



HEART & MIND TOWARDS EXCELLENCE

**RANCANG BANGUN KONTROL PENYIRAM TAMAN OTOMATIS DAN
MONITORING KELEMBABAN TANAH MENGGUNAKAN ANDROID**

TUGAS AKHIR

Program Studi

S1 Sistem Komputer

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

Oleh :

ACHMAD MUFADLOL ROMADHONI

14410200049

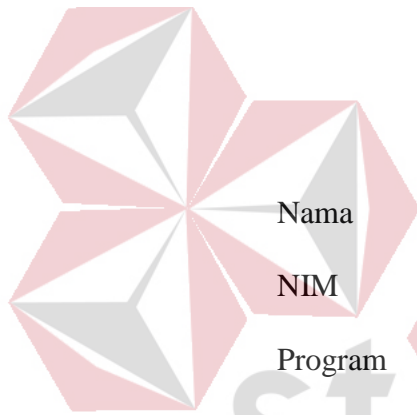
FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

2019

**RANCANG BANGUN KONTROL PENYIRAM TAMAN OTOMATIS DAN
MONITORING KELEMBABAN TANAH MENGGUNAKAN ANDROID
TUGAS AKHIR**

Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Komputer



Disusun Oleh :

Nama : Achmad Mufadlol Romadhoni

NIM : 14.41020.0049

Program : S1 (Strata Satu)

Jurusan : Sistem Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

2019



“Usaha Besar + Doa Besar = Hasil Besar”

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

Syukur Alhamdulillah, Segala Puji Bagi ALLAH SWT Shalawat dan salam tidak lupa selalu tcurahkan kepada Baginda Rasullulah Muhammad SAW.

Akhirya penulis dapat menyelesaikan Tugas Akhir ini,

Tugas Akhir ini saya persembahkan kepada

Ayah, Ibu, Kakak dan semua Keluarga yang selalu meberi dukungan dan mendoakan saya.

Terima kasih kepada Dosen Pembimbing

Serta para dosen – dosen yang memberikan ilmu dan memberikan motivasi kepada saya.

Terima kasih kepada pacar saya Iqlimatus Sa'adah sudah memberikan semangat dan dapat memahami keadaan.

Untuk seluruh rekan – rekan di S1 Sistem Komputer dan dikampus Institut Bisnis dan Informatika Stikom Surabaya yang membantu dan memberikan motivasi kepada saya.

Beserta semua orang yang telah membantu.

TUGAS AKHIR
RANCANG BANGUN KONTROL PENYIRAM TAMAN OTOMATIS DAN
MONITORING KELEMBABAN TANAH MENGGUNAKAN ANDROID

Dipersiapkan dan disusun oleh
Achmad Mufadlol Romadhoni
NIM : 14410200049


Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

Pada : Januari 2019

Susunan Dewan Penguji

Pembimbing

- I. **Harianto, S.Kom., M.Eng.**
NIDN. 0722087701
- II. **Yosefine Triwidayastuti, M.T.**
NIDN. 0729038504

 31/1/2019



Pembahas

- I. **Heri Pratikno, M.T., MTCNA., MTCRE.**
NIDN. 0716117302



Tugas Akhir in diterima sebagai salah satu persyaratan
Untuk memperoleh gelar Sarjana




FAKULTAS TEKNOLOGI
DAN INFORMATIKA

stikom
SURABAYA

Dr. Jusak

Dekan Fakultas Teknologi dan Informatika

 1/19
1/2

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Achmad Mufadlol Romadhoni
NIM : 14410200049
Program Studi : S1 Sistem Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **RANCANG BANGUN KONTROL PENYIRAM TAMAN OTOMASTIS DAN MONITORING KELEMBABAN TANAH MENGGUNAKAN ANDROID**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.



Surabaya, 28 Januari 2019

Yang menyatakan

Achmad Mufadlol Romadhoni

Nim : 14410200049

ABSTRAK

Penyiraman adalah salah satu faktor penting dalam produksi tanaman. Keberhasilan yang terjadi dalam sistem penanaman akan tercapai apabila diatur waktu dan kelembaban tanah pada tanaman. Tanpa perawatan intensif tanaman bisa saja mati. Maka dari itu butuh perhatian khusus untuk mengoptimalkan pertumbuhannya. Tanaman yang sehat harus diikuti dengan kondisi tanah yang baik. Kondisi tersebut adalah nilai kelembaban yang ideal dan penyiraman dengan waktu yang tepat.

Pada tugas akhir ini dibuatlah rancang bangun kontrol penyiram taman otomatis dan monitoring kelembaban tanah menggunakan Android. Alat ini berfungsi untuk menyiram taman secara otomatis berdasarkan kelembaban tanah dan waktu penyiraman tanaman yang ideal. Alat ini juga dapat memonitoring dari jauh menggunakan Android apakah penyiram tanaman tersebut berjalan dengan baik atau tidak, serta mikrokontroller *Wemos D1 R2* dan sensor *Soil Moisture Sensor* dan *Real Time Clock*.

Pada pengujian monitoring jarak jauh dengan jarak 36 km menunjukkan 18% data pengujian monitoring yang tidak mengalami *delay* antara *serial monitor* dan *LCD* dengan aplikasi *Blynk*. Terdapat 82% data pengujian mengalami *delay* melalui sistem ini ke aplikasi *Blynk*. Nilai rata – rata *delay* bernilai sebesar 0.88s. Notifikasi tingkat kelembaban dan proses penyiraman muncul pada jam 8 pagi dan 4 sore pada *Smartphone Android* dan *Email*.

Kata kunci: *Real Time Clock, Wemos D1 R2, Soil Moisture Sensor.*

KATA PENGANTAR

Pertama – tama penulis panjatkan puji dan syukur atas kehadiran Allah SWT, karena berkat izin, Rahmat dan hidayah-Nya penulis dapat menyelesaikan laporan penelitian ini dengan judul “RANCANG BANGUN KONTROL PENYIRAM TAMAN DAN MONITORING KELEMBABAN TANAH MENGGUNAKAN ANDROID” yang merupakan salah satu syarat menempuh Tugas Akhir pada Program Studi S1 sistem Komputer di Institut Bisnis dan Informatika Stikom Surabaya. Sholawat serta salam tidak lupa selalu penulis panjatkan kepada Rasulullah SAW.

Di dalam buku Laporan Tugas Akhir ini melakukan pembahasan mengenai pembuatan control penyiram taman dan monitoring kelembaban tanah menggunakan Android. Harapan penulis semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan Laporan Tugas Akhir ini masih terdapat banyak kekurangan.

Dalam usaha menyelesaikan Laporan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak baik moral maupun materi. Oleh karena itu penulis mengucapkan terima kasih dan penghargaan setinggi – tingginya kepada:

1. Ayah, Ibu dan saudara – saudara saya tercinta yang telah memberikan dukungan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir maupun laporan ini.

2. Bapak Dr. Jusak selaku Dekan Fakultas Teknologi dan Informatika Stikom (FTI) Institut Bisnis dan Informatika Stikom Surabaya telah membantu proses penyelesaian Tugas Akhir yang dibuat oleh penulis dengan baik.
3. Kepada Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi Sistem Komputer Stikom Surabaya atas ijin yang diberikan untuk mengerjakan Tugas Akhir ini.
4. Kepada Bapak Harianto, S.Kom., M.Eng., dan Ibu Yosefine Triwidyastuti, M.T., selaku Dosen Pembimbing. Terima kasih atas bimbingan yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik.
5. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen Penguji memberi masukan dalam menyusun buku Tugas Akhir.
6. Semua staf dosen yang telah mengajar dan memberikan ilmunya.
7. Terima kasih terhadap seluruh rekan – rekan S1 Sistem Komputer khususnya rekan – rekan seperjuangan angkatan 2014 yang selalu memberikan semangat dan bantuannya.
8. Terima kasih kepada Iqlimatus Sa'adah selaku kekasih yang sudah memberi semangat dan motivasi sehingga penulis dapat menyelesaikan Tugas Akhir ini.
9. Serta semua pihak lain yang tidak dapat disebut secara satu per satu, yang telah membantu dalam menyelesaikan Tugas Akhir ini baik secara langsung maupun tidak langsung.

Surabaya, Januari 2019

DAFTAR ISI

ABSTRAK	vi
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Pembatasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	4
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Kadar Kelembaban Tanah	6
2.2 <i>Soil Moisture Sensor</i> (YL-69)	7
2.3 Wemos D1 R2	8
2.4 <i>Liquid Crystal Display</i> (LCD)	9
2.5 <i>Relay</i>	10

2.6	Internet	11
2.7	Android.....	13
2.8	Aplikasi <i>Blynk</i>	14
2.9	Real Time Clock.....	15
2.10	Pompa Air AC.....	16
2.11	Arduino Software IDE (<i>Integrated Development Environment</i>)	17
BAB III METODE PENELITIAN.....		20
3.1	Metode Penelitian.....	20
3.2	Model Perancangan	21
3.3	Perancangan Mekanik	22
3.4	Perancangan Elektronika.....	26
3.4.1	Rangkaian Tombol <i>Setting</i> RTC	27
3.4.2	Rangkaian LCD	28
3.4.3	Rangkaian RTC	30
3.4.4	Rangkaian <i>Relay</i> dan <i>Solenoid Valve</i>	31
3.4.5	Rangkaian <i>Soil Moisture Sensor</i>	32
3.5	Perancangan <i>Software</i>	33
3.5.1	<i>Flowchart</i> Penyiram Taman	33
3.5.2	Aplikasi Monitoring dan Kontrol	34
3.5.3	Konfigurasi Arduino IDE	54
3.5.4	Program Koneksi Wemos D1 R2 dengan <i>WiFi</i>	57
3.5.5	Program Wemos D1 R2 Untuk mengirim Data ke <i>Web Server</i>	58
3.5.6	Pengaturan Web Server	59

BAB IV HASIL PENGUJIAN DAN PENGAMATAN.....	61
4.1 Pengujian Wemos D1 R2	61
4.1.1 Tujuan Pengujian Wemos D1 R2	61
4.1.2 Alat Pengujian Wemos D1 R2	62
4.1.3 Prosedur Pengujian Wemos D1 R2	62
4.1.4 Hasil Pengujian Wemos D1 R2.....	62
4.2 Pengujian <i>Soil Moisture Sensor</i>	64
4.2.1 Tujuan Pengujian <i>Soil Moisture Sensor</i>	64
4.2.2 Alat Pengujian <i>Soil Moisture Sensor</i>	64
4.2.3 Prosedur Pengujian <i>Soil Moisture Sensor</i>	64
4.2.4 Hasil Pengujian <i>Soil Moisture Sensor</i>	65
4.3 Pengujian Notifikasi	66
4.3.1 Tujuan Pengujian Notifikasi.....	66
4.3.2 Alat Pengujian Notifikasi	66
4.3.3 Prosedur Pengujian Notifikasi.....	66
4.3.4 Hasil Pengujian Notifikasi.....	67
4.4 Pengujian <i>Relay</i> dan <i>Valve</i>	69
4.4.1 Tujuan Pengujian <i>Relay</i> dan <i>Valve</i>	69
4.4.2 Alat Pengujian <i>Relay</i> dan <i>Valve</i>	69
4.4.3 Prosedur Pengujian <i>Relay</i> dan <i>Valve</i>	70
4.4.4 Hasil Pengujian <i>Relay</i> dan <i>Valve</i>	70
4.5 Pengujian <i>Real Time Clock</i>	74
4.5.1 Tujuan Pengujian <i>Real Time Clock</i>	74

4.5.2	Alat Pengujian <i>Real Time Clock</i>	74
4.5.3	Prosedur Pengujian <i>Real Time Clock</i>	74
4.5.4	Hasil Pengujian <i>Real Time Clock</i>	75
4.6	Pengujian Monitoring Sensor Pada Aplikasi <i>Blynk</i>	76
4.6.1	Tujuan Pengujian Monitoring Sensor Pada Aplikasi <i>Blynk</i>	76
4.6.2	Alat Pengujian Monitoring Sensor pada Aplikasi <i>Blynk</i>	76
4.6.3	Prosedur Pengujian Monitoring Sensor pada Aplikasi <i>Blynk</i>	77
4.6.4	Hasil Pengujian Monitoring Sensor pada Aplikasi <i>Blynk</i>	77
4.7	Pengujian Monitoring Jarak Jauh Alat dengan Smartphone Android	80
4.7.1	Tujuan Monitoring Jarak Jauh Alat dengan Smartphone Android	80
4.7.2	Alat Pengujian Monitoring Jarak Jauh Alat dengan <i>Smartphone Android</i>	80
4.7.3	Prosedur Pengujian Monitoring Jarak Jauh Alat dengan <i>Smartphone Android</i>	80
4.7.4	Hasil Pengujian Monitoring Jarak Jauh Alat dengan <i>Smartphone Android</i>	81
BAB V KESIMPULAN		87
5.1	Kesimpulan	87
5.2	Saran	87
DAFTAR PUSTAKA		88
LAMPIRAN		90
BIODATA PENULIS		97

DAFTAR GAMBAR

Gambar 2.1 <i>Soil Moisture Sensor</i> (YL - 69)	7
Gambar 2. 2 Wemos D1 R2	8
Gambar 2.3 <i>Datasheet</i> Wemos D1 R2.....	9
Gambar 2.4 LCD.....	9
Gambar 2. 5 <i>Relay</i>	10
Gambar 2.6 Aplikasi <i>Blynk</i>	14
Gambar 2.7 <i>Real Time Clock</i>	15
Gambar 2.8 Pompa Air AC.....	17
Gambar 2.9 Arduino IDE.....	19
Gambar 3.1 Blok Diagram.....	21
Gambar 3.2 Desain Rancang Mekanik Tampak Keseluruhan	23
Gambar 3.3 Hasil Rancang Mekanik Bagian Luar	23
Gambar 3.4 Hasil Rancang Mekanik Bagian Dalam	24
Gambar 3.5 Perancangan Elektronik.....	26
Gambar 3.6 Rangkaian Tombol <i>Setting</i> RTC	27
Gambar 3.7 Rangkaian LCD I2C.....	29
Gambar 3.8 Rangkaian RTC	30
Gambar 3.9 Rangkaian <i>Relay</i> dan <i>Solenoid Valve</i>	31
Gambar 3.10 Rangkaian <i>Soil Moisture Sensor</i>	32

Gambar 3.11 <i>Flowchart</i> Penyiram Taman.....	34
Gambar 3.12 Tampilan <i>Creat New Account</i>	35
Gambar 3. 13 Tampilan <i>log in</i>	36
Gambar 3.14 Tampilan <i>New Project</i>	37
Gambar 3.15 Tampilan <i>Creat New Project</i>	37
Gambar 3.16 <i>Auth Token</i> yang dikirim ke <i>Gmail</i>	38
Gambar 3.17 <i>Auth Token</i> yang diterima oleh <i>Gmail</i>	39
Gambar 3.18 Tampilan Awal dan <i>Widget Box</i>	40
Gambar 3. 19 Tampilan Aplikasi <i>Blynk</i>	40
Gambar 3.20 <i>Setting Gauge</i>	42
Gambar 3.21 Program Kelembaban Tanah.....	42
Gambar 3.22 <i>Setting Value Display</i>	43
Gambar 3.23 Program RTC	44
Gambar 3.24 <i>Setting Syled Button</i>	45
Gambar 3. 25 <i>Setting Notification</i>	47
Gambar 3.26 Program Notifikasi	48
Gambar 3.27 <i>Setting Email</i>	49
Gambar 3.28 Program Notifikasi Email	50
Gambar 3.29 <i>Setting SuperChart</i>	51
Gambar 3.30 <i>Setting Waktu SuperChart</i>	52
Gambar 3. 31 Program Kelembaban Tanah.....	53
Gambar 3.32 Instalasi addon ESP8266 dengan <i>Arduino Board Manager</i>	54
Gambar 3.33 <i>Instal Wemos Board Manager</i>	55

Gambar 3.34 Pemilihan <i>Board Wemos D1 R2</i>	55
Gambar 3. 35 <i>Instal Library Blynk</i>	56
Gambar 3.36 <i>Manager Library</i>	56
Gambar 3. 37 <i>Include Library Blynk</i>	57
Gambar 3.38 Kode Koneksi <i>Wemos D1 R2</i> dengan <i>WiFi</i>	58
Gambar 3.39 Kode Mengirim Data <i>Web Server</i> pada <i>Wemos D1 R2</i>	59
Gambar 3.40 <i>Blynk Cloud</i>	60
Gambar 4.1 Tampilan Proses <i>Upload</i> dari <i>Arduino IDE</i> Kepada <i>Wemos D1R2</i>	63
Gambar 4.2 Tampilan <i>Comment</i> Saat Berhasil <i>Upload</i> Kepada <i>Wemos D1 R2</i>	63
Gambar 4.3 Pembacaan <i>Soil Moisture Sensor</i> Pada <i>Serial Monitor</i>	65
Gambar 4.4 Notifikasi pada <i>Email</i>	68
Gambar 4.5 Notifikasi pada <i>Smartphone Android</i>	68
Gambar 4.6 <i>Relay</i> Bernilai <i>High / 1</i>	71
Gambar 4. 7 Kontrol <i>Valve ON</i>	71
Gambar 4.8 Terjadi Proses Penyiraman.....	72
Gambar 4.9 <i>Relay</i> Bernilai <i>Low / 0</i>	72
Gambar 4.10 Kontrol <i>Valve OFF</i>	73
Gambar 4.11 Tidak Terjadi Proses Penyiraman	73
Gambar 4.12 Tampilan Proses Pembacaan <i>Real Time Clock</i> Pada <i>Serial Monitor</i>	75
Gambar 4.13 Tampilan Data pada <i>LCD</i>	78
Gambar 4.14 Tampilan Data pada Aplikasi <i>Blynk</i>	78
Gambar 4.15 Tampilan Data pada <i>LCD</i>	82
Gambar 4.16 Tampilan Data pada Aplikasi <i>Blynk</i>	82

Gambar 4.17 Jarak Monitoring antara Alat dan Aplikasi 83

Gambar 4.18 Posisi Alat dan Aplikasi *Blynk* 83

DAFTAR TABEL

Tabel 2.1 Presentase Kelembaban Tanah..... 6

Tabel 4. 1 Pengujian Monitoring pada Aplikasi *Blynk* 79

Tabel 4.2 Pengujian Monitoring pada Alat Secara Jarak Jauh..... 84



BAB I

PENDAHULUAN

1.1 Latar Belakang

Penyiraman adalah salah satu faktor penting dalam produksi tanaman. Keberhasilan yang terjadi dalam sistem penanaman akan tercapai apabila diatur waktu dan kelembaban tanah pada tanaman. Air di dalam tanah selalu membawa nutrisi dalam larutannya untuk menggantikan air yang hilang karena penguapan yang terjadi pada tanah dan transpirasi dari tanaman. Perawatan tanaman sangat berbeda dengan kepemilikan benda mati. Tanpa perawatan intensif tanaman bisa saja mati. Maka dari itu butuh perhatian khusus untuk mengoptimalkan pertumbuhannya. Tanaman yang sehat harus diikuti dengan kondisi tanah yang baik. Kondisi tersebut adalah nilai kelembaban yang ideal dan penyiraman dengan waktu yang tepat. Kelembaban tanah yang paling bagus adalah 65% dengan rentang 55%-75% (Riajaya P. D. dan Kadarwati T. F.,2007).

