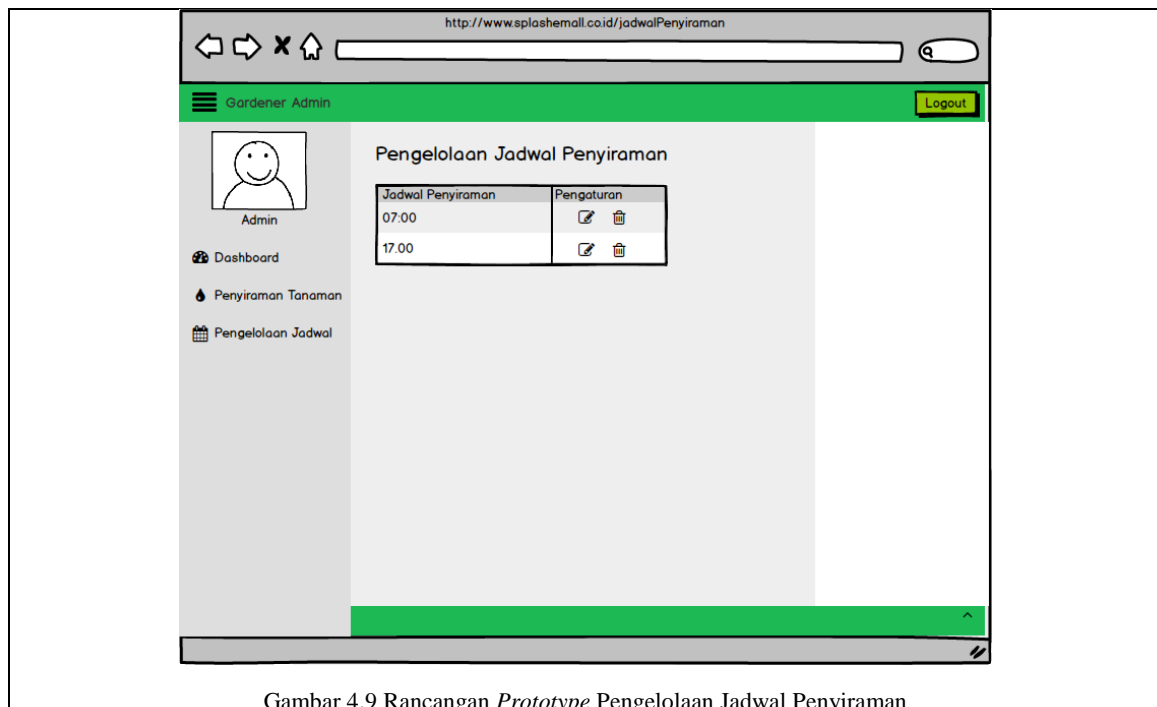
Berikut ini merupakan perancangan *prototype* penyiraman tanaman untuk mengoperasikan perangkat penyiram tanaman melakukan penyiraman tanaman atau membuka pompa air. Gambaran *prototype* penyiraman tanaman akan dipaparkan pada gambar 4.8



Gambar 4.8 Rancangan *Prototype* Penyiraman Tanaman

4. Perancangan Prototype Pengelolaan Jadwal Penyiraman

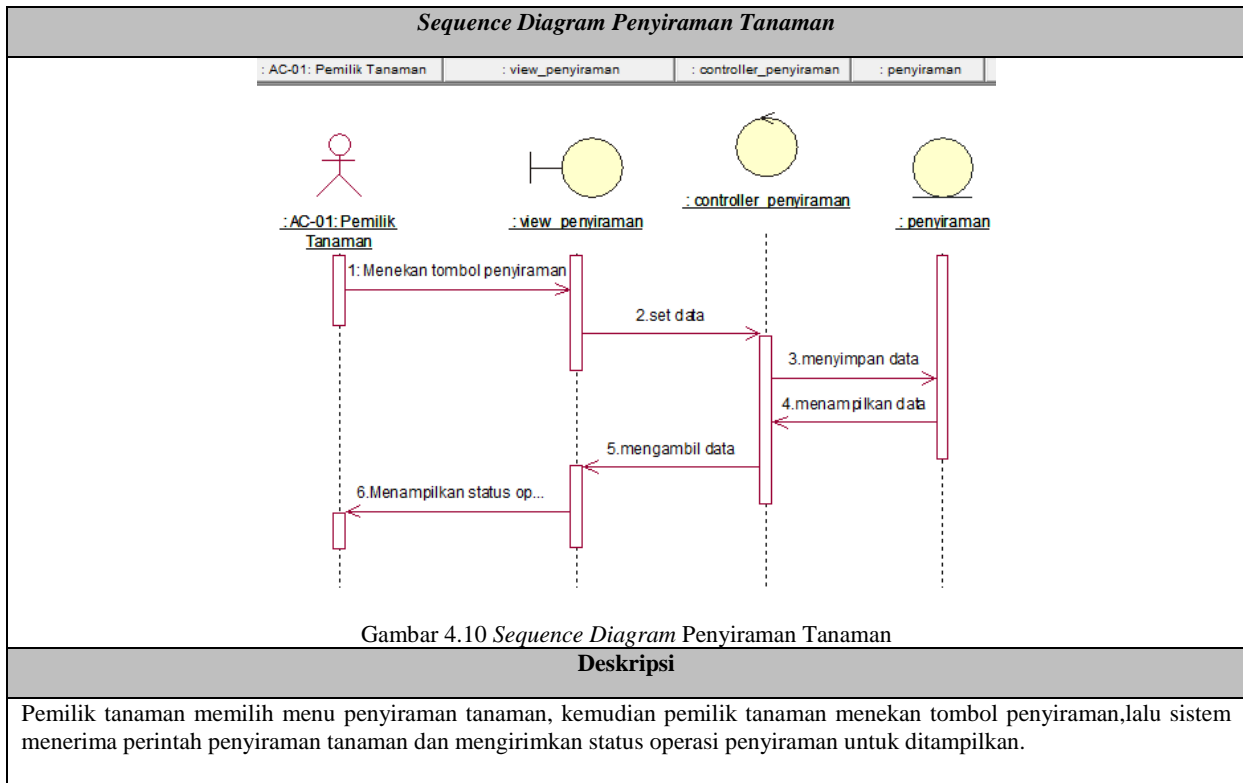
Berikut ini merupakan perancangan *prototype* pengelolaan jadwal penyiraman, untuk mengatur jadwal penyiraman tanaman sesuai waktu yang diinginkan. Gambaran *prototype* akan dipaparkan pada gambar 4.9 di bawah



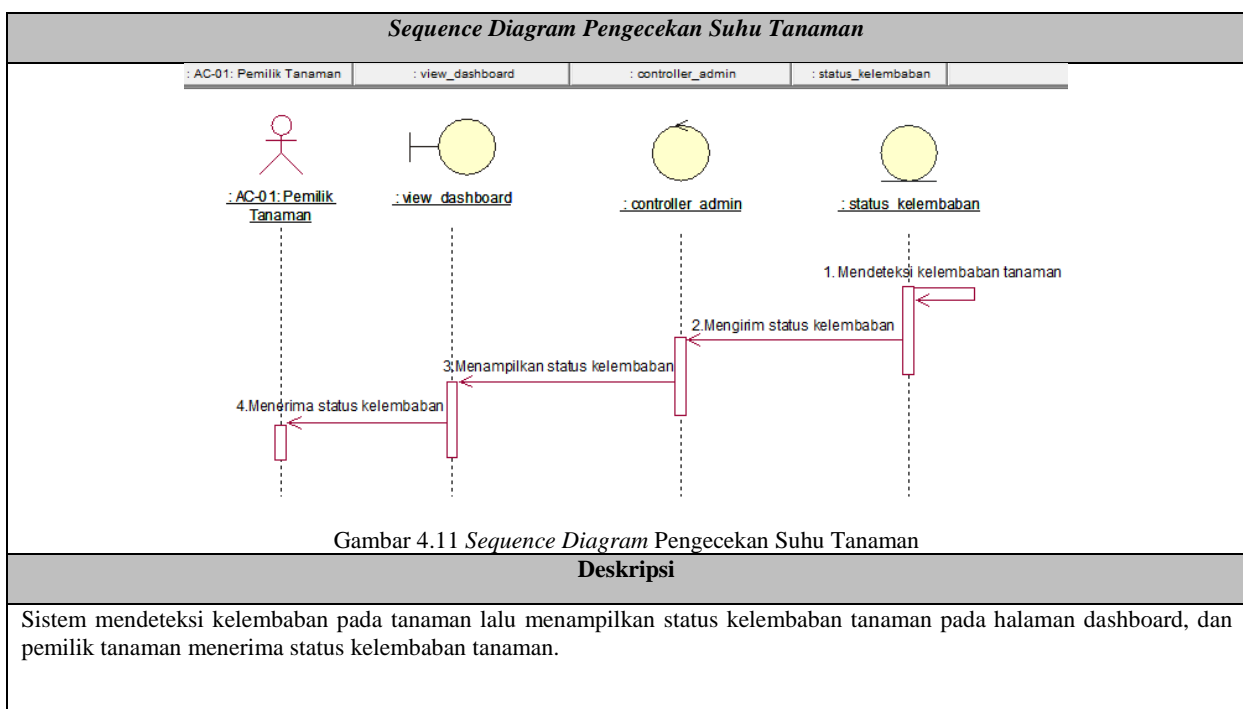
Gambar 4.9 Rancangan *Prototype* Pengelolaan Jadwal Penyiraman

4.1.2.5 Sequence Diagram

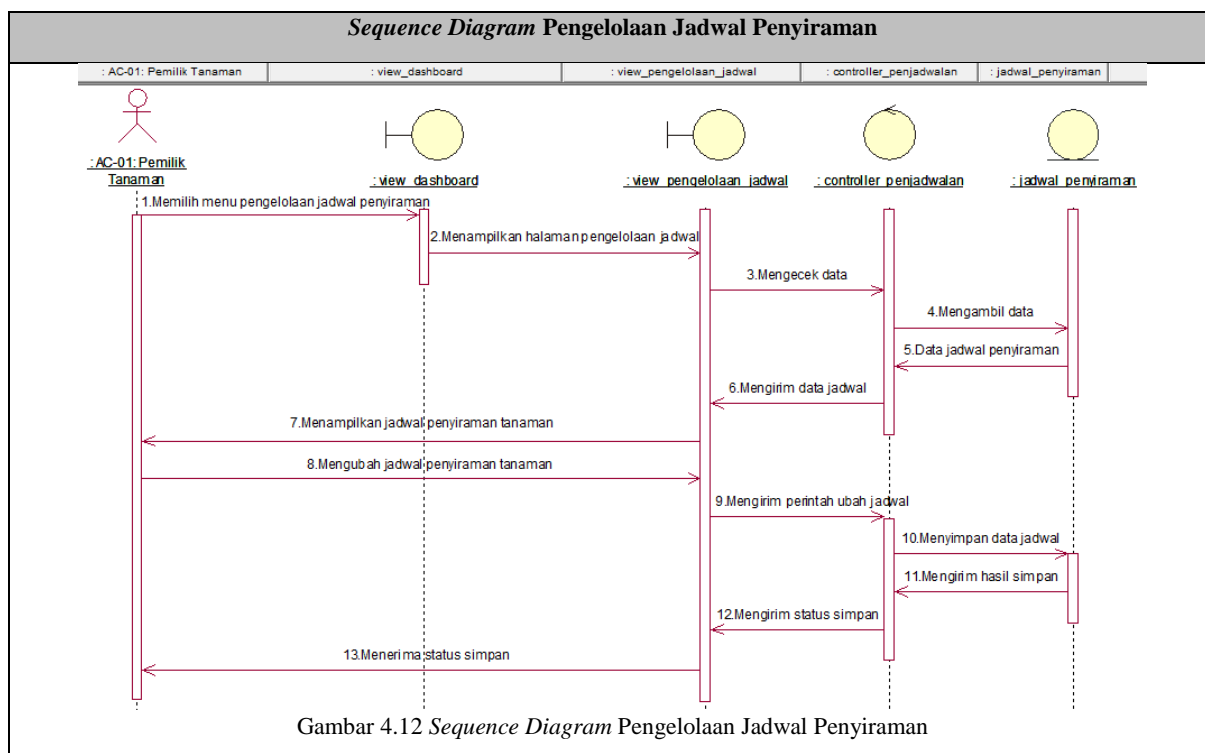
Sequence Diagram merupakan diagram yang merepresentasikan proses-proses yang terjadi di dalam interaksi antar kelas. Proses tersebut bisa berupa penyampaian suatu perintah ataupun hanya mengirimkan pesan-pesan tertentu. Berikut ini merupakan *sequence diagram* penyiraman tanaman akan dipaparkan pada Gambar 4.10