Selain kelembaban tanah waktu penyiraman juga harus diperhatikan, karena waktu yang baik untuk melakukan penyiraman adalah pada saat pagi hari sebelum jam 10.00 atau sore hari setelah pukul 17.00. Jadwal penyiraman juga perlu diperhatikan apakah satu kali sehari atau dua kali sesuai dengan kebutuhan masing-masing tanaman (Nasrullah, Trisanto & Tamin, 2011)

Pada jaman sekarang ini sering kali kita melihat orang melakukan penyiraman tanaman secara manual, banyak dengan menggunakan tenaga manusia seperti

penyiraman menggunakan ember dan selang penyemprot. Monitoring dengan menggunakan metode SMS merupakan cara yang tidak begitu bagus, dikarenakan menghabiskan biaya untuk SMS dan gangguan sinyal pada operator seluler. Cara lain yaitu dengan menggunakan *smartphone* yang terhubung dengan internet, tentu saja *smartphone* tersebut sudah terpasang aplikasi khusus untuk memantau keadaan pada taman tersebut.

Pada tugas akhir ini dibuatlah rancang bangun kontrol penyiram taman otomatis dan monitoring kelembaban tanah menggunakan Android. Alat ini berfungsi untuk menyiram taman secara otomatis berdasarkan kelembaban tanah dan waktu penyiraman tanaman yang ideal. Alat ini juga dapat memonitoring dari jauh menggunakan Android apakah penyiram tanaman tersebut berjalan dengan baik atau tidak. Alasan kenapa menggunakan aplikasi Android dikarenakan pada jaman sekarang pengguna *smartphone* sudah cukup banyak (keminfo. 2015).

Sensor yang digunakan adalah *soil moistur sensor* sebagai pengukur kelembaban tanah, *real time clock* digunakan sebagai pengatur waktu penyiraman, serta *solenoid valve* digunakan sebagai pengatur buka tutup laju air.

1.2 Rumusan Masalah

Adapun permasalahan yang akan dihadapi dalam pengerjaan tugas akhir ini diantaranya adalah:

1. Bagaimana cara mengolah data yang dihasilkan oleh sensor kelembaban tanah dan waktu agar dapat diproses oleh mikrokontroller?

2. Bagaimana cara mengolah data agar dapat dimonitoring secara jarak jauh menggunakan Android?
3. Bagaimana alat dapat memunculkan notifikasi darurat apabila sensor kelembaban tanah dan waktu tidak normal?

1.3 Pembatasan Masalah

Dalam perancangan dan pembuatan terdapat beberapa batasan masalah, antara lain:

1. Menggunakan Wemos D1 R2 sebagai microprosesor pengolah data,
2. Untuk sensor yang digunakan adalah sensor kelembaban tanah dan *Real Time Clock*.
3. Taman yang digunakan berukuran 1 x 1 meter,
4. Aplikasi yang digunakan untuk memonitoring hanya berjalan dengan Android.
5. Penggunaan alat dekat dengan sumber air dan listrik.

1.4 Tujuan

Penelitian ini bertujuan membuat sistem pengendalian otomatis untuk menyiram taman dengan menggunakan pengecekan terhadap waktu dan sensor kelembaban tanah serta alat dan sistem dapat dimonitoring secara jarak jauh serta diolah datanya menggunakan Android.

1.5 Manfaat

Manfaat dari penelitian rancang bangun kontrol penyiram taman otomatis dan monitoring menggunakan Android adalah mempermudah perawatan dan pengawasan dalam proses pemeliharaan taman berdasarkan kelembaban tanah dan waktu penyiraman.

1.6 Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, perumusan masalah, pembatasan masalah, tujuan, manfaat, serta sistematika penulisan pada Tugas Akhir.

BAB II LANDASAN TEORI

Bab ini menjelaskan tentang teori dasar penunjang penelitian yang secara singkat sebagai acuan penelitian Tugas Akhir.

BAB III METODE PENELITIAN

Bab ini membahas tentang tahapan dalam perancangan Tugas Akhir meliputi perancangan sistem, arsitektur mekanisme, arsitektur elektronik, dan arsitektur program.

BAB IV HASIL PENGUJIAN DAN PENGAMATAN

Bab ini membahas tentang pengujian dari masing-masing komponen yang digunakan, meliputi pengujian *soil moisture sensor*, *real time*

clock, pengujian aplikasi *blynk* pada *Android*, dan pengujian sistem keseluruhan.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilaksanakan serta saran untuk pengembangan dalam penelitian berikutnya.



BAB II

LANDASAN TEORI

2.1 Kadar Kelembaban Tanah

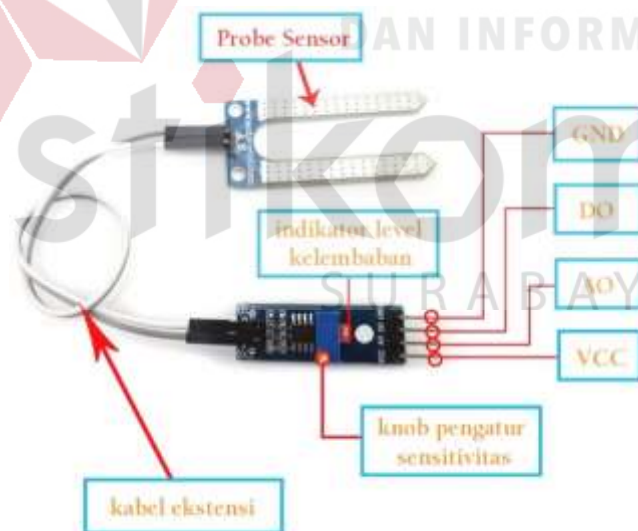
Kelembaban tanah adalah kadar air yang tertahan dalam tanah setelah proses penyiraman terjadi. Teknik untuk mengukur kelebihan air yang dapat dilakukan secara manual yaitu mengukur perbedaan berat tanah yang disebut dengan metode gravimetri dan secara langsung dapat dilakukan pengukuran sifat lain dari tanah. Teknik pengukuran gravimetri memiliki akurasi yang sangat tinggi namun membutuhkan waktu dan tenaga yang sangat besar untuk melakukannya. Standar kelembaban tanah yang baik dan sesuai untuk tanaman yaitu memiliki kadar air 50%-70%. Hal ini disebabkan tanaman membutuhkan air sebagai penunjang pertumbuhan alaminya (Pambudi, 2009). Presentase kelembaban tanah dapat dilihat pada Tabel 2.1.

Tabel 2.1 Presentase Kelembaban Tanah

Persen	Status Tanah
78,01%	Basah
71,65%	Basah
58,46%	Lembab
51,12%	Lembab
37,43%	Lembab
31,47%	Kering
29,13%	Kering
25,22%	Kering
38,42%	Lembab
78,01%	Basah

1.2 Soil Moisture Sensor (YL-69)

Pada Gambar 2.1 dibawah merupakan *Soil Moisture Sensor* yang dapat mendeteksi kelembaban tanah, sedangkan tanah yang kering sangat sulit menghantarkan listrik disekitarnya. Sensor ini terdiri dari dua *probe* untuk melewati arus listrik dalam tanah, kemudia membaca resistansinya untuk mendapatkan nilai tingkat kelembaban. Semakin banyak air membuat tanah lebih mudah menghantarkan listrik. Pada sensor ini sudah dilengkapi dengan potensiometer yang digunakan untuk mengatur tingkat sensitifitas dari sensor. Sensor ini sangat membantu mengingatkan tingkat kelembaban pada tanaman untuk memantau kelembaban tanah (Pambudi, 2009).



Gambar 2.1 *Soil Moisture Sensor* (YL - 69)

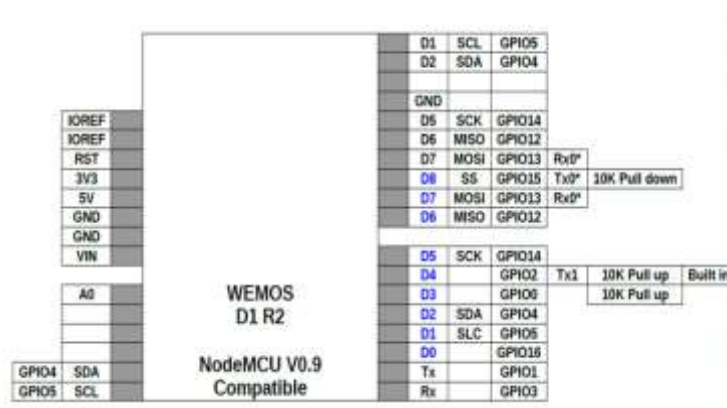
2.3 Wemos D1 R2

Wemos merupakan salah satu board yang dirancang khusus untuk keperluan IoT (*Internet of Thing*). Wemos menggunakan chip *WiFi* yang cukup terkenal yaitu ESP8266. Wemos memiliki beberapa kelebihan yaitu *Arduino compatible*, artinya dapat diprogram menggunakan Arduino IDE dengan sintaks program dan *library* yang banyak. Wemos D1 salah satu product yang memiliki bentuk seperti Arduino uno. Sehingga memudahkan kita untuk menghubungkan dengan Arduino shield lainnya.

Wemos dapat *running stand alone* tanpa perlu dihubungkan dengan mikrokontroler lainnya karena sudah terdapat CPU yang dapat diprogram melalui Serial port ataupun via OTA (*Over The Air*) atau mengirim program secara *wireless*. *High Frequency* CPU dengan processor utama 32bit berkecepatan 80MHz. Wemos dapat mengeksekusi program lebih cepat dibandingkan mikrokontroler 8bit yang digunakan di Arduino (Rudiawan Eko, 2016). Untuk lebih jelasnya dapat dilihat pada Gambar 2.2. dan pada Gambar 2.3 dijelaskan *datashhet* dari Wemos D1 R2.



Gambar 2. 2 Wemos D1 R2



Gambar 2.3 Datasheet Wemos D1 R2

2.4 Liquid Crystal Display (LCD)

Liquid Crystal Display atau dengan singkatan *LCD* adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. Terdapat banyak jenis *LCD* yang beredar di pasaran. Namun ada standarisasi yang cukup populer digunakan merupakan *LCD* dengan tampilan 16x2 (16 kolom x 2 baris) dengan konsumsi daya yang rendah. *LCD* dengan jenis seperti ini memungkinkan pemrogram untuk mengoperasikan komunikasi data secara 8 bit atau 4 bit. Untuk lebih jelasnya dapat dilihat pada Gambar 2.4.



Gambar 2.4 LCD

2.5 Relay

Pada Gambar 2.5 merupakan *relay* yang berfungsi sebagai saklar mekanik. Fungsi *relay* yaitu memisahkan rangkaian listrik tegangan tinggi dengan rangkain listrik tegangan rendah. *Relay* pada Gambar 2.5 mempunyai lima buah kaki. Dua kaki digunakan untuk mengaktifkan koil. Kedua kaki ini tidak bertanda, artinya boleh terbalik dalam pemasangannya. Tiga kaki lainnya berfungsi sebagai saklar yang terdiri dari kaki *Common* (COMM), kaki *Normally Open* (NO), dan kaki *Normally Closed* (NC). Dalam keadaan koil tidak dialiri arus listrik, kaki COMM akan terhubung ke kaki NC. Dalam keadaan koil dialiri arus listrik, kaki COMM akan terhubung dengan kaki NO (Langi, Wuwung, & Lumenta, 2014).



Gambar 2. 5 *Relay*

2.6 Internet

Menurut (Nurahman, 2012) *Internet (interconnected computer networks)* bisa didefinisikan network komputer tiada batas yang menjadi penghubung pengguna komputer dengan pengguna komputer lainnya serta dapat berhubungan dengan komputer di sebuah wilayah ke wilayah di penjuru dunia, di mana di dalam jaringan tersebut mempunyai berbagai macam informasi serta fasilitas layanan internet *browsing* atau *surfing*. Internet juga merupakan sistem global jaringan komputer yang berhubungan menggunakan standar *Internet Protocol Suite (TCP/IP)* untuk melayani miliaran pengguna di seluruh dunia. Ini adalah jaringan dari jaringan yang terdiri dari jutaan jaringan pribadi, umum, akademik, bisnis, dan jaringan pemerintah dari local ke lingkungan global, yang dihubungkan oleh sebuah kode *array* yang luas dari teknologi jaringan elektronik, nirkabel, dan *optic*. Internet juga dapat didefinisikan sebagai interkoneksi seluruh dunia komputer dan jaringan komputer yang memfasilitasi sharing atau pertukaran informasi di antara pengguna. Ada beberapa jenis internet, antara lain:

1. *Local Area Network (LAN)*

Merupakan jaringan milik pribadi di dalam sebuah Gedung atau kampus yang berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer – komputer pribadi dan *workstation* dalam kantor suatu perusahaan atau pabrik – pabrik untuk memakai bersama sumberdaya (misalnya printer) dan saling bertukar informasi.

2. *Metropolitan Area Network (MAN)*

Pada dasarnya merupakan versi LAN yang berukuran lebih besar dan biasanya menggunakan teknologi yang sama dengan LAN. MAN dapat mencakup kantor –

kantor perusahaan yang letaknya berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi (swasta) atau umum. MAN mampu menunjang data dan suara, bahkan dapat berhubungan dengan jaringan televisi kabel.

3. *Wide Area Network (WAN)*

Jangkauannya mencakup daerah geografis yang luas, seringkali mencakup sebuah negara bahkan benua. WAN terdiri dari kumpulan mesin – mesin yang bertujuan untuk menjalankan program – program (aplikasi) pemakai.

4. *Internet*

Sebenarnya terdapat banyak jaringan didunia ini, seringkali menggunakan perangkat keras dan perangkat lunak yang berbeda- beda. Orang yang terhubung ke jaringan sering berharap untuk bisa berkomunikasi dengan orang lain yang terhubung ke jaringan lainnya. Keinginan seperti ini memerlukan hubungan antar jaringan yang seringkali tidak kompatibel dan berbeda. Biasanya untuk melakukan hal ini diperlukan sebuah mesin yang disebut *gateway* guna melakukan hubungan dan melaksanakan terjemahan yang diperlukan, baik perangkat keras maupun perangkat lunaknya. Kumpulan jaringan yang terinterkoneksi inilah yang disebut dengan internet.

6. *Jaringan Tanpa Kabel*

Jaringan tanpa kabel merupakan suatu solusi terhadap komunikasi yang tidak bisa dilakukan dengan jaringan yang menggunakan kabel. Misalnya orang yang ingin mendapat informasi atau melakukan komunikasi walaupun sedang berada diatas mobil atau pesawat terbang, maka mutlak jaringan tanpa kabel diperlukan karena koneksi kabel tidaklah mungkin dibuat di dalam mobil atau pesawat. Saat ini jaringan tanpa kabel sudah marak digunakan dengan memanfaatkan jasa satelit dan mampu

memberikan kecepatan akses yang lebih cepat dibandingkan dengan jaringan yang menggunakan kabel.

2.7 Android

Menurut **Invalid source specified**. Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*.

Google menginginkan agar Android bersifat terbuka dan gratis, oleh karena itu hampir setiap kode program Android diluncurkan berdasarkan lisensi *open-source* Apache yang berarti bahwa semua orang yang ingin menggunakan Android dapat mendownload penuh *source code*-nya.

Disamping itu produsen perangkat keras juga menambahkan *extension*-nya sendiri ke dalam Android sesuai kebutuhan produk mereka, model pengembangannya yang sederhana membuat Android menarik bagi vendor-vendor perangkat keras (contoh: Samsung).

Keuntungan utama dari Android adalah adanya pendekatan aplikasi secara terpadu. Pengembang hanya berkonsentrasi pada aplikasi saja, aplikasi tersebut bisa berjalan pada beberapa perangkat yang berbeda selama masih ditenagai oleh Android (pengembang tidak perlu mempertimbangkan kebutuhan jenis perangkatnya).

2.8 Aplikasi *Blynk*

Blynk adalah sebuah layanan *server* yang digunakan untuk mendukung *project Internet of Things*. Layanan *server* ini memiliki lingkungan *mobile user* baik Android maupun iOS. *Blynk* Aplikasi sebagai pendukung IoT dapat diunduh melalui *Google Play Store*. Aplikasi *Blynk* mendukung berbagai macam *hardware* yang dapat digunakan untuk *project Internet of Things*. Aplikasi *Blynk* adalah *dashborad* digital dengan fasilitas antarmuka grafis dalam pembuatan *project*. Penambahan komponen pada aplikasi *Blynk* dengan cara *Drag and Drop* sehingga memudahkan dalam penambahan komponen *Input/Output* tanpa perlu kemampuan pemrograman Android maupun iOS.

Pada Gambar 2.5 Aplikasi *Blynk* diciptakan dengan tujuan untuk kontrol dan monitoring *hardware* secara jarak jauh menggunakan komunikasi data internet ataupun intranet (jaringan LAN). Kemampuan untuk menyimpan data dan menampilkan data secara visual baik menggunakan angka, warna ataupun grafis semakin memudahkan dalam pembuatan *project* dibidang *Internet of Things*.



Gambar 2.6 Aplikasi *Blynk*

2.9 Real Time Clock

Real Time Clock (RTC) adalah rangkaian elektronika terpadu yang berfungsi sebagai pengatur waktu. RTC umumnya memiliki catu daya terpisah dari catu daya komputer (umumnya berupa baterai litium) sehingga dapat berfungsi ketika catu daya komputer terputus, dan waktu yang disimpan terus berjalan. DS1307 merupakan IC serial RTC dimana alamat dan data ditransmisikan secara serial melalui sebuah jalur data dua arah I2C.

Karena menggunakan jalur data I2C maka hanya memerlukan dua buah *pin* saja untuk komunikasi yaitu *pin* untuk data dan untuk sinyal *clock*. Pada dasarnya, sistem I2C terbagi atas dua bagian, yaitu suatu *device* yang bertindak sebagai pengontrol atau *master* dan suatu *device* yang dikontrol atau *slave*. *Master* dan *Slave* saling berkomunikasi melalui jalur data *bus* I2C. Alat yang mengendalikan komunikasi data disebut *Master* dan alat yang dikendalikan oleh *Master* dikenal sebagai *slave*.

Pada satu jalur data I2C yang sama dapat terdapat *slave* lebih dari satu, oleh karena itu I2C *Bus* harus dikendalikan *Master* yang dapat membangkitkan *serial clock* (SCL), mengontrol sistem komunikasi data (SDA), dan dapat menghasilkan kondisi-kondisi “START” dan “STOP”. Pada hal ini DS1307 beroperasi sebagai *slave* pada I2C *bus*. Untuk lebih jelas dapat dilihat pada Gambar 2.6



Gambar 2.7 *Real Time Clock*

2.10 Pompa Air AC

Pada dasarnya setiap pompa air dilengkapi dengan peralatan otomatis ketika kita membeli mesin pompa air di toko, ini berguna untuk memudahkan kita pada saat pengoperasian, sehingga waktu kita menjadi lebih efektif dan efisien dan tidak memerlukan aktifitas menghidupkan ataupun mematikan pompa, sebab sudah ada sensor otomatisnya, yang bekerja berdasarkan tekanan yang terdapat pada pipa atau saluran air pada keluaran pompa.

Pada mesin pompa air ada saluran hisap dan ada saluran buang, alat otomatis atau sensornya menggunakan sensor tekanan atau disebut juga Pressure Switch dan dipasang pada tabung pada saluran keluaran pompa, ketika pompa dihidupkan atau dihubungkan dengan tegangan jala-jala, maka pompa akan berputar sehingga dibagian dalam pompa terjadi *vaccum* karena adanya perbedaan tekanan, sehingga air yang ada didalam tanah akan terhisap naik.

Pada saat mesin pompa air berputar dan semua kran air yang ada dirumah tertutup maka pada saluran keluaran pompa akan timbul tekanan yang cukup besar, ketika tekanan yang dihasilkan melebihi tekan set yang ada pada sensor atau pressure switch maka sensor akan bekerja dan pompa air akan mati seketika, pompa air akan hidup lagi jika ada salah satu kran air terbuka disebabkan tekanan air sudah turun dan begitulah seterusnya.

Dengan demikian saat kita lupa untuk mematikan pompa air, maka mesin pompa air tidak akan terbakar disebabkan kerja yang terus menerus, dan lagi kita tidak perlu memasang atau mencabut steker dari mesin pompa air sebab segalanya akan bekerja secara otomatis. Untuk lebih jelas dapat dilihat pada Gambar 2.7.



Gambar 2.8 Pompa Air AC

2.11 **Arduino** Software IDE (*Integrated Development Environment*)

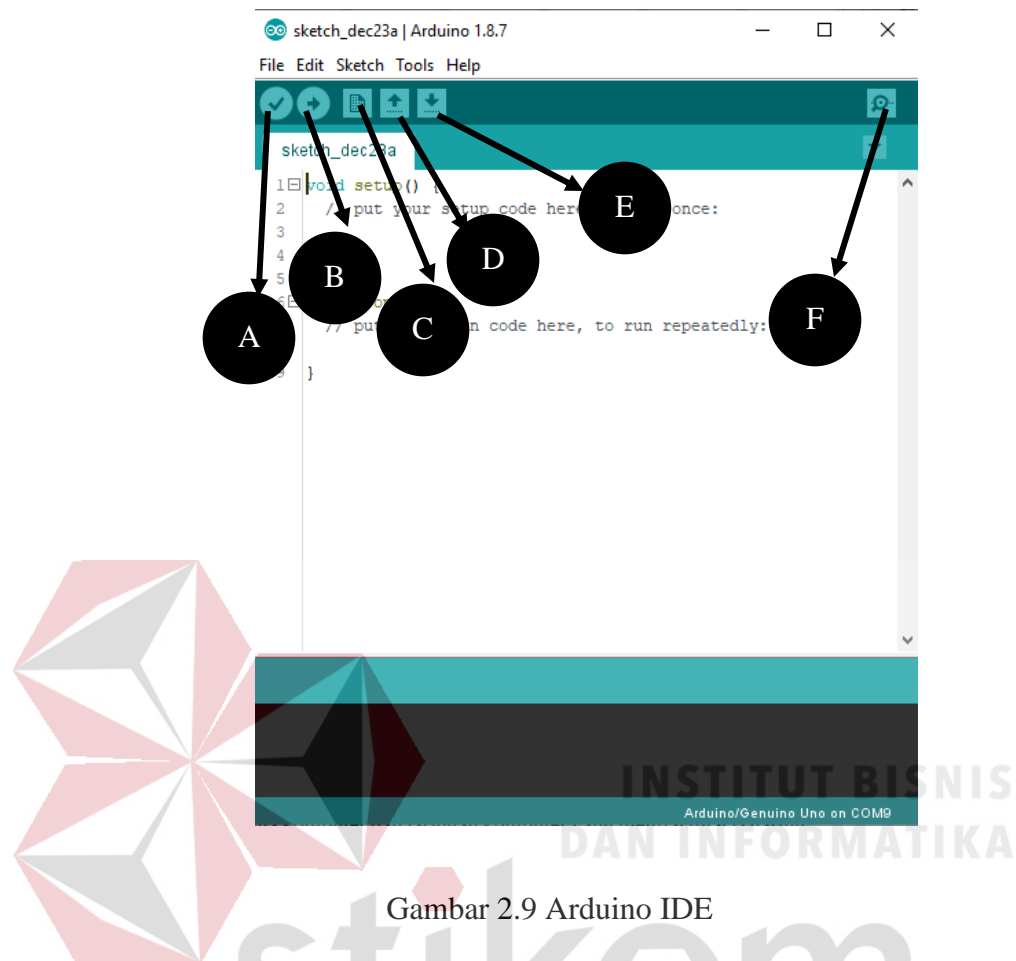
Menurut (Ecadio, 2015) adalah *Platform* dari *Arduino* disusun pada sebuah *software* yang diberi nama *Arduino IDE*. *Software* inilah yang paling utama membantu menjembatani antara bahasa mesin yang begitu rumit sehingga menjadi bahasa dan *logic* yang lebih mudah dimengerti manusia. Merupakan perangkat lunak yang telah disiapkan oleh *Arduino* bagi para perancang untuk melakukan berbagai proses yang berkaitan dengan pemrograman *Arduino*. Perangkat lunak disediakan secara gratis dan bisa didapatkan secara langsung pada halaman resmi *Arduino* yang bersifat *open-source*. *Arduino IDE* ini juga sudah mendukung berbagai sistem operasi populer saat ini seperti Windows, Mac, dan Linux. *Arduino IDE* terdiri dari:

1. Editor program, sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.

2. *Verivy / Compiler*, sebuah modul yang mengubah kode program (bahasa *processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*, yang dipahami oleh mikrokontroler adalah kode biner.
3. Pengunggah, sebuah modul yang memuat kode biner dari komputer ke dalam memori mikrokontroler di dalam papan *Arduino*.

Pada Gambar 2.8 terdapat *menu bar*, kemudian pada bawahnya terdapat bagian *toolbar*, dan sebuah area putih untuk *editing sketch*, area hitam dapat disebut sebagai *process area*, dan paling bawah dapat disebut sebagai *status bar*.





Gambar 2.9 Arduino IDE

Keterangan:

- A. Compile "Verify"
- B. Upload
- C. New Program
- D. Open Program
- E. Save Program
- F. Serial Monitor

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Dalam perancangan sistem ini, awal mula program *blynk* yang sudah ada pada *play store* di *download* dan di *instal* pada *smartphone Android*. Pada aplikasi *blynk* terdapat beberapa fitur yang dapat digunakan untuk membuat aplikasi control penyiram taman otomatis dan monitoring.

Langkah pertama untuk menggunakan aplikasi *blynk* ini, user harus *log in* dengan memasukkan *username* dan *password* yang sudah dibuat. Setelah *log in* user akan diarah untuk *setting* tampilan awal. Beri nama *project*, *setting device* yang akan digunakan lalu *creat*. Setelah *creat blynk* akan mengirimkan *auth token* ke email yang sudah di daftarkan sebelumnya. *Auth token* sendiri berfungsi sebagai sinkronisasi antara *Wemos D1 R2* dan aplikasi *blynk*.

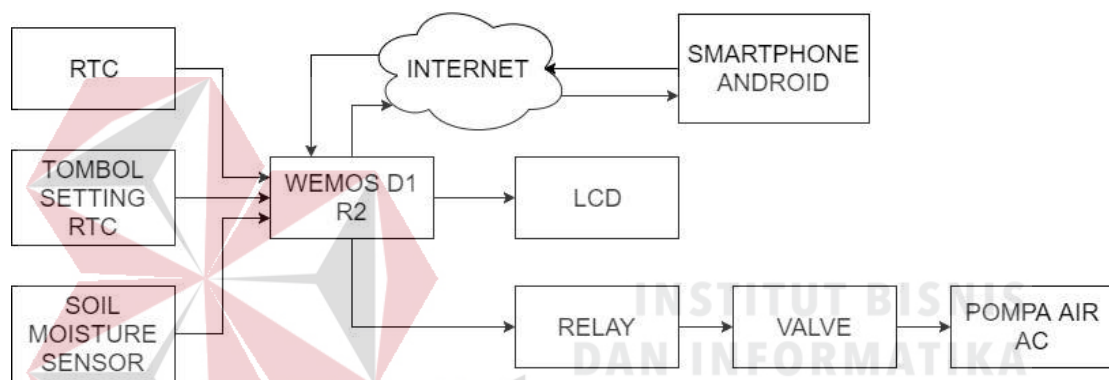
Setelah *creat*, dapat menggunakan *widget box* sesuai dengan kebutuhan. Untuk memonitoring kelembaban tanah menggunakan *gauge* lalu *setting input* yang digunakan. Untuk suhu menggunakan *value display*, lalu *setting input* yang digunakan dan untuk notifikasi menggunakan *notification* lalu *setting notification* yang digunakan.

Pada sistem penyiram taman terdapat *module Wemos D1 R2* yang akan terhubung dengan koneksi internet. *Wemos D1 R2* berfungsi sebagai pengirim data ke *web server*

melalui *internet*. *Wemos D1 R2* dapat terhubung dengan internet untuk mendapatkan *IP Address* agar dapat mengakses *web server*.

3.2 Model Perancangan

Pada perancangan pengerjaan Tugas Akhir ini penggambaran perancangan sistem seperti pada Gambar 3.1 berikut:



Gambar 3.1 Blok Diagram

Tiap – tiap bagian dari blok diagram pada Gambar 3.1 dapat dijelaskan sebagai berikut:

1. *Input* pada *Wemos D1 R2*
 - a. *RTC* : Sensor ini berfungsi untuk mengatur waktu proses penyiraman.
 - b. *Tombol Setting RTC* : Tombol ini berfungsi untuk mengatur waktu pada RTC apabila RTC terjadi *error*.
 - c. *Soil Moisture Sensor* : Sensor ini berfungsi sebagai pengukur kelembaban tanah.

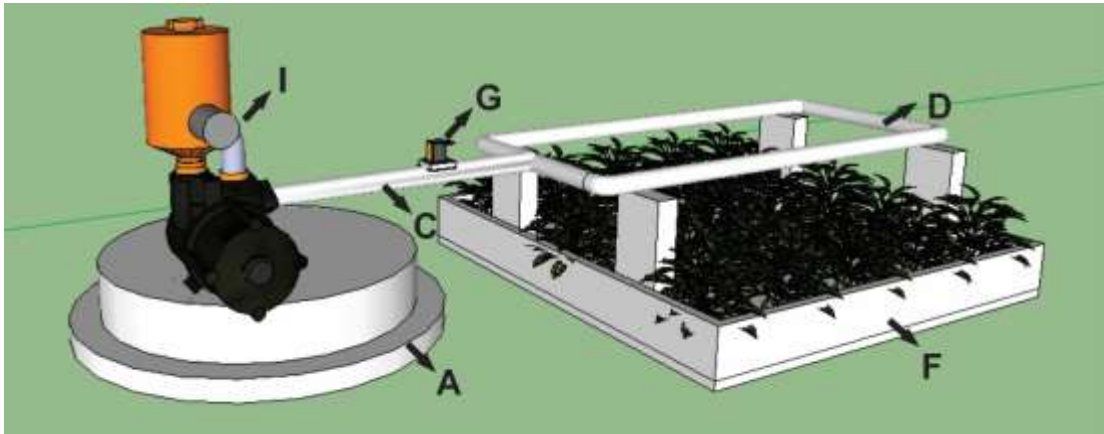
- d. *Smartphone Android* : Pada *Smartphone Android* yang sudah terpasang aplikasi *Blynk* dapat sebagai *input* yaitu mengatur buka tutup *valve*.

2. *Output* pada *Wemos D1 R2*

- a. *Relay* : Sebagai saklar elektrik yang berguna untuk mengaktifkan atau memutus aliran listrik.
- b. *Solenoid Valve* : Sebagai kran otomatis berfungsi untuk mengalirkan air dari tempat penampung air ke taman apabila terjadi proses penyiraman.
- c. Pompa Air AC : Berfungsi menarik air dari penampung air menuju *valve*.
- d. LCD : Sebagai penampil data kelembaban tanah dan waktu.
- f. *Smartphone Android* : Pada *Smartphone Android* yang sudah terpasang aplikasi *Blynk* dapat sebagai *output* yaitu menampilkan data kelembaban tanah dan waktu.

3.3 Perancangan Mekanik

- 1. Berikut ini desain mekanik dari rancang bangun kontrol penyiram taman otomatis dan monitoring kelembaban tanah menggunakan Android pada Gambar 3.2.



Gambar 3.2 Desain Rancang Mekanik Tampak Keseluruhan

2. Berikut ini hasil mekanik dari rancang bangun kontrol penyiram taman otomatis dan monitoring kelembaban tanah menggunakan Android pada Gambar 3.3 dan Gambar 3.4.



Gambar 3.3 Hasil Rancang Mekanik Bagian Luar



Gambar 3.4 Hasil Rancang Mekanik Bagian Dalam

Keterangan:

- A. Penampung air, berfungsi apabila taman yang digunakan jauh dari sumber air.
- B. Wadah plastik, digunakan untuk menyimpan alat elektronik dari penyiram taman. Wadah plastik yang digunakan harus kedap dari air agar tidak terjadi kerusakan alat karena terkena air.
- C. Pipa paralon, pada bagian c pipa paralon berfungsi menyalurkan air dari penampung ke pompa air AC

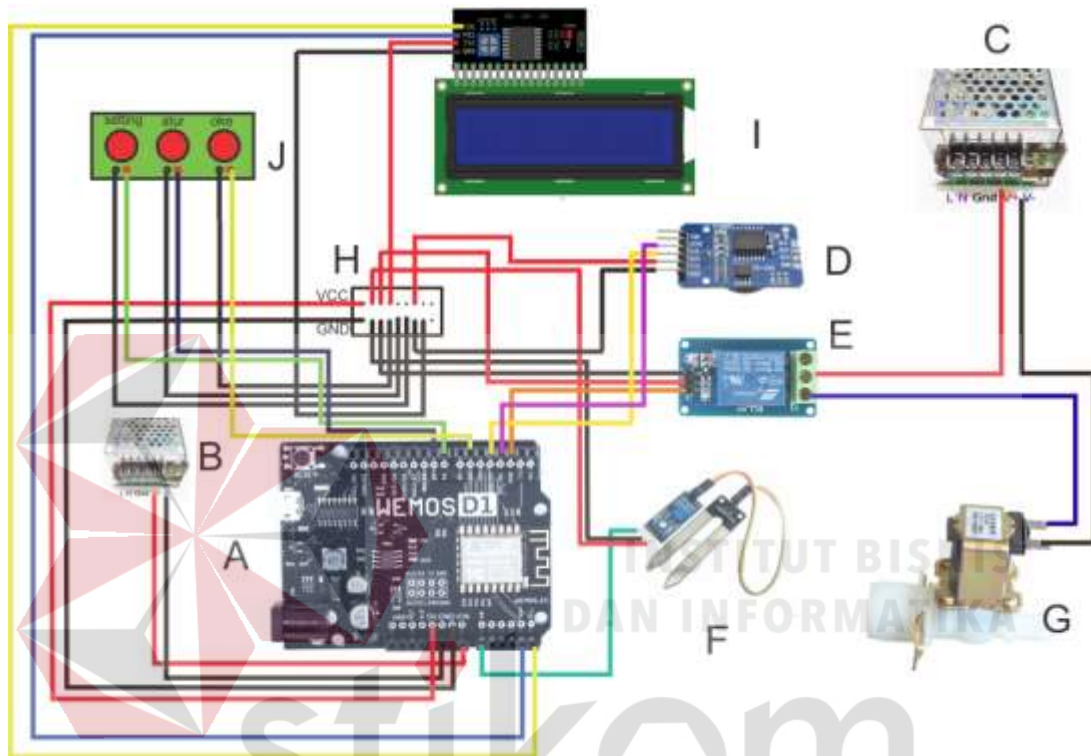
- D. Pipa paralon, pada bagian d pipa paralon berfungsi menyalurkan air dari pompa menuju ke penyiram taman (Sprinkkler)
- E. *Sprinkler* taman, berfungsi untuk penyiraman taman. Material plastik dengan inti kuningan yang bias diatur dengan memutar kepala kuningan. Radius dari alat ini sejauh 1 – 3meter dengan drat ½ inch.
- F. Kayu, digunakan sebagai penompang dari paralon dan wadah dari media tanah pda tumbuhan.
- G. *Solenoid valve*, berfungsi sebagai kran air yang berkerja secara otomatis. Bekerja dengan tegangan 12V DC secara *normaly close* dengan temperature air 1 - 100°C.
- H. Kran air, digunakan untuk menahan laju air yang besar dari pompa air.
- I. Pompa air, berfungsi sebagai penyedot air dari penampungan. Pompa air yang digunakan sudah berjalan dengan otomatis.

Setelah dilakukan perakitan seluruh mekanik dan pemasangan komponen elektronika pada Gambar 3.3 dan Gambar 3.4, dilakukan pengukuran dimensi pada rancang bangun dan dihasilkan dimensi sebagai berikut :

- 1. Panjang Rancang Bangun : 93 cm
- 2. Lebar Rancang Bangun : 82 cm
- 3. Tinggi Rancang Bangun : 16 cm

3.4 Perancangan Elektronik

Pada Gambar 3.5 merupakan gambar keseluruhan dari rangkain elektronik dari kontrol penyiram taman otomatis dan monitoring menggunakan Android.



Gambar 3.5 Perancangan Elektronik

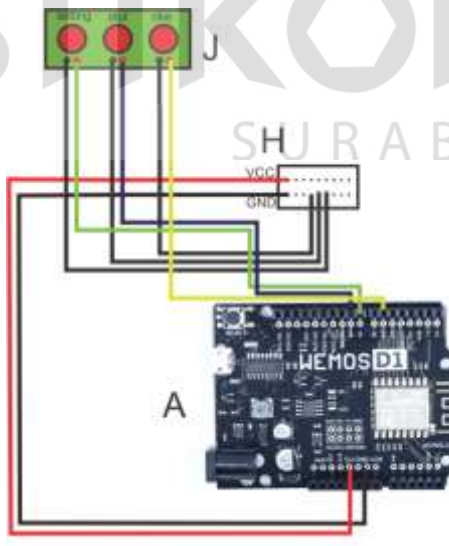
Keterangan pada Gambar 3.5, sebagai berikut:

- A. Wemos D1 R2
- B. Adaptor 5 V
- C. Adaptor 12 V
- D. RTC
- E. Relay
- F. Soil Moisture Sensor

- G. Solenoid Valve
- H. Rangkaian VCC GND
- I. LCD I2C
- J. Tombol Setting

3.4.1 Rangkaian Tombol *Setting* RTC

Pada Gambar 3.6 dijelaskan alur rangkaian tombol untuk *setting real time clock*. Terdapat 3 tombol yang digunakan untuk mengatur RTC. Tombol *setting* digunakan untuk menghentikan sistem utama dan masuk pada menu *setting*. Tombol *setting* digunakan juga untuk memilih menu jam, menit dan detik yang akan diatur. Tombol *atur* digunakan untuk mengatur atau menyesuaikan jam, menit, detik yang ada. Tombol *oke* digunakan untuk memasukkan data RTC yang sudah diatur dan mengembalikan ke sistem utama.



Gambar 3.6 Rangkaian Tombol *Setting* RTC

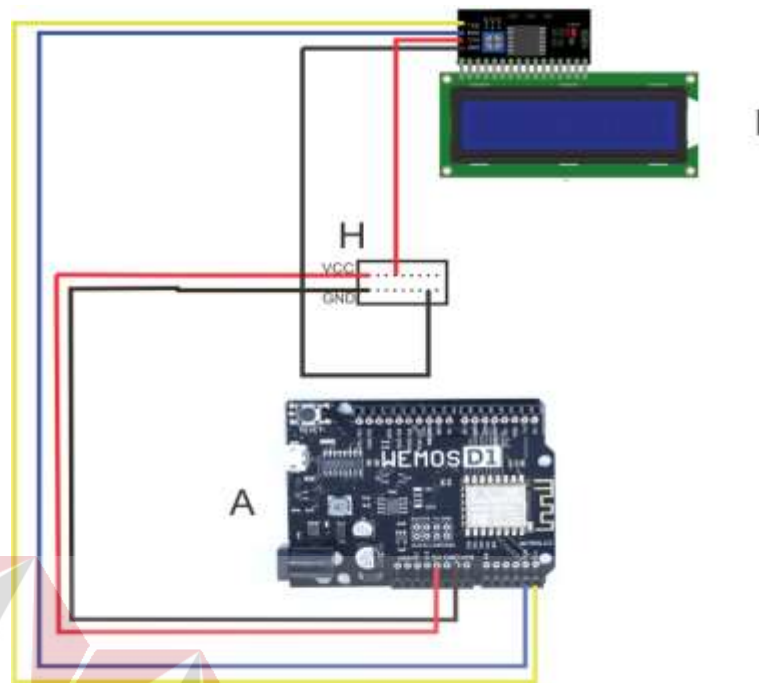
Keterangan:

Pada Tombol *Setting* RTC terdapat pin – pin yang terhubung ke Wemos D1 R2

- a. Pin Data dari Tombol *Setting* terhubung dengan pin D6 pada Wemos D1 R2.
- b. Pin GND dari Tombol *Setting* terhubung dengan GND pada rangkaian VCC/GND.
- c. Pin Data dari Tombol Atur terhubung dengan pin D7 pada Wemos D1 R2.
- d. Pin GND dari Tombol Atur terhubung dengan GND pada rangkaian VCC/GND.
- e. Pin Data dari Tombol Oke terhubung dengan pin D4 pada Wemos D1 R2.
- f. Pin GND dari Tombol Oke terhubung dengan GND pada rangkaian VCC/GND.
- g. Pin VCC dari rangkaian VCC/GND terhubung dengan pin 5V dari Wemos D1 R2.
- h. Pin GND dari rangkaian VCC/GND terhubung dengan pin GND dari Wemos D1 R2.