Berikut ini merupakan *sequence diagram* pengecekan suhu tanaman akan dipaparkan pada Gambar 4.11



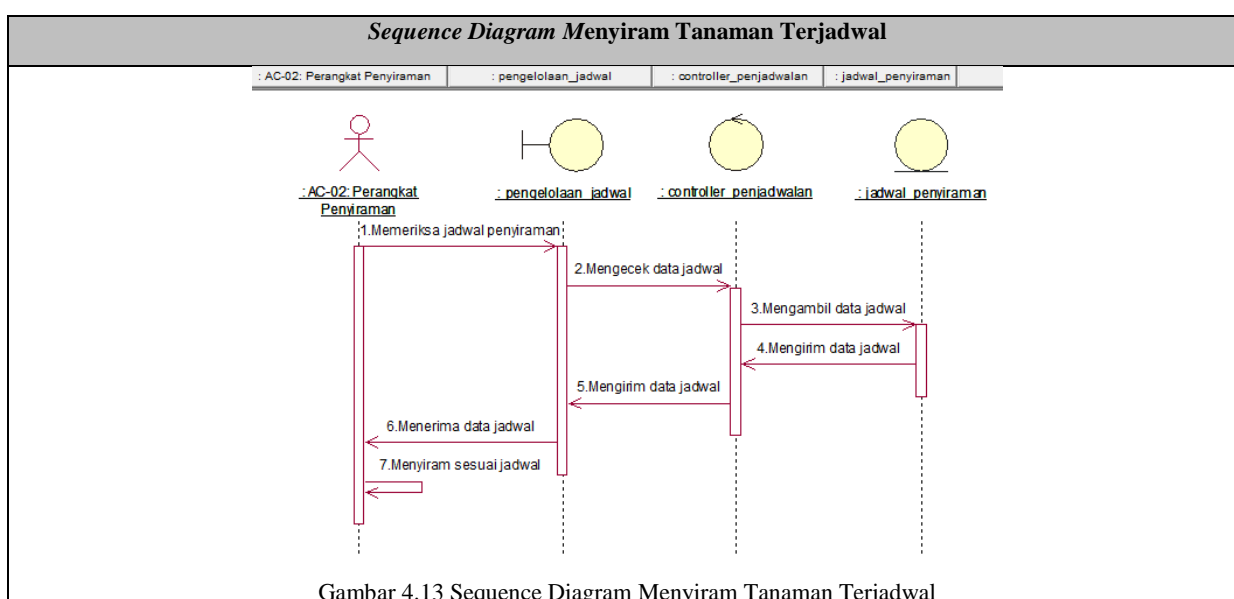
Berikut ini merupakan *sequence diagram* pengelolaan jadwal penyiraman akan dipaparkan pada Gambar 4.12



Deskripsi

Pemilik tanaman memilih menu pengelolaan jadwal penyiraman, lalu sistem menampilkan halaman pengelolaan jadwal penyiraman dan menampilkan jadwal penyiraman tanaman, pemilik tanaman mengubah jadwal penyiraman kemudian sistem menerima perintah ubah jadwal, menyesuaikan jadwal yang ditentukan dan menyimpan jadwal penyiraman, sistem menampilkan pesan jadwal telah tersimpan.

Berikut ini merupakan *sequence diagram* menyiram tanaman terjadwal akan dipaparkan pada Gambar 4.13

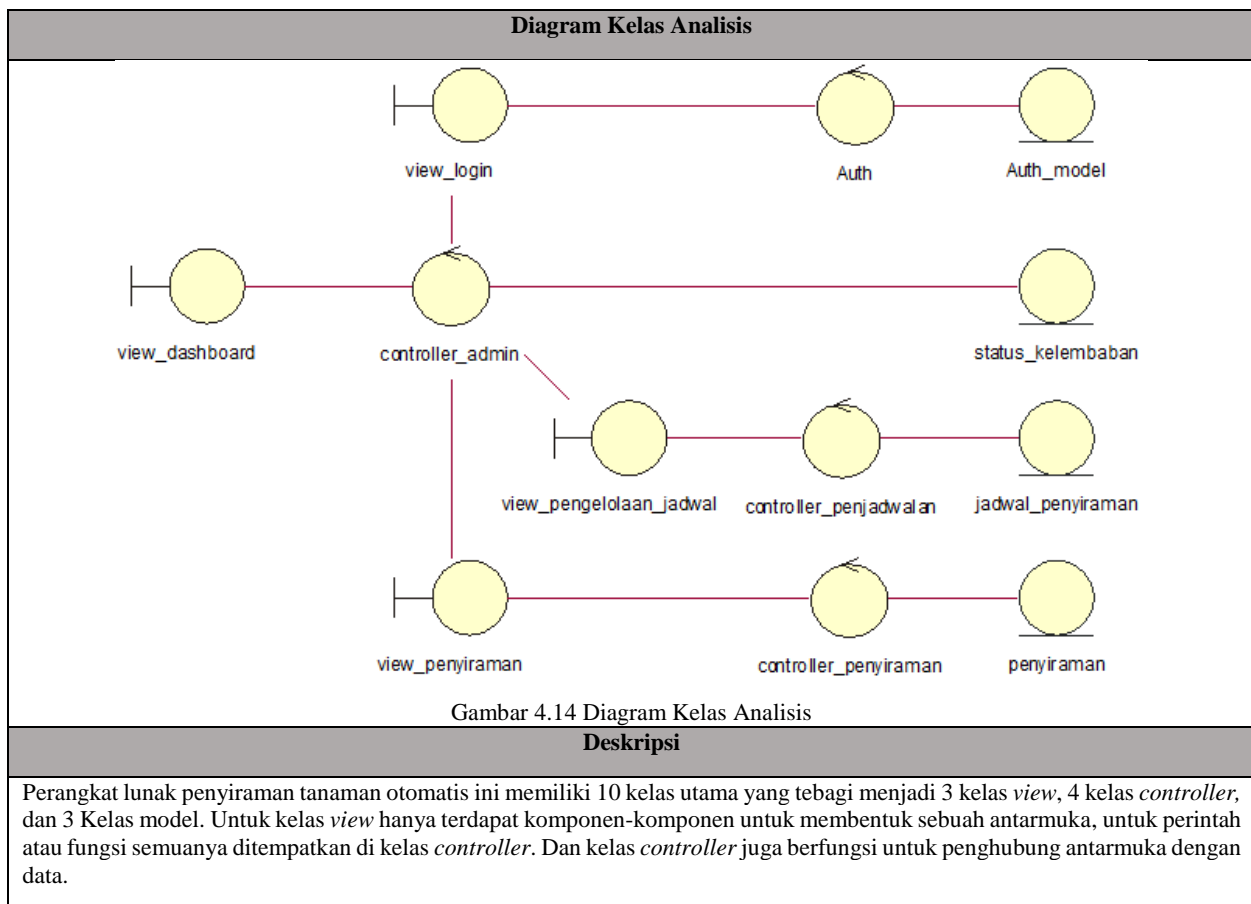


Deskripsi

Perangkat penyiraman tanaman memeriksa jadwal penyiraman, sistem mengecek data jadwal penyiraman lalu mengirim data jadwal, perangkat penyiraman tanaman menerima jadwal penyiraman dan melakukan operasi penyiraman tanaman.

4.1.2.6 Diagram Kelas Analisis

Diagram kelas analisis digunakan untuk merepresentasikan interaksi antar objek-objek kelas dari perangkat lunak yang dibangun. Objek kelas tersebut digambarkan dengan *Boundary*, *Controller*, dan *Entity*. *Boundary* merupakan objek kelas yang merepresentasikan objek kelas untuk *interface* atau antarmuka pada perangkat lunak. *Controller* merupakan objek kelas yang merepresentasikan objek kelas untuk kelas kontrol yang menangani fungsi-fungsi pada perangkat lunak. *Entity* merupakan objek kelas yang merepresentasikan objek kelas untuk struktur data yang berfungsi untuk mengelola basis data. Berikut diagram kelas analisis untuk perangkat lunak penyiraman tanaman otomatis akan ditampilkan pada gambar 4.14



4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan proses secara bertahap dimana seluruh kebutuhan atau persyaratan yang telah dianalisis kemudian diterjemahkan menjadi cetak biru (*blue print*) yang digunakan untuk membangun perangkat lunak. Berikut ini perancangan perangkat lunak yang dibangun.

4.2.1 Perancangan Kelas

Pada diagram kelas analisis dipresentasikan hubungan antar kelas yang akan dibuat dalam pembangunan perangkat lunak penyiraman tanaman otomatis. Pada tahap perancangan, kelas tersebut

dijelaskan pada tabel. Berikut ini pada tabel 4.10 akan dijelaskan mengenai operasi-operasi yang terdapat pada kelas.

Tabel 4.10 Tabel Rancangan Operasi Kelas

No	Nama Kelas	Nama Operasi	Keterangan	Tipe Kelas
1.	Auth	logged_in_check()	Berfungsi untuk menangani operasi login pada perangkat lunak.	Controller
		logout()	Berfungsi untuk menangani operasi logout dari perangkat lunak.	
2.	controller_admin	dashboard()	Berfungsi untuk menampilkan data pada halaman dashboard.	Controller
		charts()	Berfungsi untuk menampilkan data charts.	
		penyiraman()	Berfungsi untuk menampilkan data pada halaman penyiraman.	
		pengelolaanJadwal()	Berfungsi untuk menampilkan data pada halaman pengelolaan jadwal.	
		Logout()	Berfungsi untuk menangani operasi logout dari perangkat lunak.	
3.	controller_penyiraman	on_penyiraman()	Berfungsi untuk menangani operasi on penyiraman.	Controller
		off_penyiraman()	Berfungsi untuk menangani operasi off penyiraman.	
4.	controller_penjadwalan	pengelolaanJadwal()	Berikut ini adalah <i>getter</i> dan <i>setter</i> dari variable-variable yang terdapat pada kelas penduduk. Berfungsi untuk mendapatkan isi dari variable dan mengisi variable tersebut.	Controller
		viewJadwal()		
		insertJadwal()		
		formUpdateJadwal()		
		updateJadwal()		
		deleteJadwal()		
		cekJadwal()		
5.	Auth_Model	validate()	Berfungsi untuk menampilkan data pada halaman dashboard.	Model
		get_data()	Berfungsi untuk menampilkan data charts.	
6.	Jadwal_penyiraman	insert()	Berikut ini adalah <i>getter</i> dan <i>setter</i> dari variable-variable yang terdapat pada kelas penduduk. Berfungsi untuk mendapatkan isi dari variable dan mengisi variable tersebut.	Model
		get_all()		
		get()		
		update()		
		delete()		
7.	status_kelembaban	setSuhu()	Berfungsi untuk menampilkan data pada halaman dashboard.	Model
		getSuhu()	Berfungsi untuk menampilkan data charts.	
8.	penyiraman	statusPenyiraman()	Berfungsi untuk menampilkan data pada halaman dashboard.	Model
		get_data()	Berfungsi untuk menampilkan data charts.	