3.4.2 Rangkaian LCD

Pada Gambar 3.7 merupakan rangkaian LCD yang digunakan untuk menampilkan data dari kelembaban tanah dan waktu. LCD ini juga untuk menampilkan menu untuk mengatur RTC apabila terjadi *error*. LCD yang digunakan pada tugas akhir ini sudah memiliki fitur I2C.



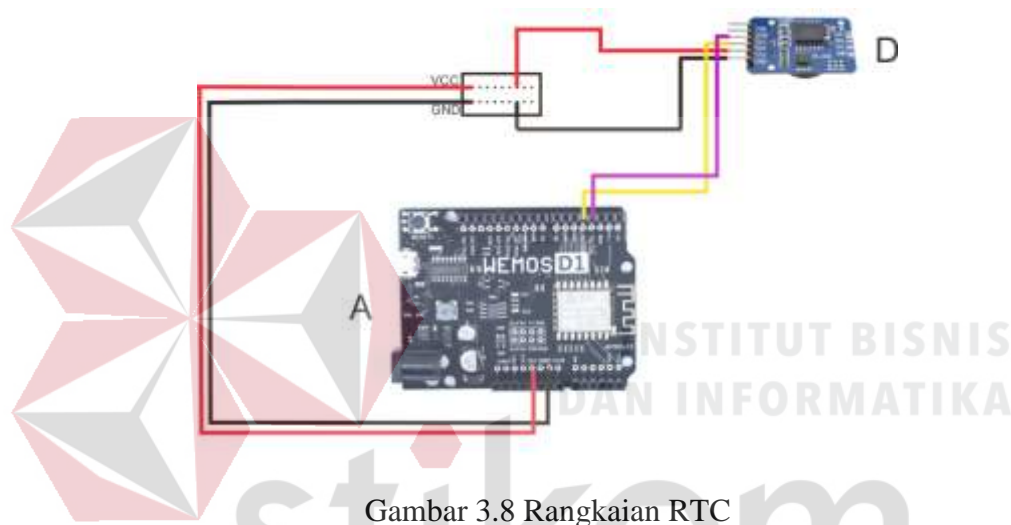
Gambar 3.7 Rangkaian LCD I2C

Keterangan:

- a. Pin SDA dari LCD I2C terhubung dengan pin SDA pada Wemos D1 R2.
- b. Pin SCL dari LCD I2C terhubung dengan pin SCL pada Wemos D1 R2.
- d. Pin GND dari LCD I2C terhubung dengan GND pada rangkaian VCC/GND.
- e. Pin VCC dari LCD I2C terhubung dengan VCC pada rangkaian VCC/GND.
- f. Pin VCC dari rangkaian VCC/GND terhubung dengan pin 5V dari Wemos D1 R2.
- g. Pin GND dari rangkaian VCC/GND terhubung dengan pin GND dari Wemos D1 R2.

3.4.3 Rangkaian RTC

Pada Gambar 3.8 merupakan rangkaian RTC yang digunakan untuk menghitung waktu dengan akurat dan menjaga / menyimpan data waktu tersebut secara *real time*. Karena jam tersebut bekerja *real time*, maka setelah proses hitung waktu dilakukan *output* datanya langsung disimpan atau dikirim ke *device* lain melalui sistem antarmuka.



Gambar 3.8 Rangkaian RTC

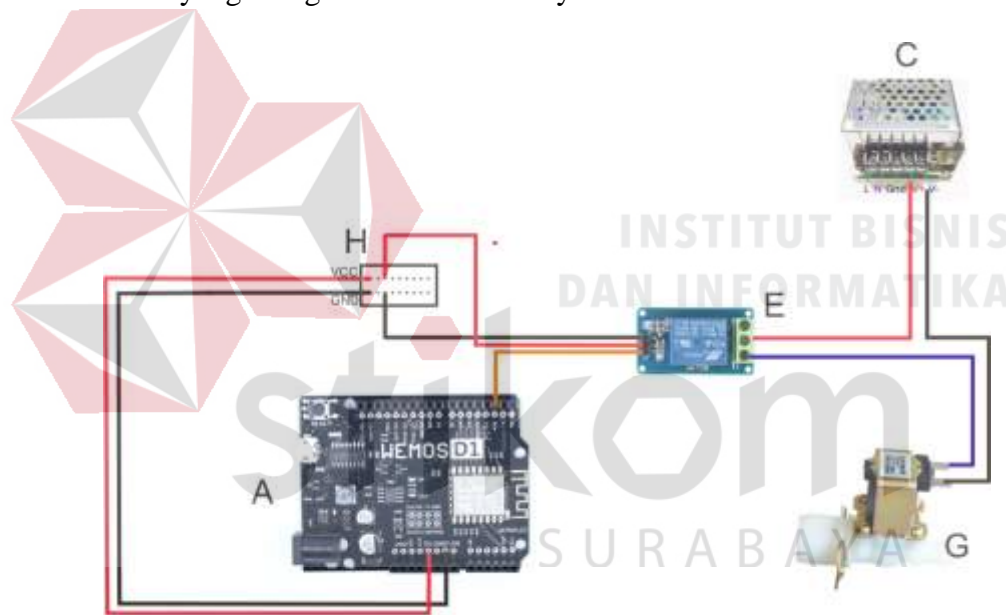
Keterangan:

- a. Pin SDA dari RTC terhubung dengan pin SDA/D2 pada Wemos D1 R2.
- b. Pin SCL dari RTC terhubung dengan pin SCL/D1 pada Wemos D1 R2.
- d. Pin GND dari RTC terhubung dengan GND pada rangkaian VCC/GND.
- e. Pin VCC dari RTC terhubung dengan VCC pada rangkaian VCC/GND.
- f. Pin VCC dari rangkaian VCC/GND terhubung dengan pin 5V dari Wemos D1 R2.

- g. Pind GND dari rangkaian VCC/GND terhubung dengan pin GND dari Wemos D1 R2.

3.4.4 Rangkaian *Relay* dan *Solenoid Valve*

Pada Gambar 3.9 merupakan rangkaian *relay* dan *Solenoid valve* fungsi *relay* sendiri berfungsi sebagai pemisah antara rangkaian listrik tegangan tinggi dengan rangkaian listrik tegangan rendah. Sedangkan pada *solenoid valve* berfungsi sebagai kran otomatis yang mengatur air untuk menyiram taman.



Gambar 3.9 Rangkaian *Relay* dan *Solenoid Valve*

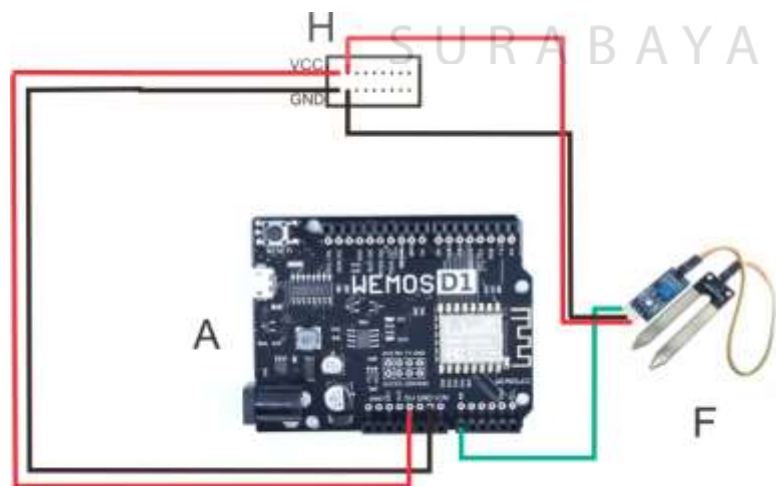
Keterangan:

- Pin IN dari *Relay* terhubung dengan pin D5 pada Wemos D1 R2.
- Pin COM dari *Relay* terhubung dengan pin *min* (-) pada Adaptor 12V.
- Pin NO dari *Relay* terhubung dengan pin *plus* (+) pada Adaptor 12V.

- d. Pin *plus* (+) dari *Solenoid Valve* terhubung dengan pin NO pada *Relay*.
- e. Pin *min* (-) dari *Solenoid Valve* terhubung dengan pin *plus* (+) pada *Adaptor 12V*.
- f. Pin VCC dari *Relay* terhubung dengan VCC pada rangkaian VCC/GND.
- g. Pin GND dari *Relay* terhubung dengan GND pada rangkaian VCC/GND.
- h. Pin VCC dari rangkaian VCC/GND terhubung dengan pin 5V dari Wemos D1 R2.
- i. Pin GND dari rangkaian VCC/GND terhubung dengan pin GND dari Wemos D1 R2.

3.4.5 Rangkain Soil Moisture Sensor

Pada Gambar 3.10 merupakan rangkaian dari sensor kelembaban tanah. sensor kelembaban tanah berfungsi untuk mengukur kadar kelembaban yang ada pada tanah dengan cara melewatkan arus listrik dalam tanah, kemudia membaca resistansinya untuk mendapatkan nilai tingkat kelembaban.



Gambar 3.10 Rangkaian *Soil Moisture Sensor*

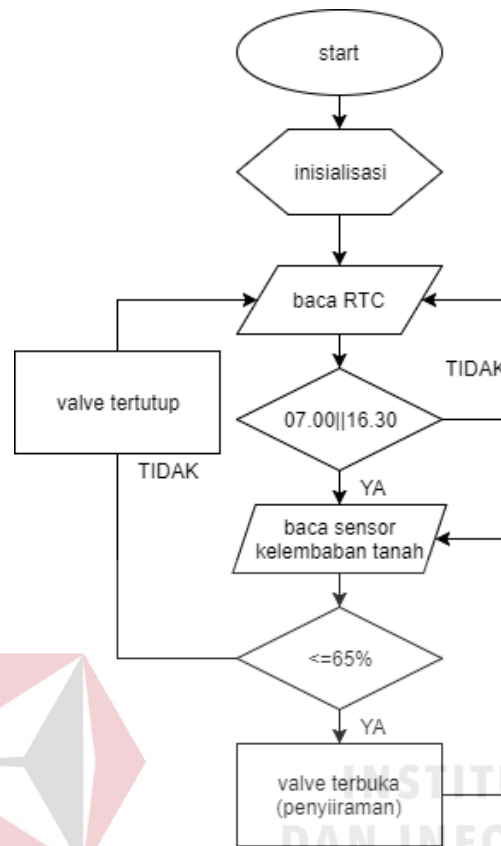
Keterangan:

- a. Pin A0 dari *Soil Moisture Sensor* terhubung dengan pin A0 pada Wemos D1 R2.
- b. Pin GND dari *Soil Moisture Sensor* terhubung dengan GND pada rangkaian VCC/GND.
- c. Pin VCC dari *Soil Moisture Sensor* terhubung dengan VCC pada rangkaian VCC/GND.
- d. Pin VCC dari rangkaian VCC/GND terhubung dengan pin 5V dari Wemos D1 R2.
- e. Pin GND dari rangkaian VCC/GND terhubung dengan pin GND dari Wemos D1 R2.

3.5 Perancangan Software

3.5.1 Flowchart Penyiram Taman

Pada Gambar 3.11 adalah alur cara kerja dari penyiraman taman. Awal mula proses melakukan inisialisasi. Setelah itu membaca nilai dari *Real Time Clock*. Jika sesuai dengan jam 7 pagi dan jam 4 sore, maka akan membaca nilai kelembaban tanah. Jika tidak sesuai dengan waktu, maka akan terus membaca nilai dari *Real Time Clock* sampai sesuai. Jika nilai dari *Real Time Clock* sudah sesuai maka akan membaca nilai dari sensor kelembaban tanah. Jika nilai kelembaban tanah kurang dari 60% maka proses penyiraman akan terjadi atau, *valve* akan terbuka sampai dengan nilai kelembaban tanah lebih dari 60 %. Apabila nilai kelembaban tanah lebih dari 60 % maka *valve* akan tertutup dan membaca nilai dari *Real Time Clock* lagi.



Gambar 3.11 Flowchart Penyiram Taman

3.5.2 Aplikasi Monitoring dan Kontrol

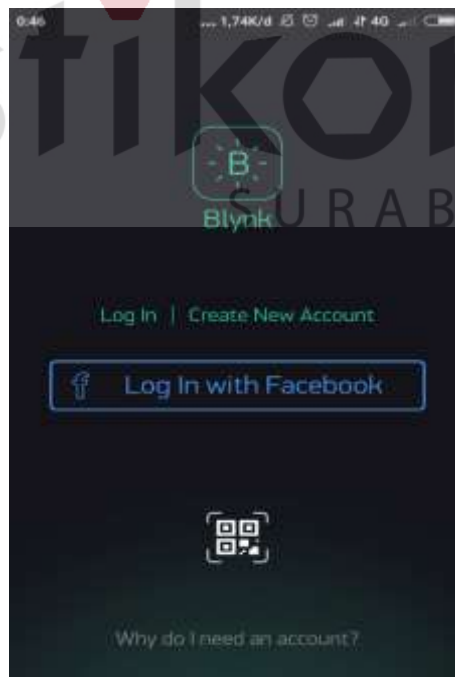
Pengerjaan tugas akhir ini diawali dengan penggunaan aplikasi *Android* yang akan digunakan *Monitoring* penyiram taman otomatis. Aplikasi *Blynk* dapat diunduh di *Play Store*. *Blynk* memungkinkan untuk membuat *project interface* dengan berbagai macam komponen *input output* yang mendukung untuk pengiriman maupun penerimaan data serta merepresentasikan data sesuai dengan komponen yang dipilih. Representasi data dapat berbentuk visual angka maupun grafik. Terdapat 4 jenis kategori komponen yang berdatang pada Aplikasi *Blynk* antara lain:

1. *Controller* digunakan untuk mengirimkan data atau perintah ke *Hardware*.

2. *Display* digunakan untuk menampilkan data yang berasal dari *hardware* ke *smartphone*.
3. *Notification* digunakan untuk mengirim pesan dan notifikasi.
4. *Interface* pengaturan tampilan pada aplikasi *Blynk* dapat berupa menu ataupun *tab*.

3.5.2.1 Tampilan Log in

Saat pertama kali memulai aplikasi yaitu tampilan *log in*, berfungsi untuk menjadi tampilan pembuka atau halaman *log in*. Ketika *user* menginputkan *username* dan *password* yang telah di daftarkan pada *Blynk*. Dapat dilihat pada Gambar 3.12 dibawah ini.



Gambar 3.12 Tampilan *Creat New Account*

Pada Gambar 3.12 di atas terdapat menu *Creat New Account* apabila *user* masih belum mendaftar. Untuk *log in* dapat menggunakan akun yang sudah didaftarkan pada *Blynk* atau menggunakan akun *Facebook*.

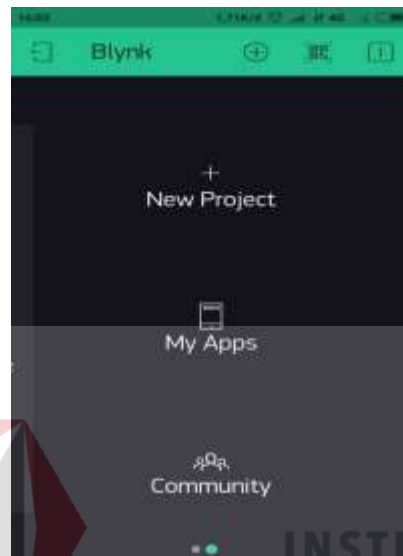


Gambar 3. 13 Tampilan *log in*

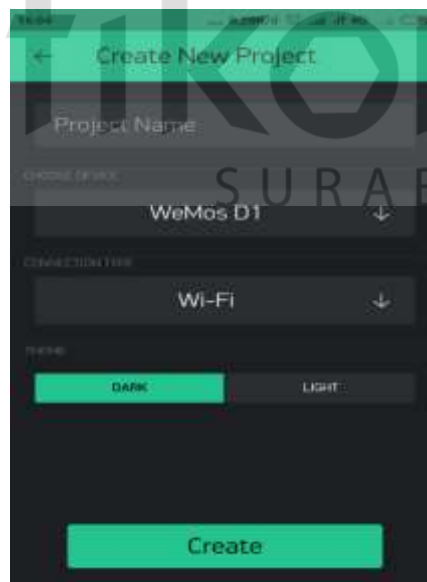
Pada Gambar 3.13 di atas merupakan tampilan *log in* pada aplikasi *Blynk*. *Email* yang digunakan merupakan akun yang sudah didaftarkan pada *Blynk*. Apabila *user* lupa dengan *password*, *user* dapat mengganti *password* dengan cara memilih *Forgot Password*.

3.5.2.2 Membuat Project Baru

Pada langkah selanjutnya Pada Gambar 3.13 pilih tanda + *New Project* untuk membuat *project* baru.



Gambar 3.14 Tampilan *New Project*

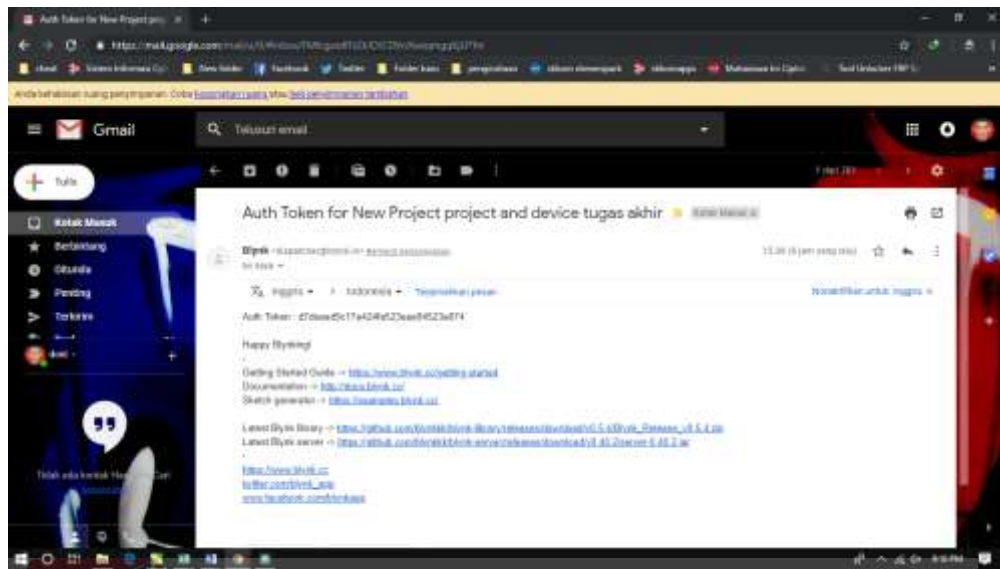


Gambar 3.15 Tampilan *Creat New Project*

Pada Gambar 3.15 terdapat 4 bagian. Yang pertama yaitu *project name* yang berfungsi untuk memberi nama pada *project* tersebut. Yang kedua *Choose Device*, digunakan untuk memilih *device* yang akan digunakan. Yang ketiga yaitu *Connection Type*, digunakan untuk memilih koneksi yang akan digunakan. Dan yang terakhir adalah *Theme* digunakan untuk memilih tema yang akan digunakan pada *project*. Untuk tugas akhir ini *Project Name* yaitu Tugas Akhir, pada *Choose Device* menggunakan *Wemos D1*. Pada *Connection Type* menggunakan *Wifi*, dan pada *Theme* menggunakan *Dark*. Apabila semua sudah diatur pilih *Create* untuk membuat *Project* baru. Ketika sudah membuat *Project* baru maka aplikasi mengirimkan sebuah *Auth Token*/



Gambar 3.16 *Auth Token* yang dikirim ke *Gmail*

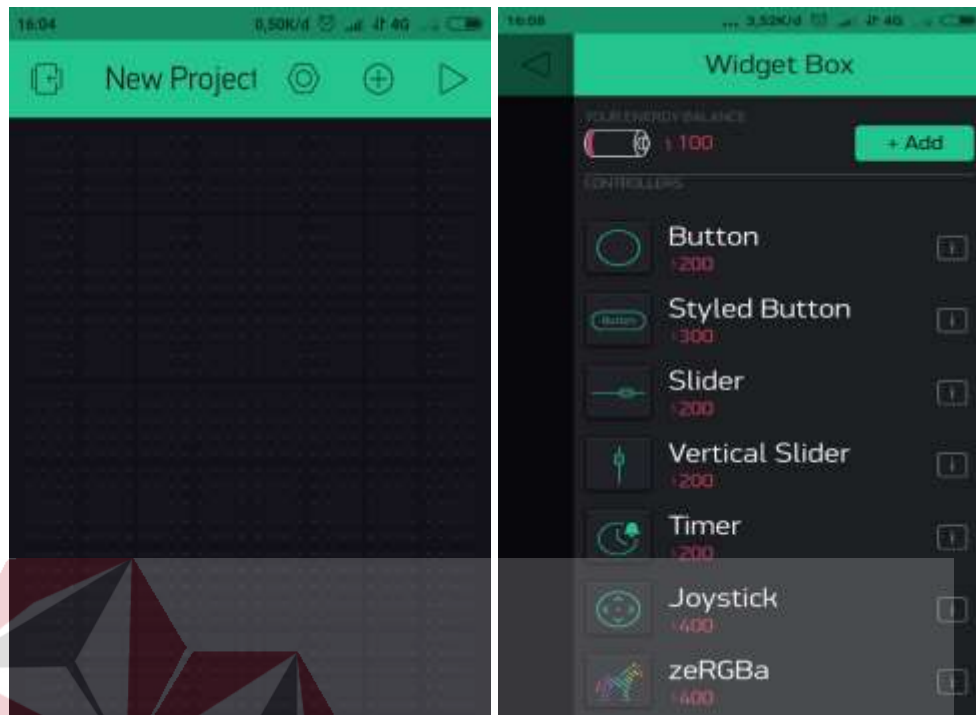


Gambar 3.17 *Auth Token* yang diterima oleh *Gmail*

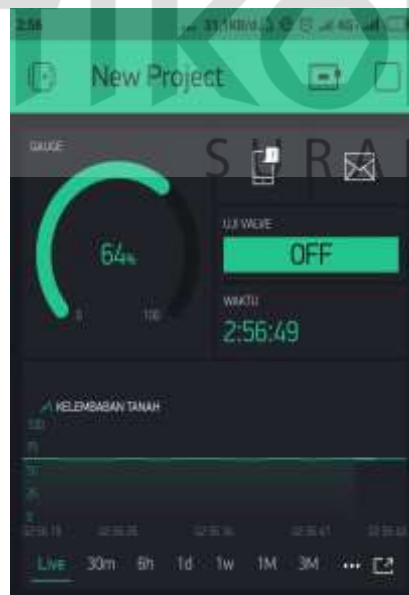
Pada Gambar 3.16 *Auth Token* mengirimkan kode ke *Email* yang sudah digunakan untuk *Log In* yang berfungsi sebagai penghubung perangkat yang digunakan. Pada Gambar 3.17 *Gmail* menerima *Auth Token* yang dikirim oleh aplikasi *Blynk*.

3.6.1.3 Pemilihan Widget Box

Pada Gambar 3.18 untuk menambah komponen *interface* pada *project* dapat menggunakan menu *plus* yang ada didalam lingkaran. Terdapat berbagi macam komponen yang diperlukan untuk mendukung tampilan. Berbagai macam komponen yang tersedia disesuaikan dengan *credit power* yang masih tersisa. *Credit power* pada saat *log in* diberikan sejumlah 2000. Apabila *credit power* habis dapat ditambah dengan membeli pada menu *Add*.



Gambar 3.18 Tampilan Awal dan *Widget Box*



Gambar 3. 19 Tampilan Aplikasi *Blynk*

Pada Gambar 3.19 merupakan gambar *Blynk* yang siap di gunakan dengan penambahan *widget* dan sudat diatu sebagaimana *port* yang akan digunakan. Untuk *widget* yang digunakan sebagai berikut:

- a. *Gauge*, digunakan untuk menampilkan presentase kelembaban tanah yang diambil dari sensor kelembaban tanah.
- b. *Value Display*, digunakan untuk menampilkan waktu yang diambil dari *Real Time Clock*
- c. *Styled Button*, digunakan untuk menampilkan tombol tulisan.
- d. *Notification*, digunakan untuk memunculkan notifikasi pada *Android*.
- e. *Email*, digunakan untuk mengirimkan notifikasi ke *Email* yang telah didaftarkan.
- f. *SuperChart*, digunakan untuk menampilkan *track record* pada sensor kelembaban tanah.

3.6.1.4 Pengaturan Widget Box

- a. *Gauge*

Pada Gambar 3.20 pada diatas nama *Gauge* menjadi KELEMBABAN TANAH, *input* diatur menjadi Virtual 5. Untuk nilai dari V5 diatur mulai dari 0 sampai dengan 100. Label digunakan untuk memberi tanda % pada nilai V5. PUSH digunakan agar tidak ada *delay* untuk pengiriman nilai pada V5.



Gambar 3.20 Setting Gauge

```
#include <EasyDDNS.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <DS3231.h>
#include <LiquidCrystal_I2C.h>

#define VALVE D5
#define BLYNK_PRINT Serial
#define SensorPin A0
#define Button1 D6
#define Button2 D7
#define Button3 D4

LiquidCrystal_I2C lcd(0x27, 16, 2);

WiFiClient client;
WiFiServer server(80);

RTCLib RTC;
DS3231 Clock;

char Auth[] = "d7deaed9c17a424fa523eaa94523a874";
char Ssid[] = "STIKOMNet";
char Pass[] = "StikomSoerabaja";

void SensorKelembapan() {
  Value = analogRead(SensorPin);
  Nilai = 100-(Value*100)/1023 ;
  Blynk.virtualWrite(V5, Nilai); //BLYNK_READ(V5);
  Serial.print("Nilai kelembaban: ");
  Serial.print(Nilai);
  Serial.println("");
  Serial.println();
}
```

Gambar 3.21 Program Kelembaban Tanah

Pada Gambar 3.21 merupakan program agar *gauge* kelembaban tanah pada aplikasi *Blynk* dapat muncul sesuai data pada *soil moisture sensor*.

b. *Value Display*

Pada Gambar 3.21 pada diatas nama *Value Display* menjadi waktu, *input* diatur menjadi Virtual 1. Untuk nilai dari V1 diatur mulai dari 0 sampai dengan 1023. *PUSH* digunakan agar tidak ada *delay* untuk pengiriman nilai pada V1.



Gambar 3.22 *Setting Value Display*

```

#include <EasyDDNS.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <DS3231.h>
#include <LiquidCrystal_I2C.h>

#define VALVE D5
#define BLYNK_PRINT Serial
#define SensorPin A0
#define Button1 D6
#define Button2 D7
#define Button3 D4

LiquidCrystal_I2C lcd(0x27, 16, 2);

WiFiClient client;
WiFiServer server(80);

RTCLib RTC;
DS3231 Clock;

char Auth[] = "d7deaed9c17a424fa523eaa94523a874";
char Ssid[] = "STIKOMNet";
char Pass[] = "Stlk0MSoerabaja";

void RealTimeClock() {
  EasyDDNS.update(10000);
  DateTime now = RTC.now();
  Jam = (now.hour());
  Menit = (now.minute());
  Detik = (now.second());
  Blynk.virtualWrite(V1, Jam, ":", Menit, ":", Detik);
  BLYNK_READ(V1);
  Serial.print(Jam); Serial.print(":");
  Serial.print(Menit); Serial.print(":");
  Serial.println(Detik);
}

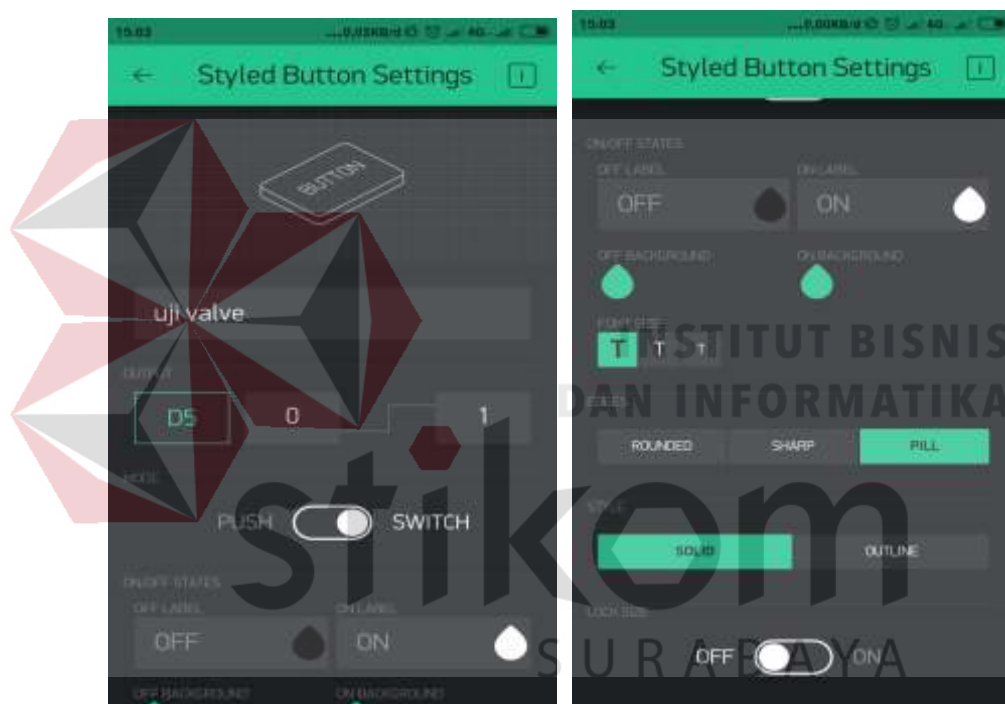
```

Gambar 3.23 Program RTC

Pada Gambar 3.23 merupakan program agar *value display* waktu pada aplikasi *Blynk* dapat muncul sesuai data pada *real time clock*.

c. *Styled Button*

Pada Gambar 3.24, *styled button* berfungsi sebagai tombol uji *valve*. Uji *valve* digunakan untuk menguji apakah *valve* berkerja dengan bai atau tidak diluar dari fungsi utama yaitu penyiram taman otomatis. Pada *styled button setting*, *name* menjadi uji *valve*. Untuk output di atur pada D5 karena IN pada relay tersambung pada Pin D5 pada *Wemos D1 R2*. Untuk nilai 0 kondisi relay *OFF* dan untuk kondisi 1 relay *ON*



Gambar 3.24 Setting *Styled Button*

Mode dipilih menjadi *Switch* sehingga pada saat ditekan *button* bernilai 1 dan pada penekanan yang kedua *button* bernilai 0.

```

1 #include <EasyGPRS.h>
2 #include <ESP8266WiFi.h>
3 #include <BlynkSimpleEsp8266.h>
4 #include <Wire.h>
5 #include <H8231.h>
6 #include <LiquidCrystal_I2C.h>
7
8 #define VALVE CO
9 #define BLYNK_PRINT Serial
10 #define SensorPin A0
11 #define Button1 D6
12 #define Button2 D7
13 #define Button3 D4
14
15 LiquidCrystal_I2C lcd(0x27, 16, 2);
16
17 WiFiClient client;
18 WiFiServer server(80);
19
20 RTClib RTC;
21 DS1307 Clock;
22
23 char Auth[] = "d7deed9c17a4262a53bea94523aE74";
24 char Ssid[] = "STIKOMNet";
25 char Pass[] = "StikomSoreboja";

```

Gambar 3. 1 Program Valve

Pada Gambar 3.24 merupakan program agar *styled button* uji *valve* pada aplikasi *Blynk* dapat digunakan sesuai fungsinya.

d. *Notification*

Pada Gambar 3. 25 *Notify When Hardware Goes Offline* diatur menjadi *On*. Berfungsi apabila Arduino tidak terhubung ke internet atau mati maka, akan mengirimkan notifikasi pada *Android*. *Offline Ignore Period* diatur menjadi *Instant* apabila Arduino tidak terhubung dengan internet atau mati maka saat itu juga mengirimkan pesan notifikasi pada *Android*.



Gambar 3. 25 Setting Notification


```

147 void Eksekusi(){
148   if((Jam == 8 || Jam == 16) && Trigger == true)
149   {
150     if(Nilai < 60)
151     {
152       Serial.println("Masuk Dibawah 50");
153       Blynk.email("doniramadhan218@gmail.com", "Blynk", "Saatnya Menyiram Teman - Teman");
154       Blynk.notify("Saatnya Menyiram Temen-Temen");
155       while(Nilai <= 50){
156       {
157         Serial.println("Masuk");
158         SensorKelembapan();
159         digitalWrite (VALVE,HIGH);
160       }
161       Blynk.notify("Kelembapan Sudah Diatas 60");
162       digitalWrite (VALVE,LOW);
163     }
164     else if(Nilai >= 60)
165     {
166       Serial.println("Masuk Diatas 60");
167       Blynk.email("doniramadhan218@gmail.com", "Blynk", "Kelembapan Sudah Diatas 60!");
168       Blynk.notify("Kelembapan Sudah Diatas 60");
169       digitalWrite (VALVE,LOW);
170     }
171     Trigger = false;
172   }
173   else if(Jam != 8 && Jam!= 16)
174   {
175     Trigger = true;
176     Serial.println("Masuk True");
177   }
178   else
179   {
180   }
181 }
182 }

```

Gambar 3.26 Program Notifikasi

Pada Gambar 3.26 merupakan program agar *nitification* pada aplikasi *Blynk* dapat digunakan sesuai fungsinya.

d. *Email*

Pada Gambar 3.27 pengaturan *Email* ditujukan kepada *Email* yang digunkann untuk *Log in*. *Blynk* akan mengirimkan notifikasi yang sudah diatur pada perangkat.



Gambar 3.27 *Setting Email*

```

147 void Eksekusi(){
148   if((Jam == 8 || Jam == 16) && Trigger == true)
149   {
150     if(Nilai < 60)
151     {
152       Serial.println("Masuk Dibawah 50");
153       Blynk.email("doniramadhan218@gmail.com", "Blynk", "Saatnya Menyiram Teman - Teman");
154       Blynk.notify("Saatnya Menyiram Temen-Temen");
155       while(Nilai <= 50){
156       {
157         Serial.println("Masuk");
158         SensorKelembapan();
159         digitalWrite (VALVE,HIGH);
160       }
161       Blynk.notify("Kelembapan Sudah Diatas 60");
162       digitalWrite (VALVE,LOW);
163     }
164     else if(Nilai >= 60)
165     {
166       Serial.println("Masuk Diatas 60");
167       Blynk.email("doniramadhan218@gmail.com", "Blynk", "Kelembapan Sudah Diatas 60!");
168       Blynk.notify("Kelembapan Sudah Diatas 60");
169       digitalWrite (VALVE,LOW);
170     }
171     Trigger = false;
172   }
173   else if(Jam != 8 && Jam!= 16)
174   {
175     Trigger = true;
176     Serial.println("Masuk True");
177   }
178   else
179   {
180   }
181 }
182 }

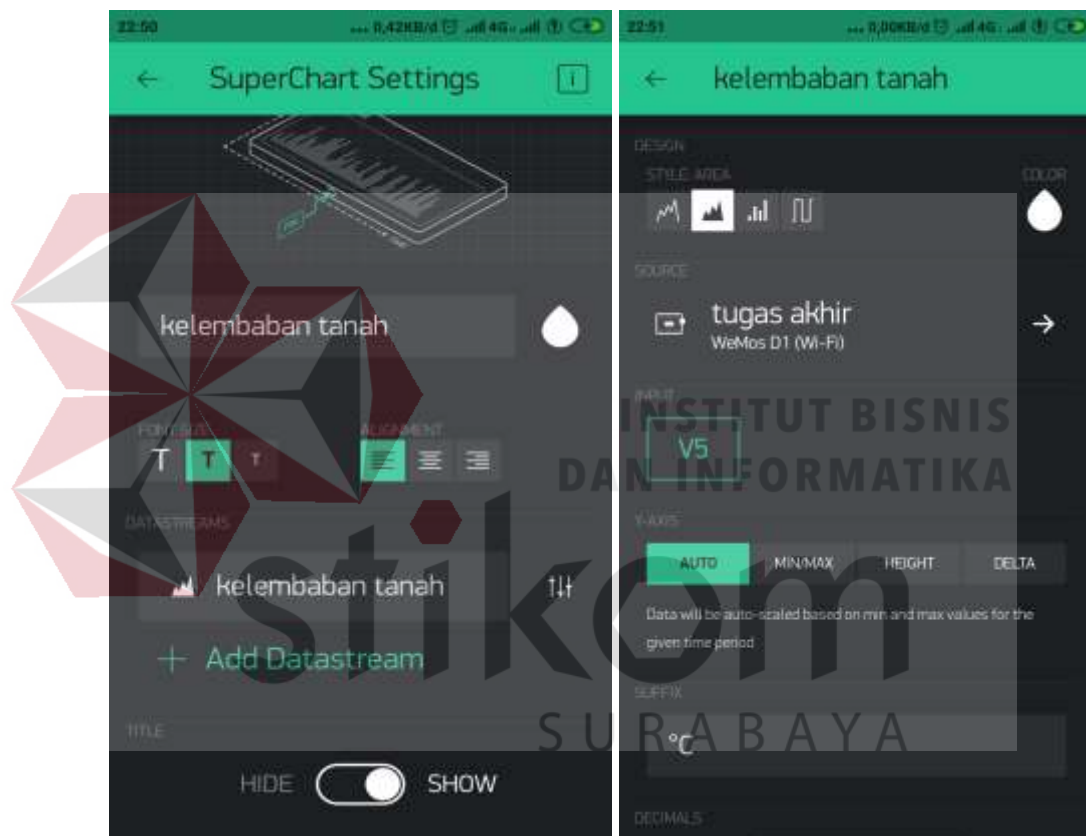
```

Gambar 3.28 Program Notifikasi Email

Pada Gambar 3.28 merupakan program agar *nitification Email* pada aplikasi *Blynk* dapat digunakan sesuai fungsinya.