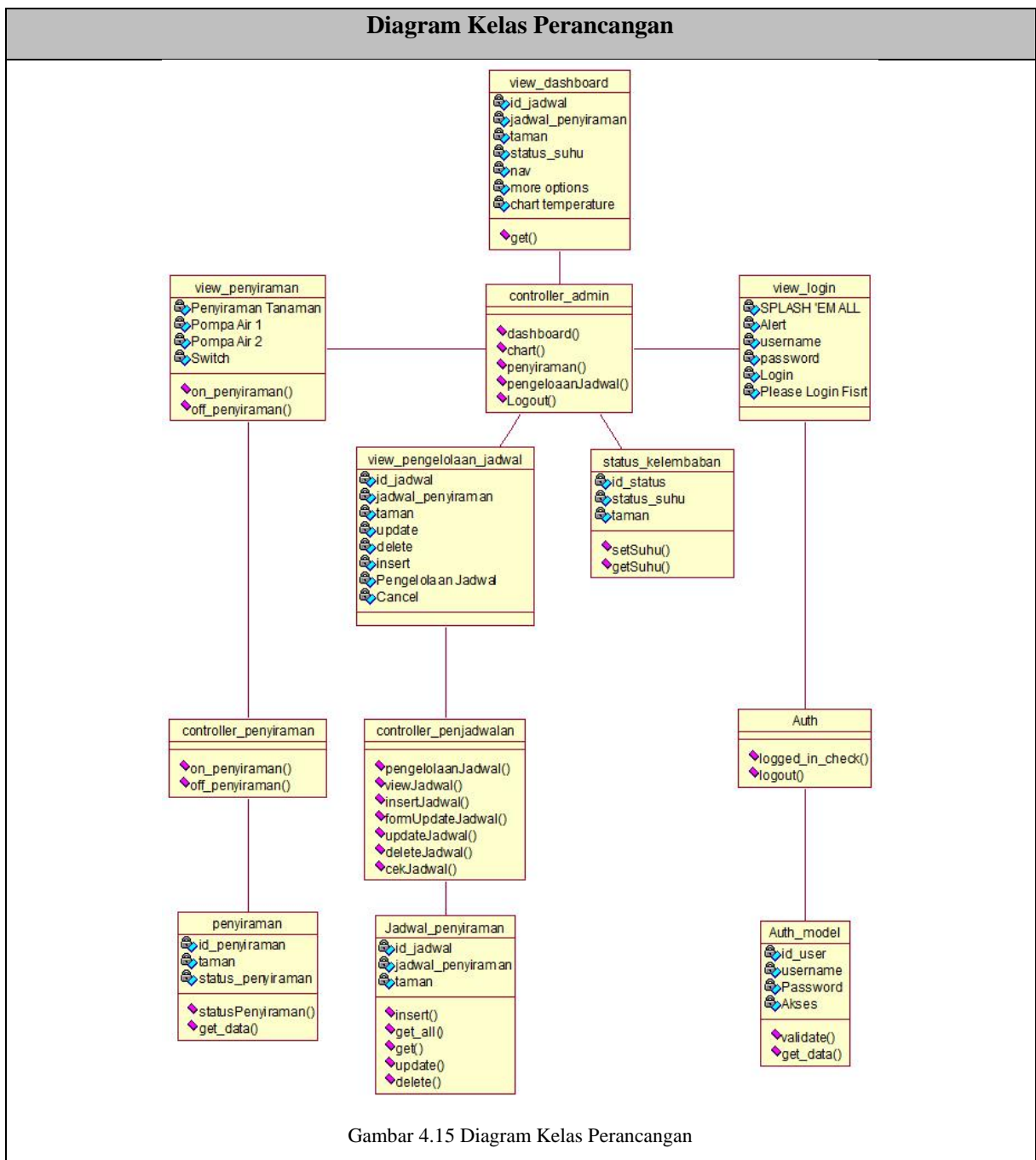
Selanjutnya setelah operasi-operasi yang terdapat pada kelas, berikut ini akan dipaparkan atribut-atribut yang terapat pada kelas pada tabel 4.11.

Tabel 4.11 Atribut kelas

No	Nama Kelas	Atribut	Tipe Data	Tipe Kelas
1.	view_dashboard	id_jadwal	Integer	View
		jadwal_penyiraman	Time	
		taman	String	
		status_suhu	Integer	
		nav	Button	
		more options	Button	
		chart temperature	String	
2.	view_penyiraman	Penyiraman Tanaman	Text	View
		Pompa Air 1	Text	
		Pompa Air 2	text	
		Switch	checkbox	
3.	view_pengelolaan_jadwal	id_jadwal	Integer	View
		jadwal_penyiraman	Time	
		taman	String	
		update	Button	
		delete	Button	
		insert	button	
		Pengelolaan Jadwal	text	
		Cancel	button	
4.	view_login	SPLASH 'EM ALL	Text	View
		Alert	Button	
		username	Text	
		password	password	
		Login	Button	
		Please Login First	text	
5.	Auth_model	id_user	Integer	Model
		username	String	
		Password	String	
		Akses	String	
6.	Jadwal_penyiraman	id_jadwal	Integer	Model
		jadwal_penyiraman	Time	
		taman	String	
7.	status_kelembaban	id_suhu	Integer	Model

No	Nama Kelas	Atribut	Tipe Data	Tipe Kelas
		status_suhu	Integer	
		taman	String	
8.	penyiraman	id_penyiraman	Integer	Model
		taman	String	
		status_penyiraman	String	

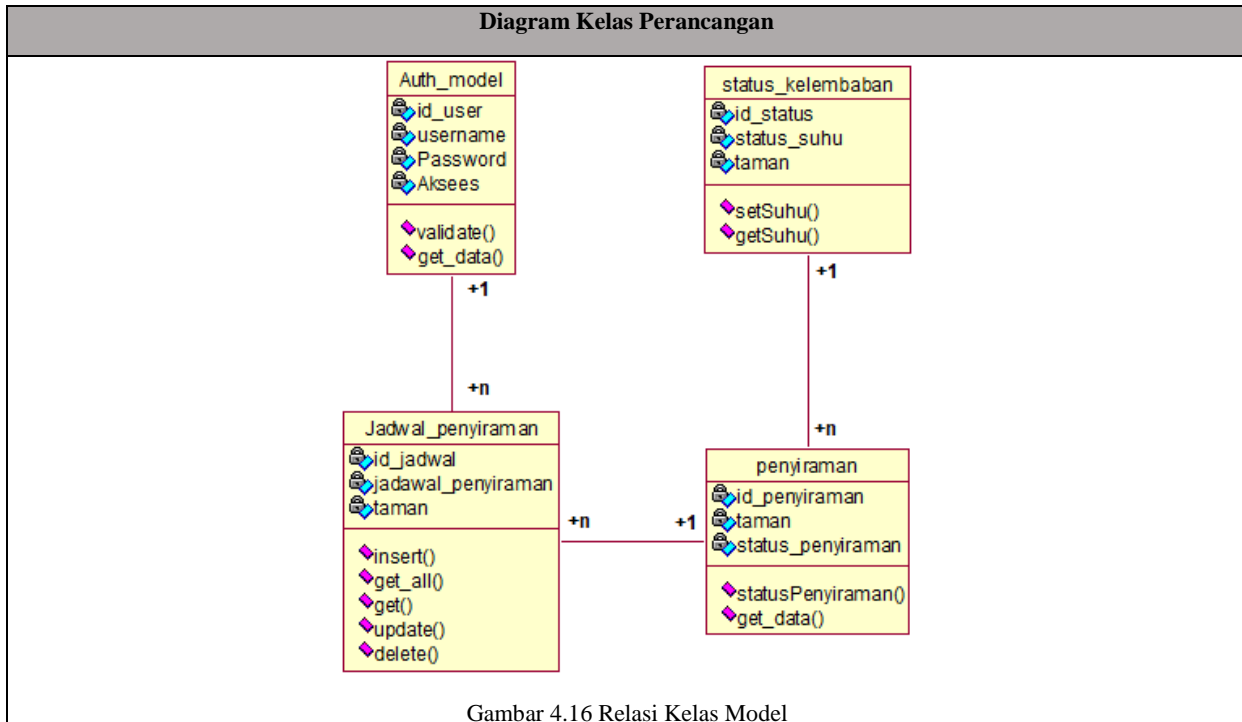
Berikut ini representasi kelas perancangan dari hasil analisis menjadi diagram kelas perancangan dapat dilihat pada gambar 4.15.



Gambar 4.15 Diagram Kelas Perancangan

4.2.2 Perancangan Data

Perancangan data merupakan pemilihan representasi logis dari objek data. Mengacu pada kelas model yang direpresentasikan pada gambar 4.16 menjadi struktur file. Pada gambar 4.16 merepresentasikan relasi antar kelas model adalah sebagai berikut :



Gambar 4.16 Relasi Kelas Model

Dari representasi relasi antar kelas model pada gambar 4.16 maka menentukan kandidat data dari perangkat lunak penyiraman tanaman otomatis pada tabel 4.12 berikut ini:

Tabel 4.12 Perancangan Data

No	Nama Kelas	Nama Tabel	Nama Field	Tipe Data
1	Auth_mdel	tb_user	Id_user	Int(11)
			Username	varchar(30)
			Password	Varchar(50)
			Akses	enum
2	jadwal_penyiraman	tb_jadwal_penyiraman	id_jadwal	int(3)
			jadwal_penyiraman	time
			taman	varchar(15)
3	penyiraman	tb_penyiraman	id_penyiraman	int(11)
			taman	int(15)
			status_penyiraman	Int(3)
4	status_kelembaban	tb_status_kelembaban	id_suhu	int(3)
			status_suhu	int(2)
			taman	varchar(15)

4.2.3 Perancangan *Algoritma* dan *Query*

Perencanaan algoritma dan query merupakan tahapan untuk mendefinisikan langkah-langkah apa saja yang dilakukan oleh perangkat lunak nantinya dalam menyelesaikan permasalahan yang ada. Bentuk dari perencanaan ini memadukan algoritma dan flowchart dari perangkat lunak sehingga pengembang akan mengetahui proses apa saja yang terdapat di dalam perangkat lunak.

4.2.3.1 Algoritma dan query dari fungsi menyiram tanaman terjadwal

Pada bagian ini akan dijelaskan algoritma saat menampilkan anggota keluarga. Berikut ini penjelasannya pada tabel 4.13 tentang *algoritma* dari fungsi menyiram tanaman terjadwal

Tabel 4.13 Algoritma dari fungsi menyiram tanaman terjadwal

Identifikasi	
Nama Kelas : controller_penyiraman	
Nama Operasi : on_penyiraman()	
Deskripsi : Berfungsi untuk memeriksa jadwal penyiraman dan mengoprasikan penyiraman.	
Algoritma	Flowchart
<ol style="list-style-type: none"> 1. Memeriksa jadwal penyiraman yang telah dibuat. 2. Mengambil data jadwal penyiraman. 3. Mengirim hasil pengecekan data jadwal penyiraman tanaman. 4. Melakukan pemilihan keputusan, ketika jadwal tidak ditemukan akan kembali memeriksa jadwal penyiraman, ketika tidak ada jadwal saat ini pengecekan akan dihentikan. 5. Dan ketika ada jadwal tepat saat ini akan mengirimkan perintah penyiraman sesuai jadwal. 	<pre> graph TD Start([Start]) --> Check[Memeriksa jadwal penyiraman] Check --> Data[/Data jadwal penyiraman/] Data --> Get[Mengambil data jadwal penyiraman] Get --> Send[Mengirim data jadwal] Send --> Decision{Jadwal} Decision -- "Tidak ada jadwal saat ini" --> End([End]) Decision -- "Jadwal saat ini" --> Water[Menyiram sesuai jadwal] Water --> End </pre>

BAB 5

IMPLEMENTASI DAN PENGUJIAN

bab ini menjelaskan mengenai tahap-tahap pembuatan aplikasi berdasarkan hasil rancangan yang telah dibuat pada tahap perancangan, dan menunjukkan hasil dari pengujian pada aplikasi.

5.1. Implementasi

Implementasi merupakan suatu tindakan atau pelaksanaan dari sebuah perencanaan yang telah disusun secara matang dan terperinci. Implementasi dilakukan setelah perancangan sudah dianggap baik. Implementasi di dalam pembangunan perangkat lunak biasa disebut dengan konstruksi yang merupakan suatu proses pengubahan spesifikasi sistem menjadi sistem yang dapat digunakan atau dapat dikatakan sebagai tahapan menerjemahkan hasil perancangan ke dalam kode program.

5.1.1 Implementasi Perangkat Lunak

Pada bagian ini dijelaskan mengenai hal hal yang harus dipersiapkan dan dilakukan untuk mengimplementasikan hasil analisis dan rancangan pada perangkat lunak penyiraman tanaman otomatis. Implementasi perangkat lunak tersebut melingkupi implementasi *web service* dan perangkat lunak berbasis *website* yang dibangun.

5.1.1.1 Kebutuhan implementasi

Kebutuhan implementasi perangkat lunak memerlukan spesifikasi perangkat keras agar perangkat lunak yang dibangun dapat berjalan dengan baik. Kebutuhan perangkat lunak penyiraman tanaman otomatis terbagi menjadi 2 (dua) bagian yaitu (1) kebutuhan bagi perangkat keras, (2) kebutuhan bagi perangkat lunak berbasis *website*. Masing masing kebutuhan perangkat lunak dijelaskan sebagai berikut:

1. Kebutuhan perangkat keras

Pada bagian ini menjelaskan mengenai minimal kebutuhan perangkat keras yang harus tersedia pada saat perangkat lunak penyiraman tanaman otomatis akan diterapkan. Penjelasan dari spesifikasi kebutuhan perangkat keras akan dijelaskan sebagai berikut:

a. Spesifikasi perangkat keras untuk pembangunan perangkat lunak berbasis *website*

Adapun minimal spesifikasi yang dibutuhkan guna mendukung penggunaan perangkat lunak penyiraman tanaman otomatis yang akan dipaparkan pada tabel 5.1 di bawah:

Tabel 5.1 Spesifikasi Perangkat Keras untuk perangkat lunak berbasis website

No	Kebutuhan Perangkat Keras	Spesifikasi Minimal
1	<i>Processor</i>	<i>Dual Core, 1GHZ</i>
2	<i>Memory</i>	512 MB
3	<i>Storage</i>	2 GB
4	<i>Bandwidth</i>	1 Mbps upload dan 1 Mbps download

2. Spesifikasi kebutuhan perangkat lunak

Pada bagian ini menjelaskan mengenai kebutuhan perangkat lunak pendukung yang harus dipersiapkan untuk menerapkan perangkat lunak penyiraman tanaman otomatis. Penjelasan kebutuhan perangkat lunak dijelaskan sebagai berikut:

a. perangkat lunak untuk pendukung perangkat lunak berbasis *website*

Adapun perangkat lunak yang harus dipersiapkan untuk perangkat lunak berbasis *website* agar dapat diterapkan, dijelaskan pada tabel 5.2.

Tabel 5.2 Perangkat lunak untuk pendukung *website*

No	Kebutuhan Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 8.0 atau lebih
2	<i>Web Server</i>	Apache
3	<i>Database</i>	MySQL

b. Perangkat lunak untuk pendukung perangkat lunak penyiraman tanaman otomatis

Adapun perangkat lunak yang harus dipersiapkan pada saat perangkat lunak penyiraman tanaman otomatis diterapkan yaitu sistem operasi yang terpasang pada *pc*. Kebutuhan minimal sistem operasi yang dibutuhkan adalah *Windows 8.0*.

5.1.1.2 Struktur program dan kode program

Pada bagian ini menjelaskan mengenai struktur dari folder dan kelas yang dibuat pada saat pembangunan perangkat lunak. Selain itu, dijelaskan pula mengenai fungsi dari masing-masing folder tersebut.

a. Struktur folder perangkat lunak penyiraman tanaman otomatis

Pada pembangunan perangkat lunak penyiraman tanaman otomatis ini menggunakan *design patern* MVC di mana dalam pembangunannya terdapat *Model, View, Controller* Adapun struktur folder dan kelas pada pembangunan perangkat lunak penyiraman tanaman otomatis dijelaskan sebagai berikut pada tabel 5.3 di bawah ini :

Tabel 5.3 Struktur folder perangkat lunak penyiraman tanaman otomatis

No	Nama Folder	Keterangan
1	model	Folder ini berisi kelas-kelas model yang berfungsi untuk manmpung data data yang dibutuhkan pada perangkat lunak.
2	View	Folder ini berisi file yang menampung semua antarmuka pada perangkat lunak dan untuk menampilkan data yang telah diolah oleh <i>controller</i> agar dapat ditampilkan di layar.
3	Controller	Folder ini berisi kelas kelas yang berfungsi untuk menampung data yang berasal dari model dan menampung logika logika pada perangkat lunak yang untuk diteruskan ke <i>view</i> .
4	Assets	Folder ini berisi file-file berupa gambar, suara, video untuk pendukung media perangkat lunak.

5.1.2 Implementasi rancangan antarmuka

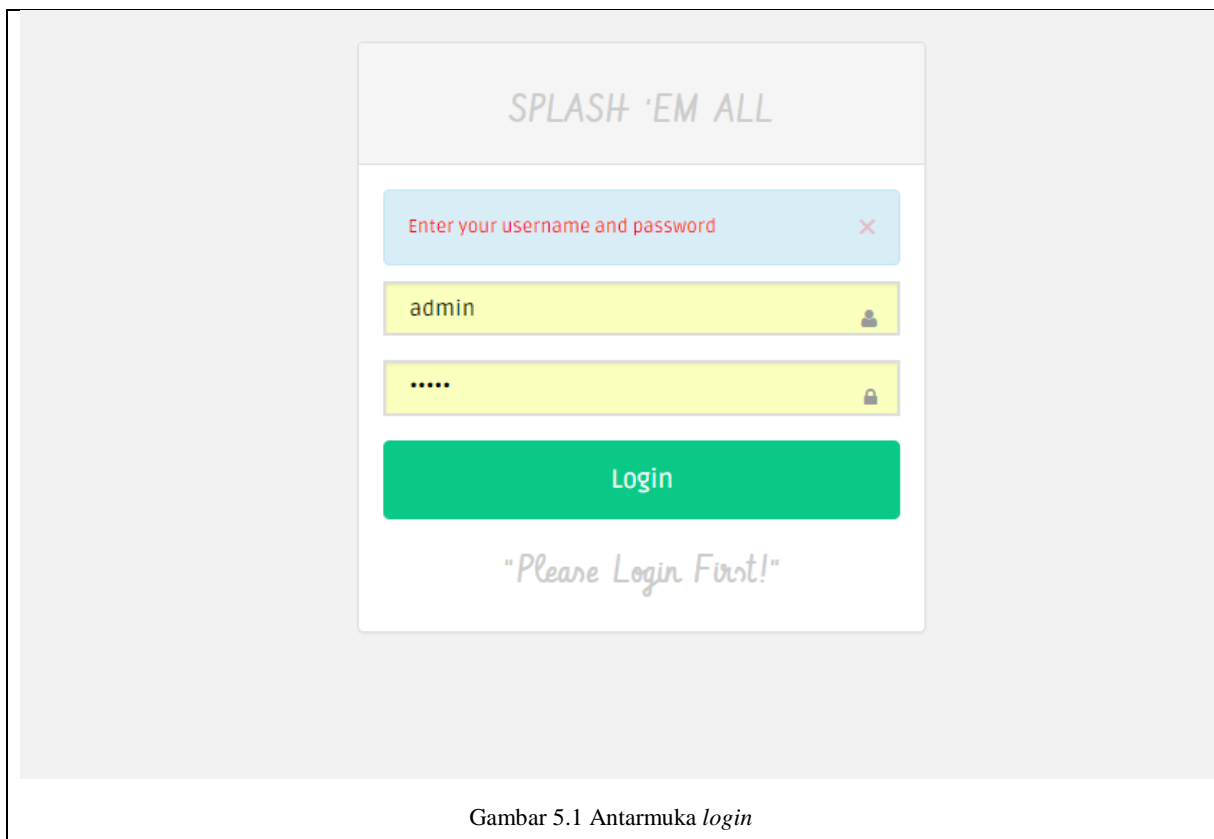
Pada bagian ini menjelaskan mengenai hasil implementasi dari perancangan antarmuka yang telah dibuat sebelumnya pada tahapan perancangan. Untuk hasil implementasi antarmuka dijelaskan sebagai berikut:

5.1.2.1 Implementasi rancangan antarmuka perangkat lunak

Pada bagian ini menjelaskan mengenai hasil implementasi dari perancangan antarmuka pada perangkat lunak penyiraman tanaman otomatis.

1. Implementasi antarmuka login

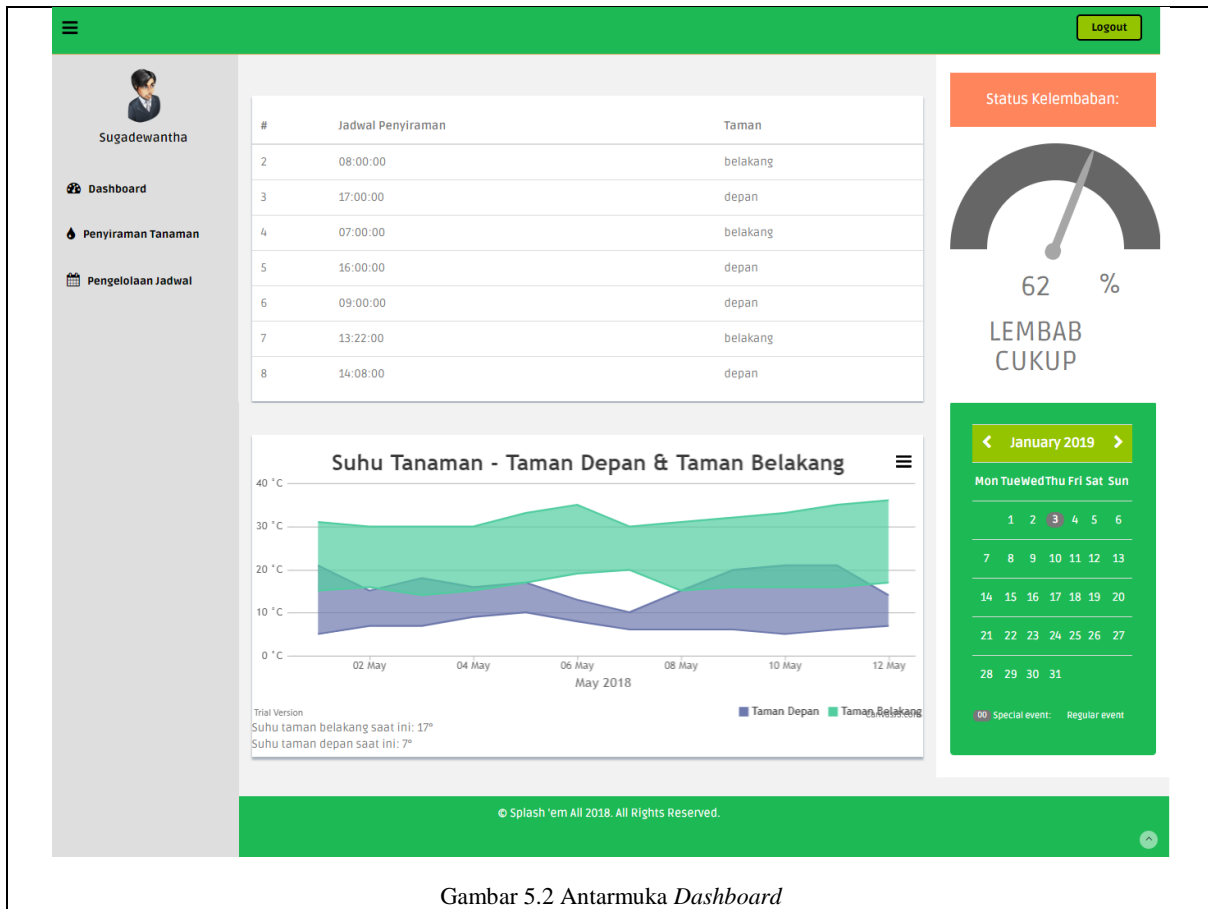
Berikut ini adalah implementasi antarmuka untuk login, pada antarmuka ini terdapat form *username* untuk mengisi *username* dan form password untuk mengisi password yang sudah terdaftar. Terdapat tombol Login berfungsi untuk memasuki kedalam halaman utama perangkat lunak. Gambaran antarmuka login akan dipaparkan pada gambar 5.1.