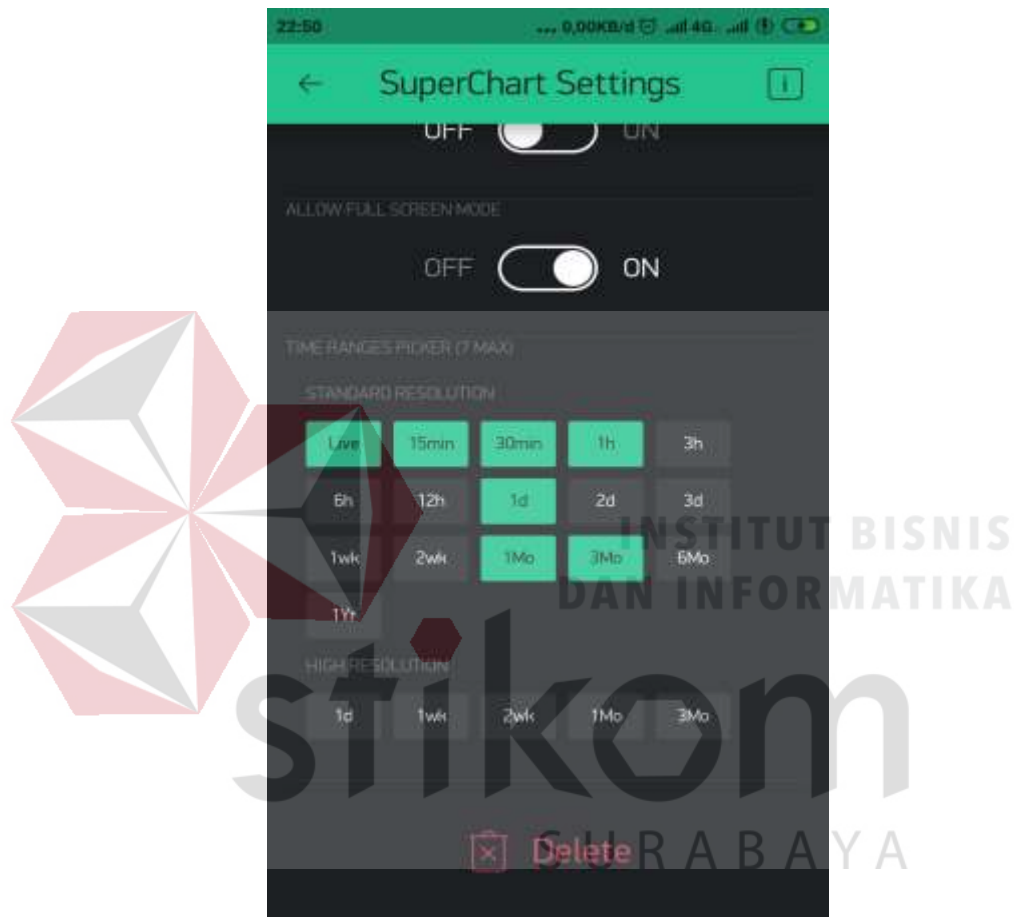
e. *SuperChart*

Pada Gambar 3.29 untuk pengaturan nama menjadi kelembaban tanah. penambahan *Datastream* digunakan untuk menampilkan nilai V5 yang diambil menjadi grafik. Perangkat yang digunakan dipilih sesuai dengan alat yang digunakan.



Gambar 3.29 *Setting SuperChart*

Pada Gambar 3.30 pengaturan waktu dapat diatur sesuai keinginan dari *user*. Kelebihan dari *widget SuperChart* adalah pengaturan waktu yang dapat diatur satu tahun.



Gambar 3.30 *Setting Waktu SuperChart*

Pada Gambar 3.31 merupakan program agar *SuperChart* kelembaban tanah pada aplikasi *Blynk* dapat muncul sesuai data pada *soil moisture sensor*.

```
#include <EasyDDNS.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <DS3231.h>
#include <LiquidCrystal_I2C.h>

#define VALVE D5
#define BLYNK_PRINT Serial
#define SensorPin A0
#define Button1 D6
#define Button2 D7
#define Button3 D4

LiquidCrystal_I2C lcd(0x27, 16, 2);

WiFiClient client;
WiFiServer server(80);

RTClib RTC;
DS3231 Clock;

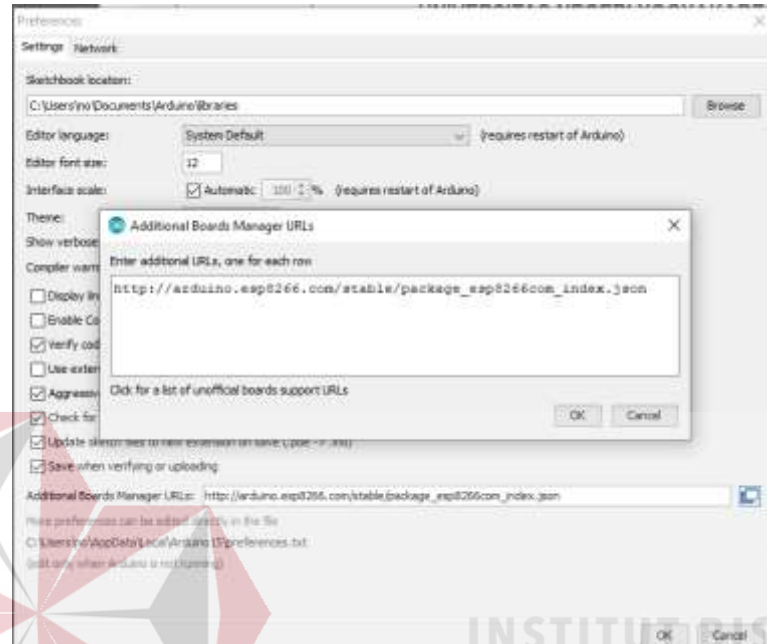
char Auth[] = "d7deaed9c17a424fa523eaa94523a874";
char Ssid[] = "STIKOMNet";
char Pass[] = "Stlk0MSoerabaja";

void SensorKelembapan(){
  Value = analogRead(SensorPin);
  Nilai = 100-(Value*100)/1023 ;
  Blynk.virtualWrite(V5, Nilai); //BLYNK_READ(V5);
  Serial.print("Nilai kelembaban: ");
  Serial.print(Nilai);
  Serial.println("");
  Serial.println();
}
```

Gambar 3. 31 Program Kelembaban Tanah

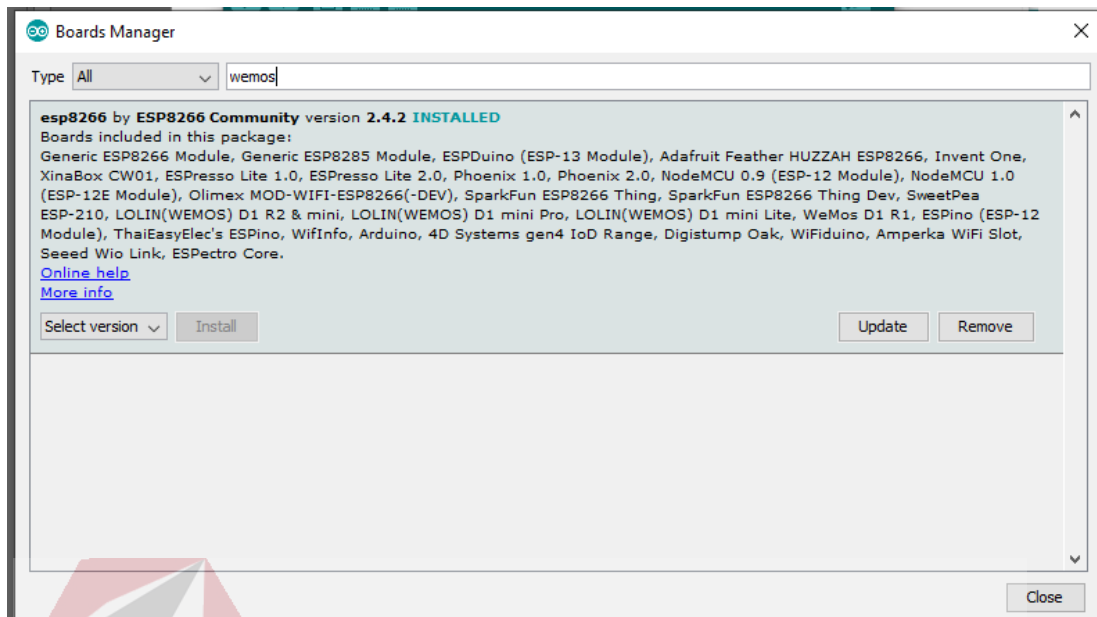
3.5.3 Konfigurasi Arduino IDE

a. Instalasi addon ESP8266 dengan *Arduino Board Manager*



Gambar 3.32 Instalasi addon ESP8266 dengan *Arduino Board Manager*

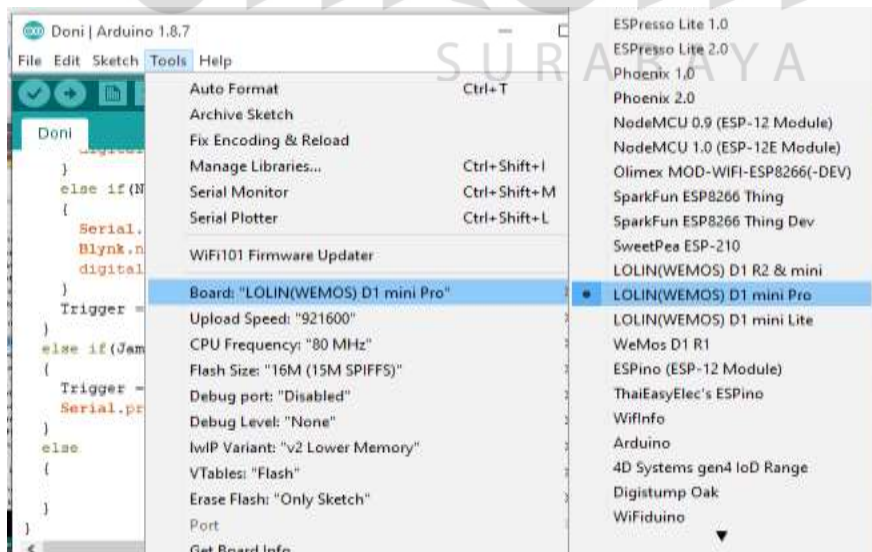
Pada Gambar 3.32 Untuk install library ESP8266 buka Arduino IDE kemudian masuk ke menu preferences (**File > Preferences**). Kemudian dibagian bawah jendela masukkan url berikut ini ke Additional Board Manager URLs. Setelah menambahkan URL pada Additional Board Manager kemudian masuk ke menu Board Manager (**Tools > Boards > Boards Manager**). Kemudian cari ESP8266 pada Filter untuk memudahkan pencarian. Setelah itu klik Install seperti pada Gambar 3.33.



Gambar 3.33 Instal Wemos Board Manager

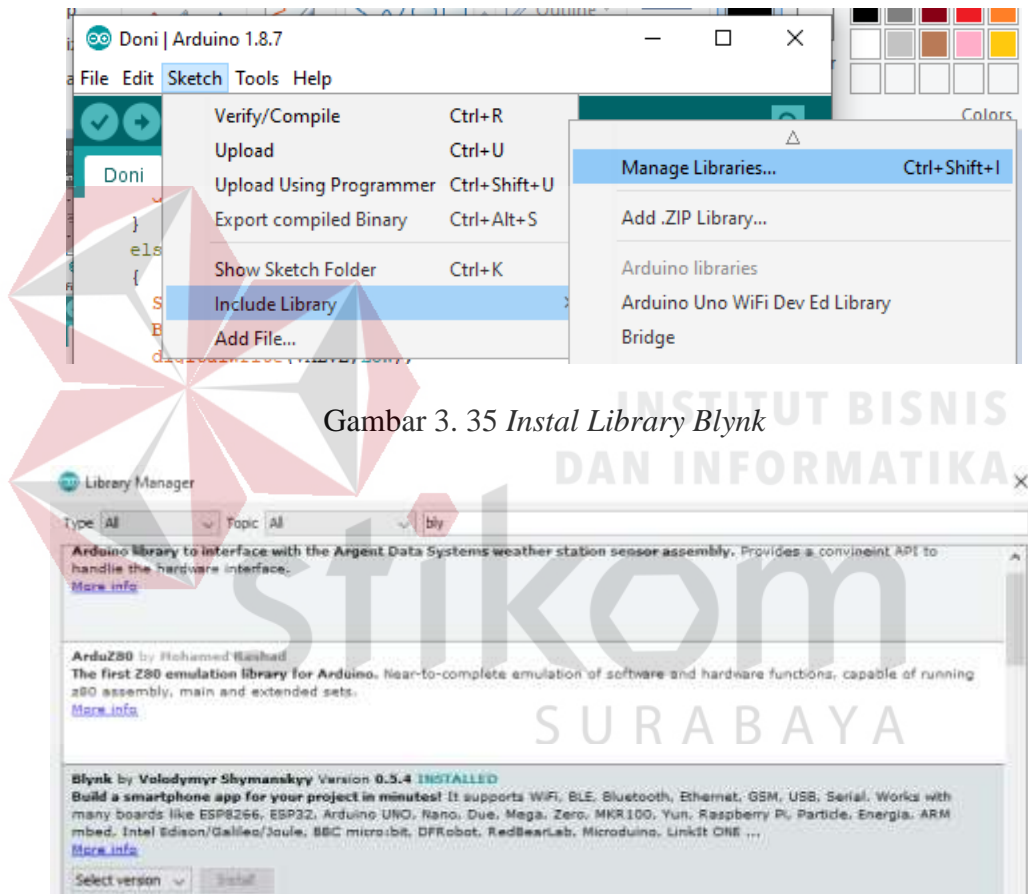
b. Memilih *board* Wemos D1 R2

Setelah menambahkan *board* Wemos D1 R2 kedalam IDE selanjutnya memilih board Wemos D1 R2 seperti pada Gambar 3.34

Gambar 3.34 Pemilihan *Board* Wemos D1 R2

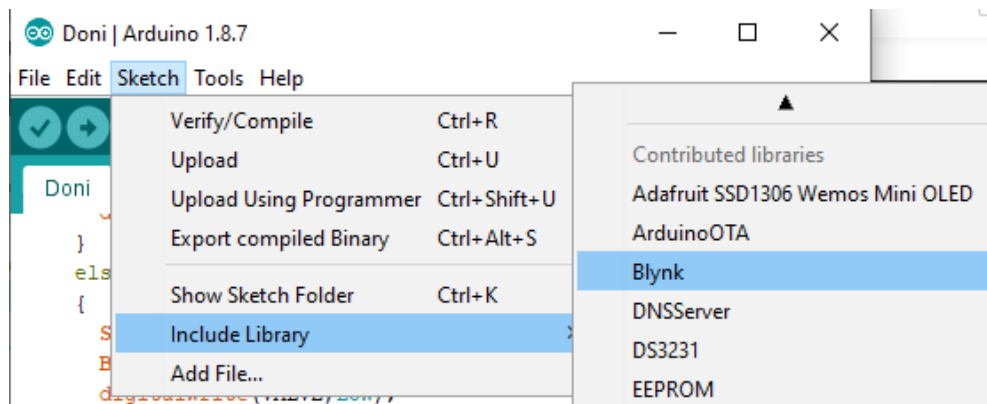
c. *Instal Library Blynk*

Pada Gambar 3.35 saat memprogram Wemos D1 R2 yang mampu berkomunikasi dengan *Blynk* digunakan *library Blynk*. Library Blynk untuk arduino dapat diunduh melalui menu *Sketch* → *Include Library* → *Manage Libraries*. Dilanjutkan dengan pencarian keyword *Blynk* pada isian pencarian.



Gambar 3. 35 *Instal Library Blynk*

Gambar 3.36 *Manager Library*



Gambar 3. 37 *Include Library Blynk*

Apabila instalasi *library Blynk* sudah selesai *Menu Library Blynk* pada *Sketch* → *Include Library* → *Blynk* sudah tersedia pada Gambar 3.36 dan pada Gambar 3.37.

3.5.4 Program Koneksi Wemos D1 R2 dengan WiFi

Pada Gambar 3.38 merupakan kode digunakan untuk menghubungkan atau mengkoneksikan Wemos D1 R2 dengan *WiFi*. Hal ini dilakukan supaya alat dapat mengakses koneksi *internet*.

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <DS3231.h>

char Auth[] = "d7deaed9c17a424fa523eaa94523a874";
char Ssid[] = "STIKOMNet";
char Pass[] = "Stlk0MSoerabaja";
```

```

void Connect() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(Ssid, Pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(50);
    Serial.print(".");
  }
  Serial.println(WiFi.localIP()); // Print the IP address
}

```

Gambar 3.38 Kode Koneksi Wemos D1 R2 dengan WiFi

Pada Gambar 3.38 pemanggilan *library* `#include <ESP8266WiFi.h>` dan `#include <BlynkSimpleEsp8266.h>` digunakan supaya dapat menggunakan fungsi WiFi pada *module* serta penghubung antara perangkat dengan aplikasi *Blynk*. Untuk perintah `WiFi.status()` berfungsi untuk melihat *status* WiFi pada *module*. Untuk perintah `WiFi.begin(Ssid, Pass);` berfungsi untuk menghubungkan *module* dengan SSID WiFi, dimana *ssid* adalah nama dari SSID WiFi dan *pass* adalah Password dari WiFi yang akan dihubungkan dengan *module* Wifi ESP8266.

3.5.5 Program Wemos D1 R2 Untuk mengirim Data ke Web Server

Program ini digunakan oleh Wemos D1 R2 untuk mengirim data dari Wemos ke *web server*. *Web server* yang digunakan pada tugas akhir ini adalah *Blynk Cloud*. Program ini hanya dapat bekerja sebagaimana mestinya jika *module* Wemos D1 R2 sudah terhubung dengan WiFi dan dapat digunakan untuk mengakses *internet*. Program dapat dilihat dibawah ini.