Gambar 5.1 Antarmuka login

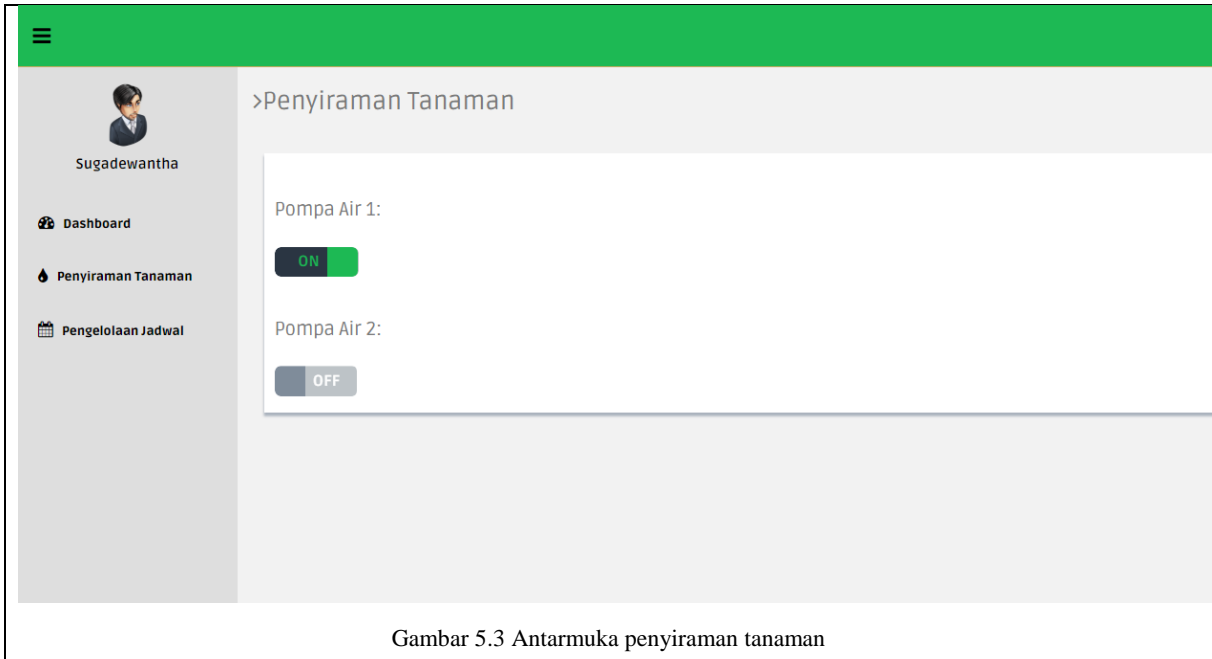
2. Implementasi antarmuka dashboard

Berikut merupakan implementasi antarmuka halaman utama yaitu *dashboard*. Pada halaman *dashboard* terdapat tabel jadwal yang telah ditentukan pada halaman pengelolaan jadwal, terdapat informasi suhu tanaman untuk mengetahui kelembaban tanaman. Gambaran antarmuka *dashboard* akan dipaparkan pada gambar 5.2 di bawah

Gambar 5.2 Antarmuka *Dashboard*

3. Implementasi antarmuka penyiraman tanaman

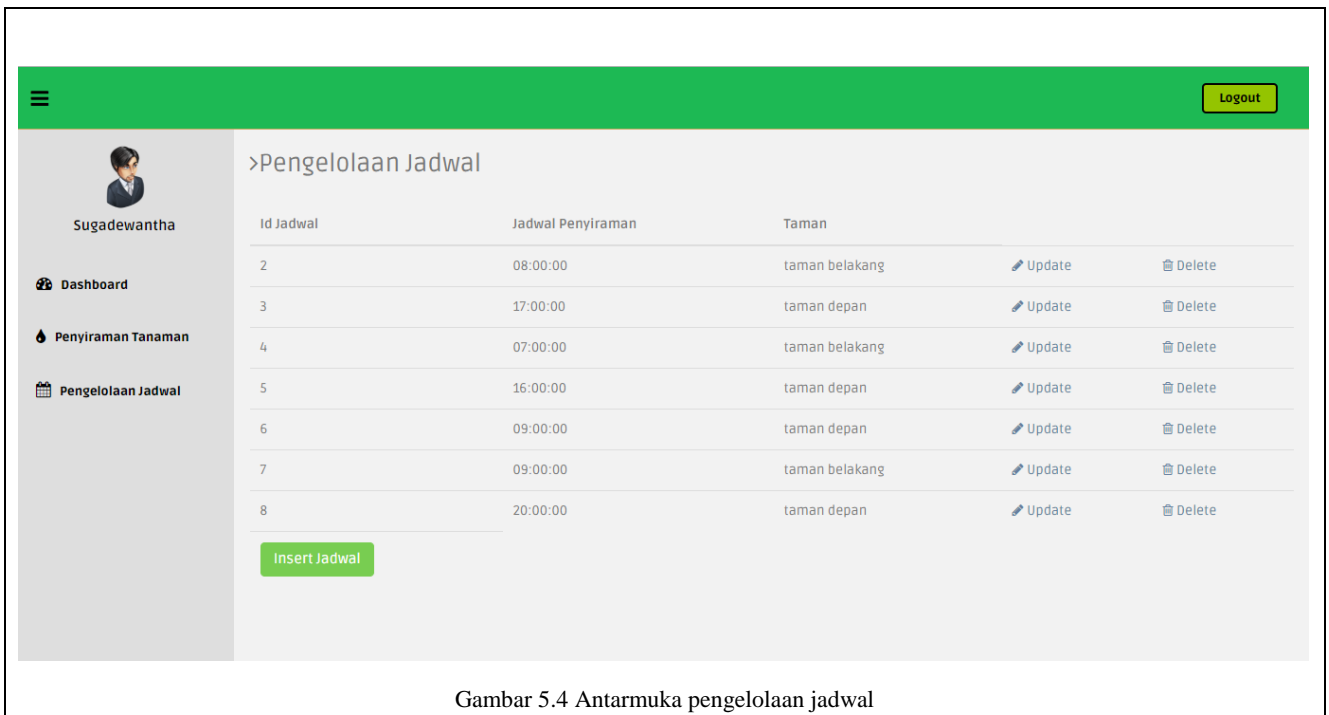
Berikut merupakan implementasi halaman penyiraman tanaman, pada halaman ini terdapat dua *toggle button* sebagai ganti kran air untuk mengaktifkan perangkat penyiraman tanaman. Gambaran antarmuka penyiraman tanaman akan dipaparkan pada gambar 5.3



Gambar 5.3 Antarmuka penyiraman tanaman

4. Implementasi antarmuka pengelolaan jadwal

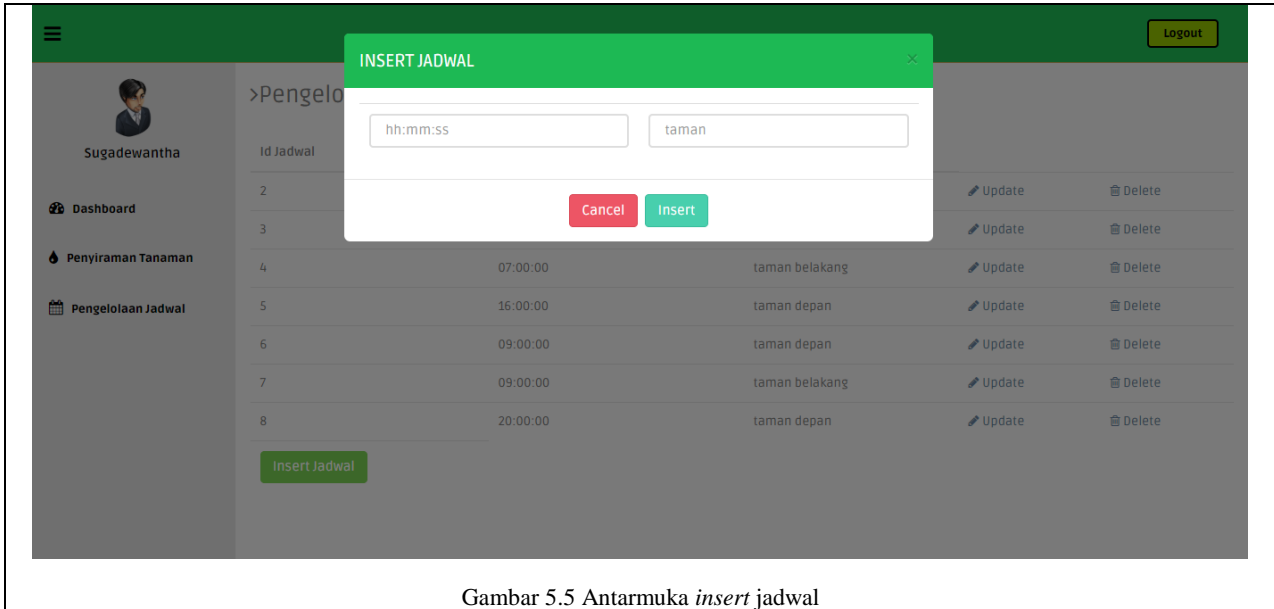
Berikut merupakan implementasi halaman pengelolaan jadwal, pada halaman ini terdapat tabel jadwal yang dapat diinputkan ke dalam database melalui tombol *Insert jadwal*, lalu pengguna dapat mengubah jadwal yang telah ditentukan melalui tombol *update* dan dapat menghapus jadwal yang telah ditentukan melalui tombol *delete*. Gambaran antarmuka pengelolaan jadwal akan dipaparkan pada gambar 5.4