```

#include <WiFiServer.h>
#include <WiFiClient.h>
#include <EasyDDNS.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <DS3231.h>

void Connect() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(Ssid, Pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(50);
    Serial.print(".");
  }
  Serial.println(WiFi.localIP()); // Print the IP address
  server.begin();
  EasyDDNS.service("blynk");
  EasyDDNS.client("blynk-cloud.com", "d7deaed9c17a424fa523eaa94523a874 n f");
}

```

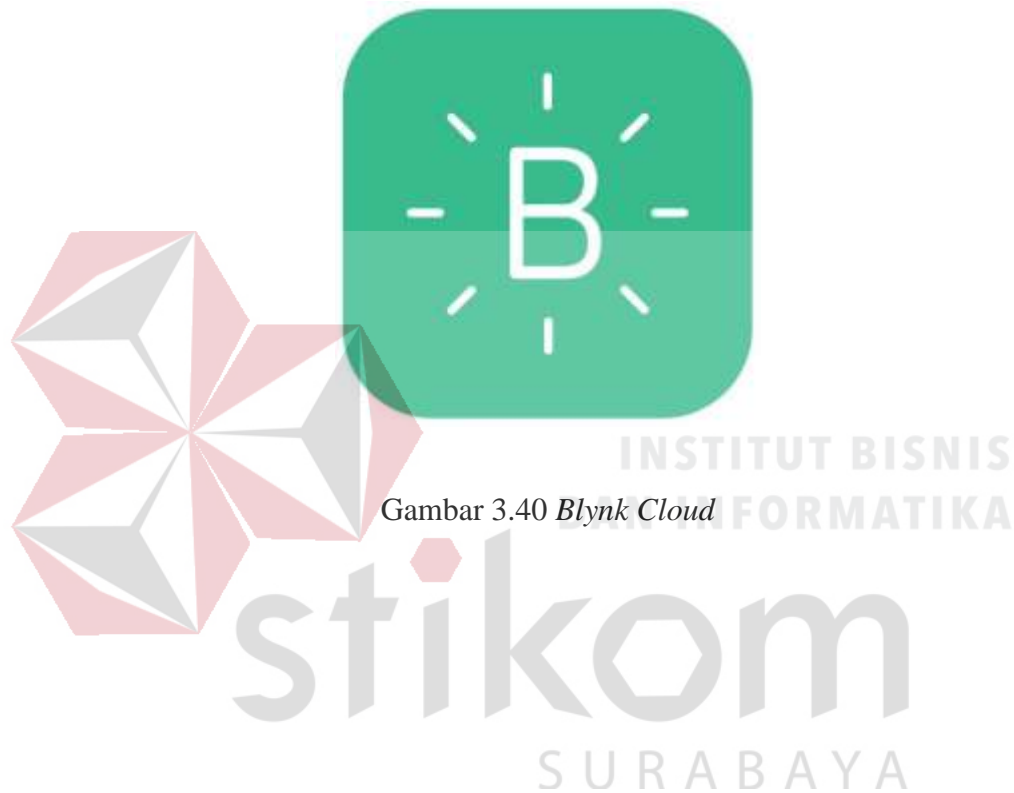
Gambar 3.39 Kode Mengirim Data Web Server pada Wemos D1 R2

Pada Gambar 3.39 pemanggilan library `#include <EasyDDN.h>` bertujuan supaya dapat menggunakan fungsi Blynk Cloud. `EasyDDNS.service('blynk')` digunakan untuk domain. `EasyDDNS.client("blynk-cloud.com", "d7deaed9c17a424fa523eaa94523a874")` berfungsi untuk mengirimkan data kepada Blynk Cloud dan Auth Token berfungsi sebagai penghubung antara perangkat dan server.

3.5.6 Pengaturan Web Server

Pada Gambar 3.40 Blynk Cloud yang berfungsi untuk menyimpan data dan sebagai perantara *monitoring*. Blynk bertanggung jawab untuk mengatur komunikasi antara aplikasi *smart phone* dengan lingkungan *hardware*. Kemampun untuk

menangani puluhan *hardware* pada saat yang bersamaan semakin memudahkan bagi para pengembang sistem IoT. Keunggulan dari *Blynk Cloud* adalah *web server* ini sudah menjadi satu dengan aplikasi *Blynk*, *user* hanya memberikan kode *Auth Token* dan mencantumkan alamat *web server*.



Gambar 3.40 *Blynk Cloud*

BAB IV

HASIL PENGUJIAN DAN PENGAMATAN

Dalam bab ini penulis akan menguraikan dan menjelaskan hasil analisis pengujian dari hasil penelitian tugas akhir ini yang telah dilakukan, pengujian dilakukan dalam beberapa bagian yang disusun dalam urutan dari yang sederhana menuju sistem yang lengkap. Pengujian dilakukan meliputi pengujian perangkat lunak (*software*) aplikasi *smartphone Android* dan perangkat keras (*hardware*) meliputi *Wemos D1 R2*, diharapkan hasil yang didapat adalah suatu sistem yang dapat menjalankan rancangan alat berjalan dengan baik, optimal, dan bermanfaat.

4.1 Pengujian Wemos D1 R2

Pada sub bab ini menjelaskan tentang pengujian pada perangkat keras *Wemos D1 R2* yang telah dilakukan. Penjelasan meliputi tujuan pengujian, alat yang dibutuhkan, prosedur pengujian dan hasil yang didapatkan dari pengujian.

4.1.1 Tujuan Pengujian Wemos D1 R2

Pengujian perangkat *Wemos D1 R2* bertujuan untuk mengetahui apakah *Wemos D1 R2* dapat berjalan dengan baik dan memastikan *Wemos D1 R2* yang dipakai tidak ada kerusakan sehingga dapat digunakan sesuai yang diharapkan.

4.1.2 Alat Pengujian Wemos D1 R2

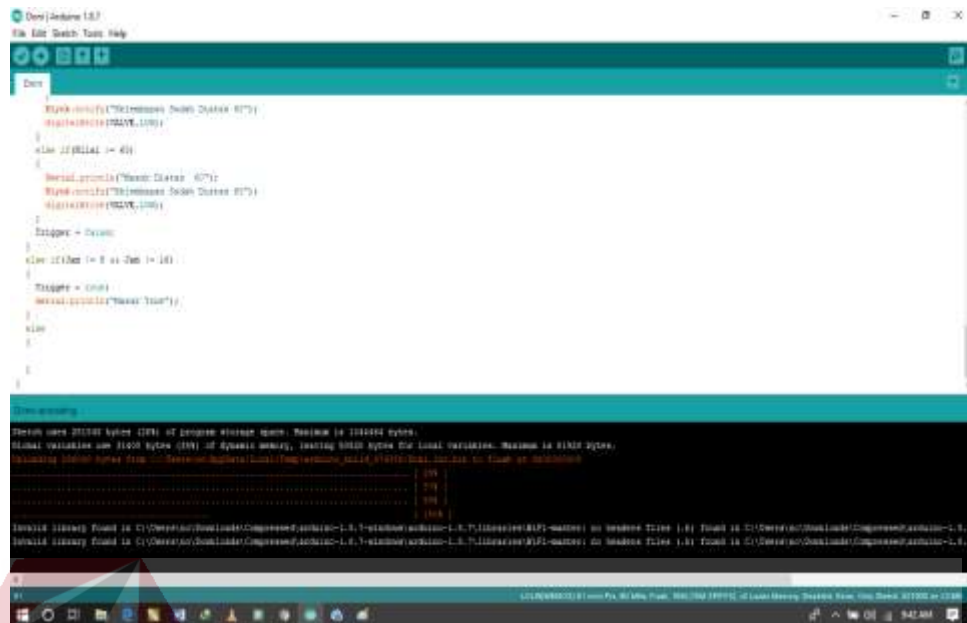
1. Komputer / Laptop
2. *Wemos D1 R2*
3. Kabel *usb type B*

4.1.3 Prosedur Pengujian Wemos D1 R2

1. Menghubungkan *Wemos D1 R2* dengan kabel *usb type B*.
2. Selanjutnya menyalakan komputer/laptop dan jalankan program *Arduino IDE*.
3. Menghubungkan kabel *usb* dengan komputer/laptop.
4. Membuka *sketch* yang akan digunakan untuk di *upload* kedalam *Wemos D1 R2*.
5. Mengatur *board*, *Serial port*, dan *programmer* sesuai dengan yang digunakan.
6. Kemudian *upload sketch* dan tunggu hingga selesai.
7. Setelah *upload* selesai akan diketahui apakah program berhasil di *download* atau tidak oleh *Wemos D1 R2*.

4.1.4 Hasil Pengujian Wemos D1 R2

Dari percobaan di atas diperoleh hasil dari proses *upload* pada jendela *comment Arduino IDE*. Apabila pada saat proses *upload* program tidak ada *comment* yang menunjukkan kegagalan atau tidak ada *comment error* dalam sambungan antara kabel *usb* dan *Wemos D1 R2*, hal itu menandakan bahwa proses berjalan dengan baik. Proses *upload* dapat dilihat pada Gambar 4.1. Apabila proses *upload* program berjalan dengan baik maka di tandai dengan tampil *comment* seperti yang ditunjukkan pada Gambar 4.2



Gambar 4.1 Tampilan Proses *Upload* dari *Arduino IDE* Kepada *Wemos D1R2*



Gambar 4.2 Tampilan *Comment* Saat Berhasil *Upload* Kepada *Wemos D1 R2*

4.2 Pengujian *Soil Moisture Sensor*

Pada sub bab ini akan menjelaskan tentang pengujian pada perangkat keras *Soil Moisture Sensor* yang telah dilakukan. Penjelasan meliputi tujuan pengujian, alat yang dibutuhkan, prosedur pengujian dan hasil yang didapatkan dari pengujian.

4.2.1 Tujuan Pengujian *Soil Moisture Sensor*

Pengujian *Soil Moisture Sensor* bertujuan untuk mengetahui *Soil Moisture Sensor* dapat berjalan dengan baik dan memastikan *Soil Moisture Sensor* yang dipakai tidak ada kerusakan sehingga data kelembaban tanah dapat digunakan sesuai yang diharapkan.

4.2.2 Alat Pengujian *Soil Moisture Sensor*

1. Komputer / Laptop
2. *Wemos D1 R2*
3. *Soil Moisture Sensor*
4. Kabel *usb type B*
5. Kabel Jumper

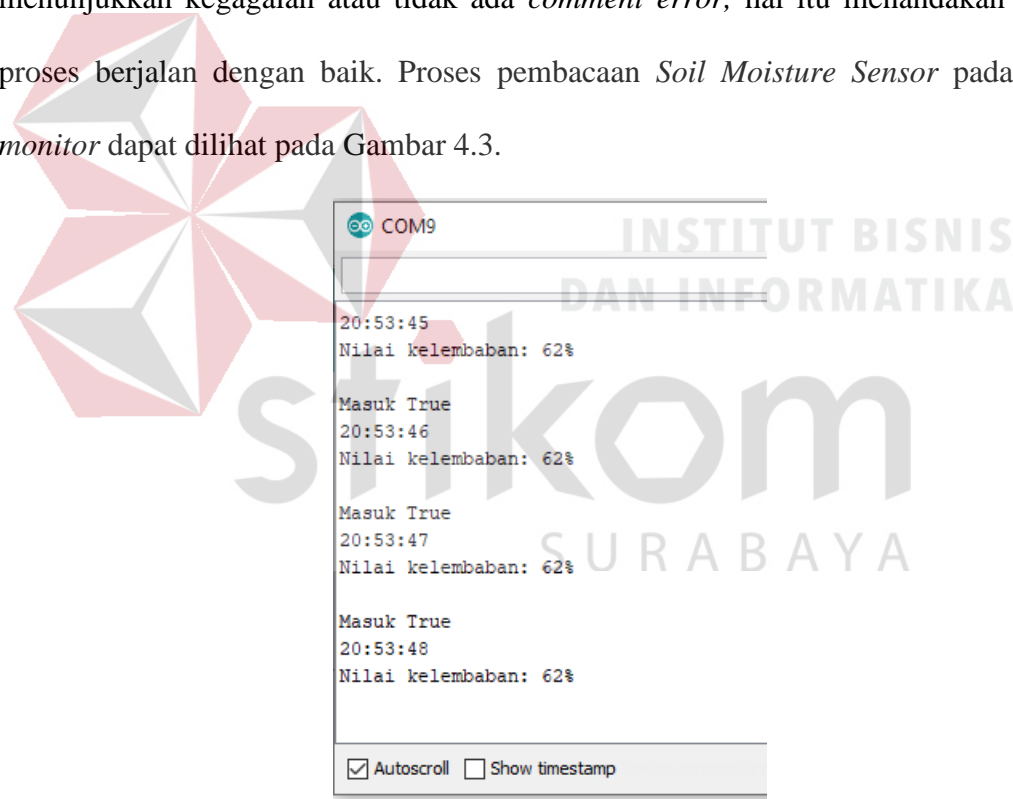
4.2.3 Prosedur Pengujian *Soil Moisture Sensor*

1. Menghubungkan *Soil Moisture Sensor* ke *Wemos D1 R2* dengan menggunakan kabel jumper.
2. Selanjutnya menyalakan komputer/laptop dan jalankan program *Arduino IDE*.
3. Menghubungkan kabel *usb type B* dengan komputer/laptop.

4. Membuka *Sketch* yang akan digunakan untuk *upload* kedalam *Wemos D1 R2*
5. *Upload Sketch* dan tunggu hingga selesai.
6. Mengatur *Board*, *Serial Port* dan *Programmer* sesuai dengan yang digunakan.

4.2.4 Hasil Pengujian *Soil Moisture Sensor*

Dari percobaan di atas diperoleh hasil dari proses *upload* pada jendela *comment* *Arduino IDE*. Apabila pada saat proses *upload* program tidak ada *comment* yang menunjukkan kegagalan atau tidak ada *comment error*, hal itu menandakan bahwa proses berjalan dengan baik. Proses pembacaan *Soil Moisture Sensor* pada *serial monitor* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Pembacaan *Soil Moisture Sensor* Pada *Serial Monitor*

4.3 Pengujian Notifikasi

4.3.1 Tujuan Pengujian Notifikasi

Pada sub bab ini menjelaskan tentang pengujian pada notifikasi aplikasi *Blynk* yang muncul pada *smartphone android* dan *Email* yang telah dilakukan. perangkat. Penjelasan meliputi tujuan pengujian, alat yang dibutuhkan, prosedur pengujian dan hasil yang didapatkan dari pengujian.

4.3.2 Alat Pengujian Notifikasi

1. Komputer / Laptop
2. *Wemos D1 R2*
3. *Smartphone Android*
4. Kabel *usb type B*
5. Kabel Jumper
6. *Soil Moisture Sensor*
7. *Real Time Clock*

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

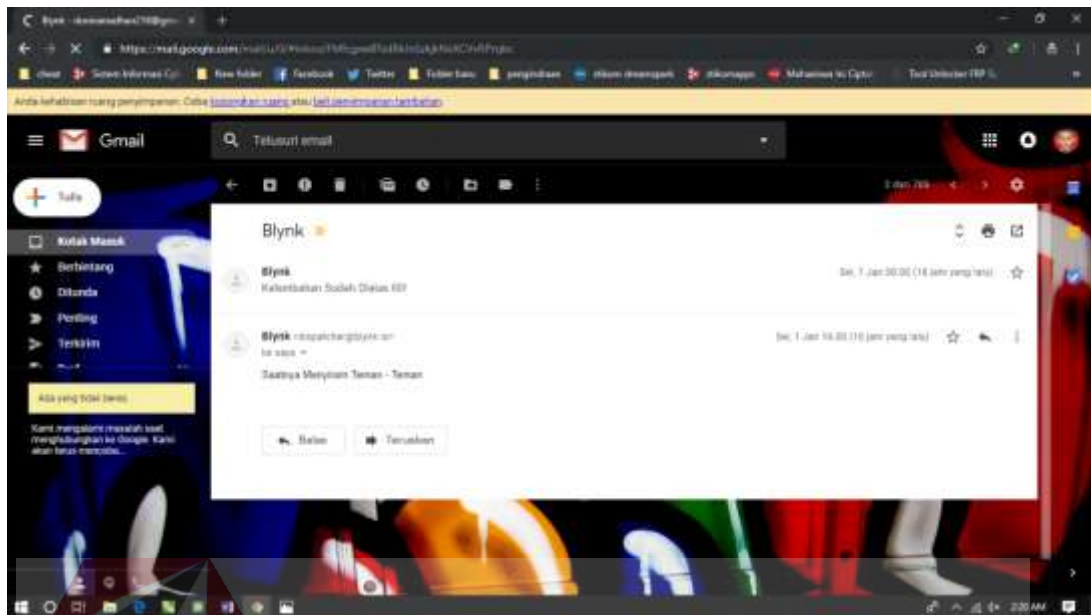
4.3.3 Prosedur Pengujian Notifikasi

1. Menyalakan laptop / komputer kemudian buka aplikasi *Arduino IDE* untuk membuka program Arduino.
2. Menghubungkan *Wemos D1 R2* dengan *soil moisture sensor* menggunakan kabel jumper sesuai dengan rangkaiannya.
3. Menghubungkan *Wemos D1 R2* dengan *real time clock* menggunakan kabel jumper sesuai dengan rangkaiannya.

4. Menghubungkan *Wemos D1 R2* dengan LCD menggunakan kabel jumper sesuai dengan rangkaiannya.
5. Membuka aplikasi *Arduino IDE*.
6. Atur *SSID* dan *password* sesuai dengan yang digunakan.
7. *Upload* program ke *Wemos D1 R2*.
8. Install aplikasi *Blynk* pada *smartphone Android*.
9. Membuka aplikasi *Blynk* kemudian memasukkan *username* dan *password* tekan tombol *Log In* untuk masuk pada menu berikutnya.
10. Melakukan pengamatan data pada aplikasi *Blynk*.
11. Menunggu notifikasi hingga muncul.

4.3.4 Hasil Pengujian Notifikasi

Dari percobaan di atas diperoleh hasil dari notifikasi yang muncul pada *smartphone Android* dan *email*. Notifikasi akan muncul pada saat jam 8 pagi dan jam 4 sore. Hal itu menunjukkan bahwa proses berjalan dengan baik. Untuk notifikasi *smartphone Android* dapat dilihat pada Gambar 4.4 dan untuk notifikasi pada *email* dapat dilihat pada Gambar 4.5.



Gambar 4.4 Notifikasi pada *Email*



Gambar 4.5 Notifikasi pada *Smartphone Android*

4.4 Pengujian *Relay* dan *Valve*

Pada sub bab ini menjelaskan tentang pengujian pada perangkat keras *Relay* dan *Valve* yang telah dilakukan. Penjelasan meliputi tujuan pengujian, alat yang dibutuhkan, prosedur pengujian dan hasil yang didapatkan dari pengujian.

4.4.1 Tujuan Pengujian *Relay* dan *Valve*

Pengujian *Relay* dan *Valve* bertujuan untuk mengetahui *Relay* dan *Valve* dapat berjalan dengan baik dan memastikan *Relay* dan *Valve* yang dipakai tidak ada kerusakan sehingga proses penyiraman dapat dilakukan sesuai yang diharapkan.

4.4.2 Alat Pengujian *Relay* dan *Valve*

1. Komputer / Laptop
2. *Wemos D1 R2*
3. *Relay*
4. *Valve*
4. Kabel *usb type B*
5. Kabel Jumper

4.4.3 Prosedur Pengujian *Relay* dan *Valve*

1. Menyalakan laptop / komputer kemudian buka aplikasi *Arduino IDE* untuk membuka program Arduino.
2. Menghubungkan *Wemos D1 R2* dengan *relay* menggunakan kabel jumper sesuai dengan rangkaiannya.
3. Menghubungkan *adaptor 12V* dengan *valve* menggunakan kabel jumper sesuai dengan rangkaiannya.
4. Menghubungkan *valve* dengan *relay* menggunakan kabel jumper sesuai dengan rangkaiannya.
5. Membuka aplikasi *Arduino IDE*.
6. Mengatur SSID dan password sesuai dengan yang digunakan.
7. *Upload* program ke *Wemos D1 R2*.
8. Memasang aplikasi *Blynk* pada *smartphone Android*.
9. Membuka aplikasi *Blynk* kemudian memasukkan username dan password lalu menekan tombol *Log In* untuk masuk pada menu berikutnya.
10. Melakukan pengujian pada aplikasi *Blynk*.

4.4.4 Hasil Pengujian *Relay* dan *Valve*

Dari percobaan di atas diperoleh hasil dari proses pengujian *relay* dan *valve*. Apabila pada saat proses penyiraman air dapat mengalir dari tempat penampung air sampai pada *sprinkle*, hal itu menandakan bahwa proses penyiraman berjalan dengan baik.



Gambar 4.6 *Relay Bernilai High / 1*

Pada Gambar 4.6 *Relay* bernilai *High* ditandai dengan lampu indikator yang menyala. Pada Gambar 4.7 *button* di aplikasi *Blynk* kondisi awal ON, apabila ditekan proses penyiraman terjadi seperti pada gambar 4.8 dan *button* menjadi OFF



Gambar 4. 7 Kontrol Valve *ON*



Gambar 4.8 Terjadi Proses Penyiraman



Gambar 4.9 *Relay* Bernilai *Low* / 0

Pada Gambar 4.9 *Relay* bernilai *Low* ditandai dengan lampu indikator yang padam. Pada Gambar 4.10 *button* di aplikasi *Blynk* kondisi awal OFF, apabila ditekan proses penyiraman tidak terjadi seperti pada gambar 4.11 dan *button* menjadi ON



Gambar 4.10 Kontrol Valve OFF



Gambar 4.11 Tidak Terjadi Proses Penyiraman

4.5 Pengujian *Real Time Clock*

Pada sub bab ini menjelaskan tentang pengujian pada perangkat keras *Real Time Clock* yang telah dilakukan. Penjelasan meliputi tujuan pengujian, alat yang dibutuhkan, prosedur pengujian dan hasil yang didapatkan dari pengujian.

4.5.1 Tujuan Pengujian *Real Time Clock*

Pengujian *Real Time Clock* bertujuan untuk mengetahui *Real Time Clock* dapat berjalan dengan baik dan memastikan *Real Time Clock* yang dipakai tidak ada kerusakan sehingga data waktu dapat digunakan sesuai yang diharapkan.

4.5.2 Alat Pengujian *Real Time Clock*

1. Komputer / Laptop
2. *Wemos D1 R2*
3. *Real Time Clock*
4. Kabel *usb type B*
5. Kabel Jumper

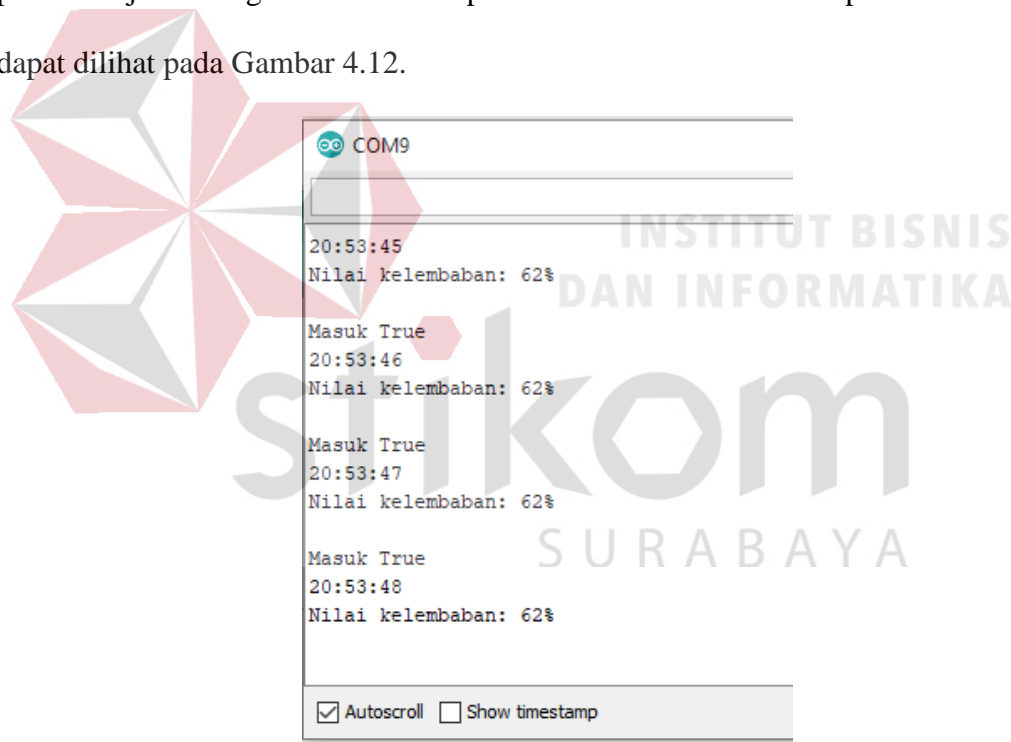
4.5.3 Prosedur Pengujian *Real Time Clock*

1. Menghubungkan *Real Time Clock* ke *Wemos D1 R2* dengan menggunakan kabel jumper.
2. Selanjutnya menyalakan komputer/laptop dan jalankan program *Arduino IDE*.
3. Menyambungkan kabel *usb type B* dengan komputer/laptop.
4. Membuka *Sketch* yang akan digunakan untuk *upload* kedalam *Wemos D1 R2*

5. *Upload Sketch* dan tunggu hingga selesai.
6. Mengatur *Board*, *Serial Port* dan *Programmer* sesuai dengan yang digunakan.