Gambar 5.4 Antarmuka pengelolaan jadwal

5. Implementasi antarmuka insert jadwal

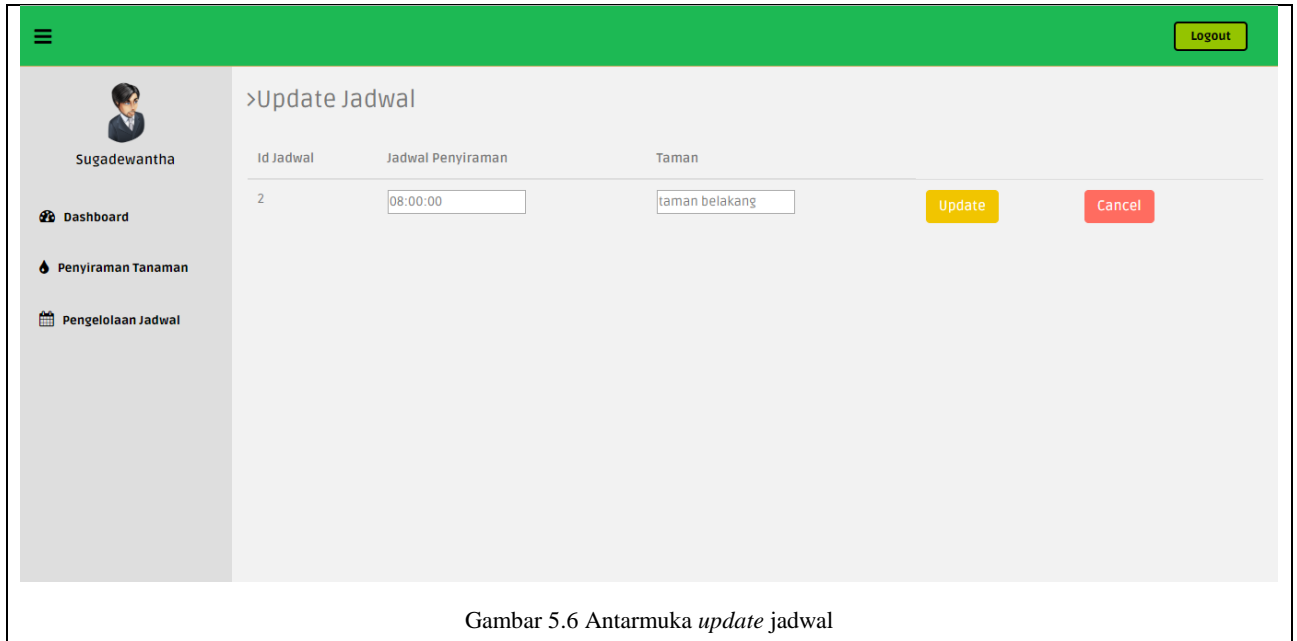
Berikut merupakan implementasi antarmuka halaman *insert* jadwal, pada bagian *insert* jadwal terdapat kolom *text* untuk menginputkan jadwal baru ke dalam *database* dan menyimpannya dengan mengklik tombol *insert* atau dapat membatalkan penyimpanan dengan mengklik tombol *cancel*. Gambaran antarmuka insert jadwal akan dipaparkan pada gambar 5.5 di bawah



Gambar 5.5 Antarmuka *insert* jadwal

6. Implementasi antarmuka update jadwal

Berikut merupakan implementasi antarmuka *update* jadwal, pada halaman *update* jadwal terdapat kolom *text* jadwal yang terpilih lalu pengguna dapat mengubahnya dengan jadwal baru dan menyimpannya dengan mengklik tombol *update* atau dapat membatalkan penyimpanan dengan mengklik tombol *cancel*. Gambaran antarmuka update jadwal akan dipaparkan pada gambar 5.6

Gambar 5.6 Antarmuka *update* jadwal

5.2. Pengujian

Pada tahap ini dilakukan pengujian terhadap perangkat lunak yang telah dibangun meliputi pengujian fungsional dan pengujian non fungsional. Pengujian ini bertujuan untuk memastikan bahwa aplikasi yang telah dibangun sesuai dengan kebutuhan yang telah didefinisikan sebelumnya. Kegiatan yang dilakukan untuk menguji perangkat lunak penyiraman tanaman otomatis dimulai dari melakukan perencanaan pengujian, memilih metode pengujian yang akan digunakan, menentukan hal-hal yang akan diuji, menetapkan langkah-langkah pengujian, dan melakukan pendokumentasian terhadap hasil pengujian. Untuk setiap langkah kegiatan dari pengujian perangkat lunak penyiraman tanaman otomatis dijelaskan sebagai berikut.

5.2.1. Rencana Pengujian

Rencana pengujian merupakan uraian kegiatan yang akan dilakukan untuk menguji suatu perangkat lunak seperti menentukan asumsi, dan menentukan pengujian yang akan digunakan.

5.2.1.1. Metode Pengujian

Pada pengujian perangkat lunak penyiraman tanaman otomatis ini akan menggunakan metode pengujian *black box testing*. *Black box testing* merupakan metode pengujian perangkat lunak yang berfokus pada fungsionalitas perangkat lunak dan hasil yang diberikan.

5.2.1.2. Fungsionalitas aplikasi

Fungsionalitas perangkat lunak merupakan penjelasan mengenai fungsi yang dimiliki oleh perangkat lunak penyiraman tanaman otomatis. Adapun fungsi yang dimiliki perangkat lunak penyiraman tanaman otomatis ditunjukkan pada tabel 5.4 berikut ini :

Tabel 5.4 Tabel fungsionalitas perangkat lunak

No	Kode Kebutuhan Fungsionalitas	Kebutuhan Fungsionalitas	Deskripsi
1	FU-01	Perangkat lunak harus dapat mengoperasikan penyiraman.	Perangkat lunak harus dapat menyiram tanaman, sehingga dibutuhkan fungsionalitas untuk mengoperasikan penyiraman tanaman.
2	FU -02	Perangkat penyiraman tanaman dapat menyiram dari dua sumber air	Perangkat penyiram tanaman harus dapat menyiram dari dua sumber air, sehingga dapat memperluas jarak penyiraman tanaman.
3	FU -03	Perangkat lunak harus dapat mengecek suhu tanaman.	Perangkat lunak harus dapat mengetahui informasi kelembaban pada tanaman, sehingga dibutuhkan fungsionalitas untuk mengecek suhu tanaman.
4	FU -04	Perangkat lunak harus dapat mengatur jadwal penyiraman.	Perangkat lunak harus dapat mengatur jadwal penyiraman tanaman, sehingga dibutuhkan fungsionalitas untuk mengatur jadwal penyiraman tanaman.
5	FU -05	Perangkat lunak harus dapat menyiram tanaman sesuai jadwal yang telah ditentukan.	Perangkat lunak harus dapat menyiram tanaman sesuai jadwal yang telah ditentukan, sehingga dibutuhkan fungsionalitas untuk menyiram sesuai jadwal.

5.1.2.2 Skenario pengujian

Skenario pengujian merupakan runtutan langkah-langkah yang dilakukan dalam proses pengujian. Pada pengujian perangkat lunak penyiraman tanaman otomatis telah dibuat beberapa skenario berdasarkan fungsional yang terdapat pada perangkat lunak. Adapun skenario pengujian yang telah dibuat untuk menguji fungsionalitas dari perangkat lunak penyiraman tanaman otomatis dapat dilihat pada tabel 5.5 di bawah

Tabel 5.5 Tabel skenario pengujian

5.1.2.3 Pengujian perangkat lunak

No	Kode Fungsional	Nama Fungsional	Skenario		Test Case	Expected Result
			Kode Skenario	Keterangan		
1.	FU - 01	Perangkat lunak harus dapat mengoperasikan penyiraman.	SK-01	Perangkat lunak harus dapat menyiram tanaman.	Mengklik tombol <i>toggle</i> untuk mengoperasikan penyiraman tanaman.	Perangkat penyiram tanaman berhasil menyiram tanaman.
2.	FU - 02	Perangkat penyiraman tanaman dapat menyiram dari dua sumber air	SK-02	Perangkat penyiram tanaman harus dapat menyiram dari dua sumber air.	Mengoperasikan kran air 1.	Perangkat penyiram tanaman dapat menyiram tanaman dari dua sumber air.
					Mengoperasikan kran air 2.	
3.	FU - 03	Perangkat lunak harus dapat mengecek suhu tanaman.	SK-03	Perangkat lunak harus dapat mengetahui informasi kelembaban pada tanaman.	Mengambil informasi suhu dari sensor kelembaban pada perangkat penyiraman, lalu informasi suhu ditampilkan pada halaman dashboard perangkat lunak.	Perangkat lunak dapat menampilkan status suhu tanaman yang dibaca oleh sensor kelembaban.
4.	FU - 04	Perangkat lunak harus dapat mengatur jadwal penyiraman.	SK-04	Perangkat lunak harus dapat mengatur jadwal penyiraman tanaman.	Mengelola jadwal penyiraman dengan menambah, membaca, mengubah dan menghapus.	Perangkat lunak dapat melakukan Create, Read, Update dan Delete jadwal penyiraman tanaman.
5.	FU - 05	Perangkat lunak harus dapat menyiram tanaman sesuai jadwal yang telah ditentukan.	SK-05	Perangkat lunak harus dapat menyiram tanaman sesuai jadwal yang telah ditentukan.	Mengambil jadwal penyiraman tanaman yang sudah ditentukan.	Perangkat lunak dapat melakukan penyiraman tanaman sesuai jadwal yang ditentukan.

Pada bagian ini akan dipaparkan mengenai kasus-kasus pengujian yang dilakukan dan hasil dari setiap kasus pengujian. Untuk lebih kelas mengenai proses pengujian dan hasil yang didapatkan dapat dilihat pada tabel 5.6 di bawah

5.1.2.4 Pengujian perangkat lunak

Pada bagian ini akan dipaparkan mengenai kasus-kasus pengujian yang dilakukan dan hasil dari setiap kasus pengujian. Untuk lebih jelasnya mengenai proses pengujian dan hasil yang didapatkan dapat dilihat pada tabel 5.6 di bawah

Tabel 5.6 Tabel Skenario Pengujian Perangkat Lunak

No	Skenario		Test Case	Pre Condition	Test Steps	Test Data	Expected Result	Actual Result	PASS/FAIL
	Kode Skenario	Keterangan							
1.	SK-01	Perangkat lunak harus dapat menyiram tanaman.	Menyimpan jadwal penyiraman pada taman depan.	Pengguna sudah menyimpan jadwal penyiraman.	Pengguna mengatur jadwal penyiraman pada taman depan.	Id_penyiraman: 01	Kran air 1 menyiram.	Perangkat lunak belum terhubung dengan perangkat penyiraman.	PASS
2.	SK-02	Perangkat penyiraman tanaman dapat menyiram dari dua sumber air	Menyimpan jadwal penyiraman pada taman depan.	Pengguna sudah menyimpan jadwal penyiraman.	Pengguna mengatur jadwal penyiraman pada taman depan.	Id_penyiraman: 01	Kran air 1 menyiram.	Perangkat lunak belum terhubung dengan perangkat penyiraman.	PASS
			Menyimpan jadwal penyiraman pada taman belakang.	Pengguna sudah menyimpan jadwal penyiraman.	Pengguna mengatur jadwal penyiraman pada taman belakang.	Id_penyiraman: 02	Kran air 2 menyiram.	Perangkat penyiraman berhasil menyiram.	PASS
3.	SK-03	Perangkat lunak harus dapat mengetahui informasi kelembaban pada tanaman.	Mengambil informasi kelembaban dari <i>soil moisture sensor</i> pada perangkat penyiraman, lalu informasi suhu ditampilkan.	Pengguna sudah mengakses halaman <i>dashboard</i> .	Pengguna mendapatkan informasi kelembaban.	68% (Lembab Cukup)	Menampilkan nilai kelembaban.	Menampilkan informasi kelembaban.	PASS
				Pengguna sudah mengakses IDE Arduino dan melakukan <i>running</i> program.		30% (Kering)		Menampilkan informasi kelembaban.	PASS
				98% (Basah)		Menampilkan informasi kelembaban.		PASS	
4.	SK-04	Perangkat lunak harus dapat mengatur	Mengelola jadwal penyiraman dengan menambah, membaca,		Pengguna mendapatkan informasi pada	Id_jadwal, jadwal_penyiraman,	Menampilkan informasi jadwal penyiraman.	Menampilkan informasi jadwal penyiraman.	PASS

No	Skenario		Test Case	Pre Condition	Test Steps	Test Data	Expected Result	Actual Result	PASS/ FAIL
	Kode Skenario	Keterangan							
		jadwal penyiraman tanaman.	mengubah dan menghapus.	Pegguna sudah mengakses halaman pengelolaan jadwal.	tabel jadwal penyiraman.	taman			
					Pegguna menginputkan jadwal penyiraman.	jadwal_penyiraman (hh:mm:ss), taman (string)	Menyimpan jadwal penyiraman.	Menyimpan jadwal penyiraman.	PASS
					Pegguna menghapus data jadwal penyiraman.	Id_jadwal	Menghapus jadwal penyiraman yang dipilih.	Menghapus jadwal penyirama yang dipilih.	PASS
				Pegguna telah mengklik tombol update, lalu diarahkan ke halaman update jadwal.	Pegguna melakukan update data jadwal penyiraman.	Id_jadwal, jadwal_penyiraman (hh:mm:ss), taman (string).	Memperbaharui jadwal penyiraman yang dipilih.	Memperbaharui jadwal penyiraman yang dipilih.	PASS
5.	SK-05	Perangkat lunak harus dapat menyiram tanaman sesuai jadwal yang telah ditentukan.	Mengambil jadwal penyiraman tanaman yang sudah ditentukan.	Pegguna sudah menyimpan jadwal penyiraman, lalu perangkat lunak mengecek jadwal yang telah ada.	Perangkat lunak mengecek jadwal otomatis, lalu melakukan penyiraman.	Relay Aktif	Melakukan penyiraman sesuai jadwal yang ada.	Menyiram tanaman sesuai jadwal yang telah ditentukan.	PASS

BAB 6

KESIMPULAN DAN SARAN

Bab ini menjelaskan mengenai kesimpulan yang didapatkan dari hasil penelitian yang dilakukan serta saran yang diberikan penulis untuk pengembangan penelitian selanjutnya.

6.1. Kesimpulan

Berdasarkan Hasil Implementasi dan pengujian pada perangkat lunak penyiraman tanaman otomatis dengan mengacu pada analisis yang telah dibuat, maka dapat diambil kesimpulan sebagai berikut :

1. Telah merancang dan merealisasikan perangkat penyiraman tanaman otomatis dengan memanfaatkan mikrokontroller *Wemos D1* dengan fungsional sebagai berikut:
 - a. Dapat dikontrol melalui *WiFi* secara lokal.
 - b. Dapat melakukan penyiraman melalui 2 *relay*.
 - c. Dapat melakukan penyiraman otomatis sesuai jadwal yang telah ditentukan.
2. Telah merancang dan merealisasikan perangkat lunak untuk sistem penyiraman tanaman otomatis berbasis *website*:
 - a. Dapat mengelola jadwal penyiraman tanaman.
 - b. Dapat menampilkan status kelembaban tanah.
 - c. Dapat terhubung dengan perangkat keras mikrokontroller melalui *WiFi* lokal.

6.2. Saran

Berdasarkan tahapan yang telah direalisasikan pada penelitian ini, diharapkan dapat menjadi dasar penelitian lebih lanjut mengingat masih banyaknya keterbatasan yang dihadapi maka diusulkan beberapa saran pengembangan yaitu :

- a. Mensederhanakan penjadwalan melalui modul tambahan pada perangkat keras yaitu RTC (*Real Time Clock*) guna untuk perangkat keras mengenal waktu *real time*.
- b. Melakukan pengembangan dengan board yang memiliki pin lebih banyak, untuk menambahkan berbagai sensor lainnya.
- c. Menambahkan fitur pengaturan *enable* dan *disable* penyiraman otomatis ketika tanah kering pada perangkat lunak.
- d. Menambahkan fitur pengaturan lama waktu menyiram pada perangkat lunak.

DAFTAR PUSTAKA

- [AND11] Andika Radenal, "Penerapan CI (Codeigniter) Dalam Pengembangan Sistem Informasi Manajemen Surat Dan Pengarsipan", UIN Syarif Hidayatullah Jakarta, 2011.
- [FAI16] Faisal Faskal Imam, "Sistem Kontrol Dan Monitoring Bendungan Berbasis Raspberry PI", Universitas Widyatama, 2016.
- [HEN14] Hendrianto Dani Eko, "Pembuatan Sistem Informasi Perpustakaan Berbasis Website Pada Sekolah Menengah Pertama Negeri 1 Donorojo Kabupaten Pacitan", Indonesian Journal on Networking and Security (IJNS) – Vol 3 No. 4 2014.
- [JUW11] Juwita Adistya Anastasya, "Perancangan Program Aplikasi Perangkat Lunak Membuat Image Paint Menggunakan Microsoft Visual Basic 6.0", Universitas Sumatera Utara, 25 Juli 2011.
- [KRE01] Kreger Heater, "Web services Conceptual Architecture", IBM Software Group, 2001.
- [NUR15] Nuryadi Agus, "Prototipe Penyiraman Tanaman Otomatis Tanaman Cabai Berbasis Mikrokontroler ATMEGA16" UIN Yogyakarta, 2015.
- [PRA15] Pranata, T, Irawan B, & Ilhamsyah, "Penerapan Logika Fuzzy Pada Sistem Penyiraman Tanaman Otomatis Berbasis Mikrokontroler", Jurnal Coding, Sistem Komputer Untan Vol 3 No. 2, 2015.
- [PRE15] *Roger S, Pressman, Bruce R, Maxim, 'Software Engineering A Practicioner's Approach Eighth Edition', McGraw-Hill 2015.*
- [PRI13] Priyanto Sihno, "Purwarupa Sistem Penyiraman Tanaman Otomatis Berbasis Sensor Kelembaban Tanah Dan Arduino UNO", Universitas Gadjah Mada, 2013.
- [RAT17] Ratnawati, Silma, "Sistem Kendali Penyiram Tanaman Menggunakan *Propeller* Berbasis *Internet Of Things*", Jurnal *Inspiration*, STMIK AKBA Vol 7 No. 2, 2017.
- [SAR17] Sari Weny Amanda, 'Sistem Penyiraman Tanaman Otomatis Dengan Menggunakan Sensor Kelembaban Tanah Berbasis Arduino', Universitas Sumatera Utara, 13 October 2017.
- [SUG17] Sugiono, Indriyani Tutuk, Ruswiansari Maretha, "Kontrol Jarak Jauh Sistem Irigasi Sawah Berbasis *Internet Of Things* (IoT)", Integer: Journal of Information Technology, Vol 2 No. 2 2017.
- [TRI13] Triseptian Anastasius, "Pengembangan Sistem Informasi Tugas Akhir Berbasis Web Untuk Sistem Informasi Akademik Universitas Atma Jaya Yogyakarta", Universitas Atma Jaya Yogyakarta, 21 Juni 2013.
- [UPT07] Upton David, 'Codeigniter for Rapid PHP Application Development', Birmingham: Packt Publishing, 2007.
- [WIR17] Wiradhika Erwins, Purbasari Ayi, Gusdya Wanda, 'Pembangunan Perangkat Lunak Penyiraman Tanaman Secara Otomatis Berbasis Android', Universitas Pasundan, 16 Juni 2017.

LAMPIRAN

Lampiran A

A.1 Penulisan kode program Bahasa pemrograman PHP

Untuk penulisan kode program pada Bahasa pemrograman php berada pada *controller* yang telah ditentukan, pada Bahasa pemrograman php file ekstensi yang digunakan adalah .php. Pada penulisan code program pada Bahasa pemrograman php terdapat cara penulisan variable, operator, tipe data, array, pengkondisian, pengulangan, method.

A.1.1 Fungsi yang terdapat pada *controller* utama masterController

Berikut ini merupakan fungsi pemanggilan kelas *view* atau fungsi link membuka halaman *web* akan dipaparkan pada Gambar A.1

```
public function login(){
    $this->load->view("view_login"); //, $load
}

public function dashboard(){
    $data['title'] = "Dashboard";
    $data['jadwal_siram'] = $this->db->query('SELECT jadwal_penyiraman.id_jadwal, jadwal_penyiraman.jadwal_penyiraman,
    jadwal_penyiraman.taman from jadwal_penyiraman');

    $this->load->view("header", $data);
    $this->load->view("sidebar");
    $this->load->view("view_dashboard");
    $this->load->view("footer");
}

public function penyiraman(){
    $data['title'] = "Penyiraman Tanaman";

    $this->load->view("header", $data);
    $this->load->view("sidebar");
    $this->load->view("view_penyiraman");
    $this->load->view("footer");
}

public function pengelolaanJadwal($message = -1, $messageData=""){
    $data['title'] = "Pengelolaan Jadwal";

    $this->load->model('Jadwal_penyiraman');
    $data['Jadwal_penyiraman'] = $this->Jadwal_penyiraman->get_all();
    $data['messageData'] = $messageData;
    $this->load->view("header", $data);
    $this->load->view("sidebar");
    $this->load->view("view_pengelolaan_jadwal",$data);
    if($message > 0)
        $this->load->view('sukses', $data);
    else if($message == 0)
        $this->load->view('gagal', $data);
    $this->load->view("footer");
}
```

Gambar A.1 fungsi pemanggilan kelas *view*

Pada perangkat lunak penyiraman tanaman otomatis ini terdapat empat halaman yaitu diantaranya halaman awal *view_login*, halaman utama *view_dashboard*, halaman penyiraman *view_penyiraman* dan halaman pengelolaan jadwal *veiw_pengelolaan_jadwal*.

Berikut ini merupakan fungsi *viewJadwal* pada *masterController* akan dipaparkan pada Gambar 0.2 di bawah

```

//jadwal_penyiraman
function viewJadwal($message = -1, $messageData=""){
    $data['active'] = "Jadwal_penyiraman";
    $this->load->model('Jadwal_penyiraman');
    $data['Jadwal_penyiraman'] = $this->Jadwal_penyiraman->get_all();
    $data['messageData'] = $messageData;

    $this->load->view("header", $data);
    $this->load->view("sidebar");
    $this->load->view("view_pengelolaan_jadwal",$data);
    if($message > 0)
        $this->load->view('sukses', $data);
    else if($message == 0)
        $this->load->view('gagal', $data);
    $this->load->view("footer");
}

```

Gambar A.2 fungsi viewJadwal

Fungsi viewJadwal ini untuk menampilkan data jadwal penyiraman yang disimpan pada *database*.

Berikut ini merupakan fungsi insertJadwal pada masterController akan dipaparkan pada Gambar 0.3

```

function insertJadwal(){
    $data['active'] = "Jadwal_penyiraman";
    $this->load->model('Jadwal_penyiraman');
    $message = $this->Jadwal_penyiraman->insert($_POST);

    $this->viewJadwal($message, "diinputkan");
}

```

Gambar A.3 fungsi insertJadwal

Fungsi insertJadwal ini untuk menyimpan data jadwal penyiraman pada *database*.

Berikut ini merupakan fungsi formUpdateJadwal pada masterController akan dipaparkan pada Gambar 0.4

```

function formUpdatejadwal($id_jadwal){
    $data['title'] = "Update Jadwal";
    $data['active'] = "Jadwal_penyiraman";
    $this->load->model('Jadwal_penyiraman');
    $data['Jadwal_penyiraman'] = $this->Jadwal_penyiraman->get($id_jadwal);

    $this->load->view("header", $data);
    $this->load->view("sidebar");
    $this->load->view("view_admin/updateJadwal",$data);
    $this->load->view('footer');
}

```

Gambar A.4 fungsi formUpdateJadwal

Fungsi formUpdateJadwal ini untuk membuka halaman updatejadwal.

Berikut ini merupakan fungsi updatetJadwal pada masterController akan dipaparkan pada Gambar 0.5 di bawah


```
function updateJadwal(){
    $data['active'] = "Jadwal penyiraman";
    $this->load->model('Jadwal_penyiraman');
    $message = $this->Jadwal_penyiraman->update($_POST);

    $this->viewJadwal($message,"diupdate");
}
```

Gambar A.5 fungsi updateJadwal

Fungsi updateJadwal ini untuk mengubah data jadwal penyiraman pada *database*.

Berikut ini merupakan fungsi deleteJadwal pada masterController akan dipaparkan pada Gambar 0.6

```
function deleteJadwal($id_jadwal){
    $data['active'] = "Jadwal penyiraman";
    $this->load->model('Jadwal_penyiraman');
    $message = $this->Jadwal_penyiraman->delete($id_jadwal);

    $this->viewJadwal($message,"dihapus");
}
```

Gambar A.6 fungsi deleteJadwal

Fungsi deleteJadwal ini untuk menghapus data jadwal penyiraman pada *database*.

A.1.2 Fungsi yang terdapat pada *controller* Auth

Pada bagian ini merupakan bagian validasi *login* pengguna pada halaman view_login

Berikut ini merupakan fungsi logged_in_check pada *controller* Auth akan dipaparkan pada Gambar 0.7

```
public function logged_in_check()
{
    if ($this->session->userdata("logged_in")) {
        redirect("controller_admin");
    }
}

public function index()
{
    $this->logged_in_check();

    $this->load->library('form_validation');
    $this->form_validation->set_rules("username", "Username", "trim|required");
    $this->form_validation->set_rules("password", "Password", "trim|required");
    if ($this->form_validation->run() == true)
    {
        $this->load->model('auth_model', 'auth');
        // check the username & password of user
        $status = $this->auth->validate();
        if ($status == ERR_INVALID_USERNAME) {
            $this->session->set_flashdata("error", "Username is invalid");
        }
        elseif ($status == ERR_INVALID_PASSWORD) {
            $this->session->set_flashdata("error", "Password is invalid");
        }
        else
        {
            // success
            // store the user data to session
            $this->session->set_userdata($this->auth->get_data());
            $this->session->set_userdata("logged_in", true);
            // redirect to dashboard
            redirect("controller_admin");
        }
    }

    // $this->load->view("header");
    $this->load->view("view_login"); // View Login Form
    // $this->load->view("footer");
}
```

Gambar A.7 fungsi logged_in_check

Fungsi `logged_in_check` ini merupakan fungsi pengecekan identitas *login* pengguna yang akan divalidasi *username* dan *passwordnya*.

Berikut ini merupakan fungsi `logout` pada *controller Auth* akan dipaparkan pada Gambar 0.8

```
public function logout()
{
    $this->session->unset_userdata("logged_in");
    $this->session->sess_destroy();
    redirect("auth");
}
```

Gambar A.8 fungsi `deleteJadwal`

Fungsi `logout` ini berfungsi untuk menghapus *session* dan *me-logout* akun pengguna dari aplikasi.

A.1.3 Penulisan kode program pada Arduino IDE

Bahasa pemrograman yang digunakan untuk perangkat keras *Wemos D1* menggunakan bahasa dasar pemrograman yaitu C dengan memanfaatkan kakas pembangunan Arduino IDE.

Berikut ini merupakan konfigurasi utama yaitu melakukan konfigurasi koneksi *WiFi* akan dipaparkan pada Gambar A.9

```
-----KONFIGURASI UTAMA-----
// Konfigurasi koneksi Wifi

const char* ssid = "sugadewantha";
const char* password = "12345";
// Akhir konfigurasi Koneksi ke Wifi

//Informasi IP Server atau hostname;
char server[] = "192.168.0.102"; //eg: 192.168.0.222

WiFiClient client;

//WiFiServer server(80);
SoftwareSerial mySerial(8,9);

// sensor Soil at Analog pin A0
int sense_Pin = 0;
int value = 0;
```

Gambar A.9 konfigurasi *WiFi* pada *Wemos*

Konfigurasi ini berfungsi untuk menghubungkan perangkat keras *Wemos* ke *WiFi client*.

Berikut ini merupakan konfigurasi `void setup` pada perangkat keras *Wemos* akan dipaparkan pada Gambar A.10

```

//-----KONFIGURASI VOID SETUP-----

void setup()
{
  Serial.begin(115200);
  delay(10);
  mySerial.begin(115200);
  SPI.begin();

  pinMode(relay1, OUTPUT); //RED
  pinMode(relay2, OUTPUT); //blue

  // Menghubungkan ke Jaringan Wifi
  Serial.println();
  Serial.print("Menghubungkan ke ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi Terhubung");
  // Memanggil server
  // server.begin();

  Serial.println("STANDBY");
  Serial.print(WiFi.localIP());
  delay(1000);
  Serial.println("\nSIAP MENGOPERASIKAN SISTEM\n");
}

```

Gambar A.10 konfigurasi *void setup* pada Wemos

Konfigurasi ini berfungsi menginisialisasi variabel-variabel yang akan digunakan, dan hanya akan dijalankan satu kali saat *Wemos* mulai menyala untuk mengetahui bahwa sistem *Wemos* dapat dioperasikan dan *WiFi* terhubung.

Berikut ini merupakan konfigurasi *void loop* pada perangkat keras *Wemos* akan dipaparkan pada Gambar A.11 di bawah

```

//-----MEMULAI PENGIRIMAN DATA-----

void loop()
{
  // memeriksa klien jika terhubung
  Serial.print("MEMERIKSA SERVER\n");
  aplikasi(); //PROSEDUR MANIPULASI DATA
  return;
}

```

Gambar A.11 konfigurasi *void loop* pada Wemos

Konfigurasi ini berfungsi menjalankan suatu siklus program, yang akan dilakukan terus-menerus hingga *Wemos* mati atau *reset* untuk memeriksa apakah terhubung dengan *client*.

Berikut ini merupakan konfigurasi *database* pada perangkat keras *Wemos* akan dipaparkan pada Gambar A.11 di bawah

```
//-----FUNGSI PENGHUBUNG KE MYSQL-----
void aplikasi()
{
    value= analogRead(sense_Pin);
    value= value/10;

    if (client.connect(server, 80)) {
        Serial.println("Terhubung ke Aplikasi\n");

        // Meminta protokol HTTP:
        Serial.println("GET /HOST_FOLDER/Update/sensor_soil?q="); //TAMPILAN PADA PC (KABEL SERIAL)
        client.print("GET /HOST_FOLDER/Update/sensor_soil?q="); //AKSES SEBAGAI KLIEN

        //STATUS
        Serial.println(value);
        client.print(value);
        Serial.println("\n\nDATA TERKIRIM\n");
        if(value<50)
        {
            Serial.println("LEMBAB");
        }
        else
        {
            Serial.println("KERING"); //JIKA KERING, SISTEM MEMERINTAHKAN MEMBUKA RELAY UNTUK KERAN AIR
            digitalWrite(relay1, HIGH);
            delay(10000);
            digitalWrite(relay1, LOW);
            delay(1000);
        }

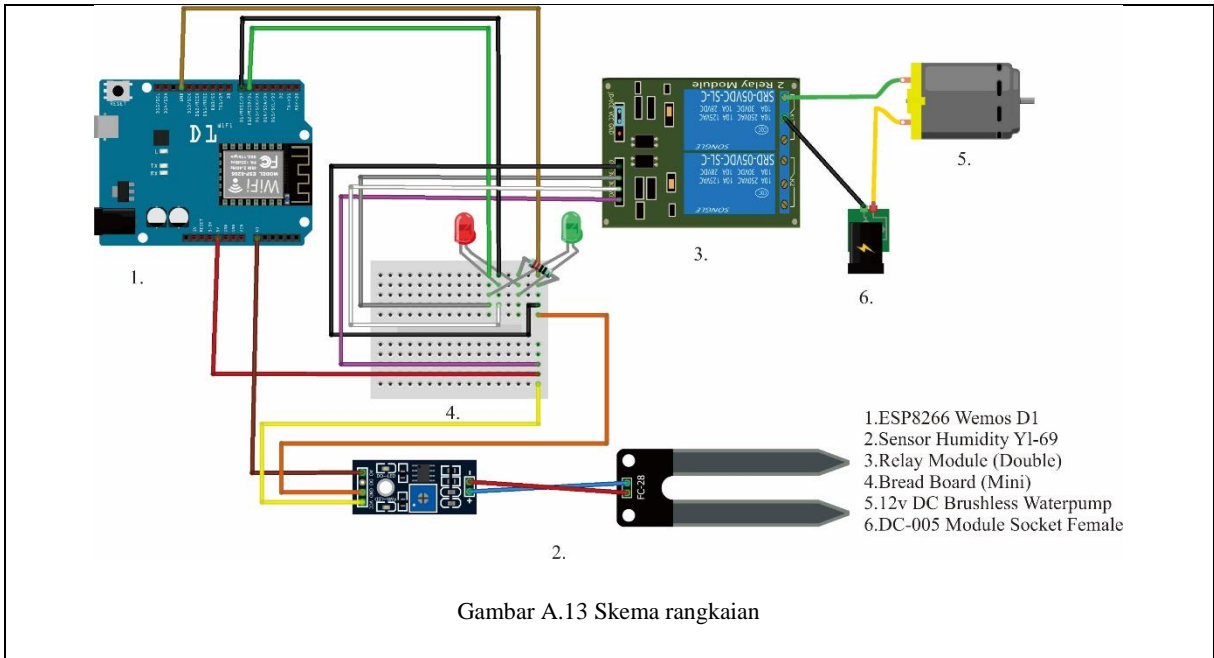
        client.print(" "); //SPACE BEFORE HTTP/1.1
        client.print("HTTP/1.1"); //HTTP REQUEST
        client.println();
        client.println("Host: 192.168.0.102");//eg: 192.168.0.222
        client.println("Connection: close");
        client.println();
    } else {
        // Kegagalan menghubungkan alat kedalam aplikasi pada server:
        Serial.println("Gagal Menghubungkan ke Server Aplikasi\n");
        delay(1000);
    }
    client.stop();
}
```

Gambar A.12 konfigurasi *void loop* pada *Wemos*

Konfigurasi ini berfungsi untuk menghubungkan perangkat keras ke *database* MySQL dan mengirimkan status kelembaban tanah.

A.2 Skema *Wemos D1 ESP8266*

Berikut ini skema rangkaian pada perangkat keras *Wemos* akan dipaparkan pada Gambar A.13 di bawah

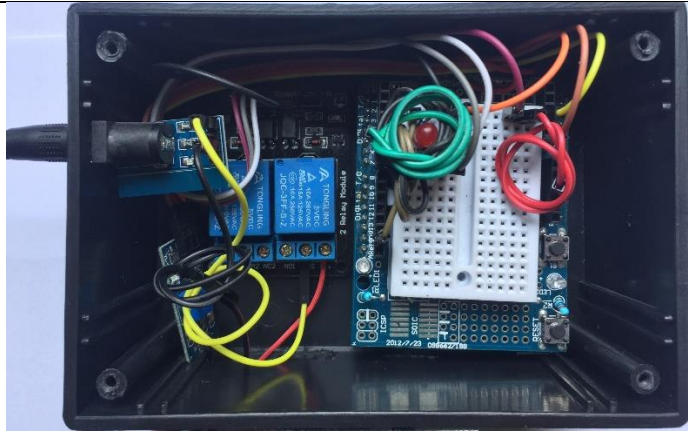


A.3 Gambar hasil akhir perangkat keras penyiraman tanaman otomatis

Berikut ini merupakan gambar hasil akhir perangkat keras penyiraman tanaman otomatis tampak luar, akan dipaparkan pada Gambar A.14 berikut ini :



Berikut ini merupakan gambar hasil akhir perangkat keras penyiraman tanaman otomatis tampak dalam, akan dipaparkan pada Gambar A.15 di bawah ini :



Gambar A.15 perangkat keras penyiraman tanaman otomatis tampak dalam