4.5.4 Hasil Pengujian *Real Time Clock*

Dari percobaan di atas diperoleh hasil dari proses *upload* pada jendela *comment* *Arduino IDE*. Apabila pada saat proses *upload* program tidak ada *comment* yang menunjukkan kegagalan atau tidak ada *comment error*, hal itu menandakan bahwa proses berjalan dengan baik. Proses pembacaan *Real Time Clock* pada *serial monitor* dapat dilihat pada Gambar 4.12.



Gambar 4.12 Tampilan Proses Pembacaan *Real Time Clock* Pada *Serial Monitor*

4.6 Pengujian Monitoring Sensor Pada Aplikasi *Blynk*

4.6.1 Tujuan Pengujian Monitoring Sensor Pada Aplikasi *Blynk*

Pengujian aplikasi *Blynk* ini dilakukan bertujuan untuk mengetahui apakah data waktu dan kelembaban tanah yang di tampilkan oleh *serial monitor* sama dengan data waktu dan kelembaban tanah yang ditampilkan oleh aplikasi *Blynk*. Sehingga dapat melihat validasi data yang ditampilkan pada *hardware* dan *software* bisa sama sesuai dengan apa yang ditampilkan pada *serial monitor Arduino IDE* dengan aplikasi *Blynk*. Dalam pengujian ini diharapkan tampilan pada aplikasi *Blynk* bisa sama dengan tampilan *serial monitor* dan *LCD*.

4.6.2 Alat Pengujian Monitoring Sensor pada Aplikasi *Blynk*

1. *Soil Moisture Sensor*
2. *Wemos D1 R2*
3. Laptop / komputer
4. Kabel usb *type B*
5. *Smartphone Android*
6. Aplikasi *Blynk*
7. Koneksi Internet
8. LCD
9. Kabel Jumper

4.6.3 Prosedur Pengujian Monitoring Sensor pada Aplikasi Blynk

1. Menyalakan laptop / komputer kemudian buka aplikasi *Arduino IDE* untuk membuka program Arduino.
2. Menghubungkan *Wemos D1 R2* dengan *soil moisture sensor* menggunakan kabel jumper sesuai dengan rangkaiannya.
3. Menghubungkan *Wemos D1 R2* dengan *real time clock* menggunakan kabel jumper sesuai dengan rangkaiannya.
4. Menghubungkan *Wemos D1 R2* dengan LCD menggunakan kabel jumper sesuai dengan rangkaiannya.
5. Membuka aplikasi *Arduino IDE*.
6. Mengatur SSID dan password sesuai dengan yang digunakan.
7. Upload program ke *Wemos D1 R2*.
8. Memasang aplikasi *Blynk* pada *smartphone Android*.
9. Membuka aplikasi *Blynk* kemudian masukkan username dan password tekan tombol *Log In* untuk masuk pada menu berikutnya.
10. Melakukan pengamatan data kelembaban dan waktu pada data yang keluar pada aplikasi *Blynk*.

4.6.4 Hasil Pengujian Monitoring Sensor pada Aplikasi Blynk

Dari prosedur diatas dilakukan pengujian dengan mengamati dan membandingkan data suhu dan kelembaban pada *serial monitor*, *LCD* dan aplikasi *Blynk*. Dari perbandingan data yang diperoleh tersebut dapat dilihat apakah data pada aplikasi *Blynk* sama dengan data yang diperoleh oleh *serial monitor* dan *LCD*. Dapat

dilihat pada gambar 4.5. Yaitu tampilan *LCD* saat memperoleh data kelembaban tanah dan waktu dari *soil moisture sensor* dan *real time clock* dan kemudian akan dikirimkan pada aplikasi *blynk* melalui bantuan koneksi *internet* dan *web server*. Dan dapat dilihat juga pada saat mengamati data antara *LCD* dengan aplikasi *Blynk* pada Gambar 4.13 dan Gambar 4.14.



Gambar 4.13 Tampilan Data pada *LCD*



Gambar 4.14 Tampilan Data pada Aplikasi *Blynk*

Dari data *serial monitor* dapat dilakukan pengujian data dari *soil moisture sensor* dan *real time clock* sebanyak 10 data kelembaban tanah dan waktu untuk membandingkan data tersebut dengan *LCD* dan aplikasi *Blynk* pada Gambar 4.14 dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Pengujian Monitoring pada Aplikasi *Blynk*

Data Alat		Data Blynk		Status	
Detik	Data	Detik	Data		
26	66	26	66	fix	3s
27	64	26	66	delay	
29	64	26	66	delay	
30	64	29	64	delay	5s
31	64	31	64	fix	
32	66	32	64	delay	
33	63	32	64	delay	
35	64	32	64	delay	
36	64	32	64	delay	
37	64	32	64	delay	
rata - rata delay					0.8s

Hasil pegujian pada tabel diatas dapat dilihat bahwa ada 20% data pegujian monitoring yang tidak mengalami *delay* antara *serial monitor* dan *LCD* dengan aplikasi *Blynk*. Terdapat 80% data pengujian mengalami *delay* melalui sistem ini ke aplikasi *Blynk*. Nilai rata – rata *delay* bernilai sebesar 0.8s.

4.7 Pengujian Monitoring Jarak Jauh Alat dengan *Smartphone Android*

4.7.1 Tujuan Monitoring Jarak Jauh Alat dengan *Smartphone Android*

Pengujian aplikasi *Blynk* ini dilakukan bertujuan untuk mengetahui apakah data waktu dan kelembaban tanah yang di tampilkan oleh *serial monitor* sama dengan data waktu dan kelembaban tanah yang ditampilkan oleh aplikasi *Blynk* secara jarak jauh. Monitoring jarak jauh berfungsi apabila *user* tidak berada ditempat. *user* masih bisa memantau secara jarak jauh alat yang digunakan. *Sehingga* dapat melihat validasi data yang ditampilkan pada *hardware* dan *software* bisa sama sesuai dengan apa yang ditampilkan pada *serial monitor Arduino IDE* dengan aplikasi *Blynk*. Dalam pengujian ini diharapkan tampilan pada aplikasi *Blynk* bisa sama dengan tampilan *serial monitor* dan *LCD*.

4.7.2 Alat Pengujian Monitoring Jarak Jauh Alat dengan *Smartphone Android*

4.7.3 Prosedur Pengujian Monitoring Jarak Jauh Alat dengan *Smartphone Android*

1. Menyalakan laptop / komputer kemudian buka aplikasi Arduino IDE untuk membuka program Arduino.
2. Menghungkan *Wemos D1 R2* dengan *soil moisture sensor* menggunakan kabel jumper sesuai dengan rangkaiannya.
3. Menghubungkan *Wemos D1 R2* dengan *real time clock* menggunakan kabel jumper sesuai dengan rangkaiannya.

4. Menghubungkan *Wemos D1 R2* dengan LCD menggunakan kabel jumper sesuai dengan rangkaiannya.
5. Membuka aplikasi *Arduino IDE*.
6. Mengatur SSID dan password sesuai dengan yang digunakan.
7. *Upload* program ke *Wemos D1 R2*.
8. Install aplikasi *Blynk* pada *smartphone Android*.
9. Membuka aplikasi *Blynk* kemudian masukkan username dan password tekan tombol *Log In* untuk masuk pada menu berikutnya.
10. Melakukan pengamatan data kelembaban dan waktu pada data yang keluar pada aplikasi *Blynk*.

4.7.4 Hasil Pengujian Monitoring Jarak Jauh Alat dengan *Smartphone Android*

Dari prosedur diatas dilakukan pengujian dengan mengamati dan membandingkan data suhu dan kelembaban pada *serial monitor*, *LCD* dan aplikasi *Blynk* secara jarak jauh. Jarak pada pengujian ini sejauh 36 km. Pada jarak dapat lihat pada Gambar 4.17 dan pada posisi dapat dilihat pada Gambar 4.18. Dari perbandingan data yang diperoleh tersebut dapat dilihat apakah data pada aplikasi *Blynk* pada saat jarak jauh sama dengan data yang diperoleh oleh *serial monitor* dan *LCD*. Dapat dilihat pada Gambar 4.15 yaitu tampilan *LCD* saat memperoleh data kelembaban tanah dan waktu dari *soil moisture sensor* dan *real time clock* dan kemudian akan dikirimkan pada aplikasi *blynk* melalui bantuan koneksi *interner* dan *web server*. Dan dapat dilihat

juga pada saat mengamati data antara *LCD* dengan aplikasi *Blynk* pada Gambar 4.15 dan Gambar 4.16.



Gambar 4.15 Tampilan Data pada LCD

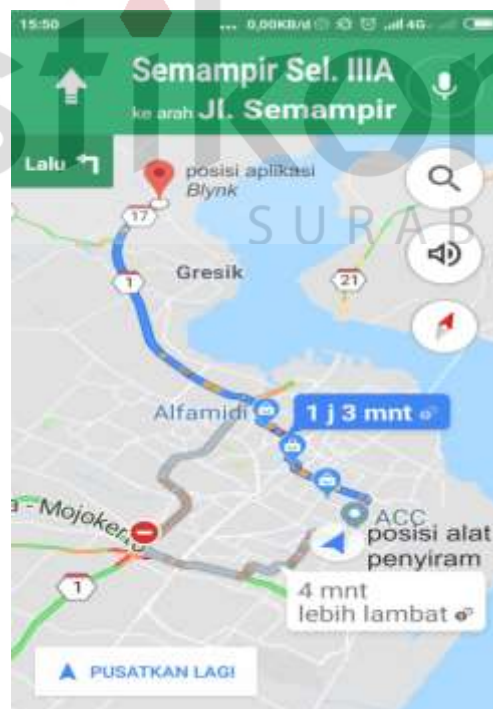


Gambar 4.16 Tampilan Data pada Aplikasi *Blynk*

Dari data *serial monitor* dapat dilakukan pengujian data dari *soil moisture sensor* dan *real time clock* sebanyak 100 data kelembaban tanah dan waktu dengan jarak 36 km seperti pada Gambar 4.17 dan Gambar 4.18 untuk membandingkan data tersebut dengan *LCD* dan aplikasi *Blynk* secara jarak jauh dapat dilihat pada Tabel 4.2.



Gambar 4.17 Jarak Monitoring antara Alat dan Aplikasi



Gambar 4.18 Posisi Alat dan Aplikasi Blynk

Tabel 4.2 Pengujian Monitoring pada Alat Secara Jarak Jauh

Data Alat		Data Blynk		Status	
Detik	Data Kelembaban	Detik	Data Kelembaban		
26	66	26	66	fix	
27	64	26	66	delay	3s
29	64	26	66	delay	
30	64	29	64	delay	
31	64	31	64	fix	
32	66	32	64	delay	5s
33	63	32	64	delay	
35	64	32	64	delay	
36	64	32	64	delay	
37	64	32	64	delay	
38	62	38	62	fix	
39	66	39	66	fix	
41	64	41	64	fix	
42	65	41	64	delay	12s
43	64	41	64	delay	
44	64	41	64	delay	
45	63	41	64	delay	
46	63	41	64	delay	
48	53	41	64	delay	
49	66	41	64	delay	
50	64	41	64	delay	
51	62	41	64	delay	
52	65	51	62	delay	
53	64	53	65	delay	
55	64	55	64	fix	
56	64	56	64	fix	
57	65	56	64	delay	5s
58	65	56	64	delay	
59	66	56	64	delay	
1	66	56	64	delay	
2	64	2	64	fix	
3	64	3	64	fix	
4	65	4	65	fix	
5	63	4	65	delay	13s
6	65	4	65	delay	

Data Alat		Data Blynk		Status	
Detik	Data Kelembaban	Detik	Data Kelembaban		
7	65	4	65	delay	
8	65	4	65	delay	
9	64	4	65	delay	
10	64	4	64	delay	
12	64	4	64	delay	
13	64	4	64	delay	
15	65	4	63	delay	
16	65	4	64	delay	
17	63	4	63	delay	
18	63	4	63	delay	
19	64	19	64	fix	
20	63	20	63	fix	
21	63	21	63	fix	
23	64	21	63	delay	3s
24	64	21	63	delay	
25	64	21	63	delay	
26	64	21	63	delay	
27	63	27	63	fix	
28	63	28	63	fix	
30	63	28	63	delay	8s
31	65	28	63	delay	
32	64	28	63	delay	
33	65	28	63	delay	
34	62	28	63	delay	
36	65	28	63	delay	
37	65	28	63	delay	
38	64	28	63	delay	
39	65	39	65	fix	
40	65	39	65	delay	1s
41	65	41	65	fix	
43	65	41	65	delay	10s
44	65	41	65	delay	
45	64	41	65	delay	
46	65	41	65	delay	
47	64	41	65	delay	
48	65	41	65	delay	
48	64	48	65	delay	

Data Alat		Data Blynk		Status	
Detik	Data Kelembaban	Detik	Data Kelembaban		
50	64	48	65	delay	
51	64	48	65	delay	
52	64	50	64	delay	
53	64	53	64	fix	
54	64	53	64	delay	28s
56	64	53	64	delay	
57	63	53	64	delay	
58	65	53	64	delay	
59	64	53	64	delay	
0	65	53	64	delay	
2	64	53	64	delay	
3	65	53	64	delay	
4	64	53	64	delay	
5	66	53	64	delay	
6	62	53	64	delay	
7	66	0	64	delay	
9	63	0	64	delay	
10	65	0	64	delay	
11	64	0	64	delay	
12	66	0	64	delay	
13	63	0	64	delay	
14	66	12	66	delay	
16	63	12	66	delay	
17	66	12	66	delay	
18	64	12	66	delay	
19	65	12	66	delay	
20	64	12	66	delay	
22	64	12	66	delay	
rata - rata delay					0.88s

Hasil pengujian pada tabel diatas dapat dilihat bahwa ada 18% data pengujian monitoring yang tidak mengalami *delay* antara *serial monitor* dan *LCD* dengan aplikasi *Blynk*. Terdapat 82% data pengujian mengalami *delay* melalui sistem ini ke aplikasi *Blynk*. Nilai rata – rata *delay* bernilai sebesar 0.88s.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan sistem dan seluruh pengujian yang telah dilakukan untuk semua kondisi yang mungkin terjadi pada kontrol penyiram taman otomatis dan monitoring menggunakan Android, maka dapat diambil kesimpulan sebagai berikut:

1. *Soil moisture sensor* dan *real time clock* pada *Wemos D1 R2* yang ditampilkan pada *serial monitor* dan LCD untuk mengolah data menunjukkan tingkat keberhasilan pembacaan kelembaban tanah dan waktu sebesar 100%.
2. Monitoring jarak jauh dengan jarak 36 km menunjukkan 18% data pengujian monitoring yang tidak mengalami *delay* antara *serial monitor* dan LCD dengan aplikasi *Blynk*. Terdapat 82% data pengujian mengalami *delay* melalui sistem ini ke aplikasi *Blynk*. Nilai rata – rata *delay* bernilai sebesar 0.88s.
3. Notifikasi tingkat kelembaban dan proses penyiraman muncul pada jam 8 pagi dan 4 sore pada *smartphone Android* dan *Email*.

5.2 Saran

Pengembangan lebih lanjut mengenai penelitian Tugas Akhir ini, maka ada beberapa saran sebagai berikut:

1. Gunakan koneksi internet yang stabil untuk proses pengiriman agar data yang ditampilkan pada aplikasi lebih stabil dan akurat
2. Pengaplikasian penyiraman dilakukan pada taman sesungguhnya.

DAFTAR PUSTAKA

- Alvin. (2018). *www.coursehero.com*. Retrieved from <https://www.coursehero.com/file/9786652/Cara-Kerja-Mesin-Pompa-Air-Otomatis-SONI/>
- Arif, N. (2012). *Instalasi dan Trouble Shooting Speedy*.
- Indonesia, P. C. (2015, Agustus 20). *Pengertian Web Server dan Fungsinya*. Retrieved from ID Cloud Hos: <https://idcloudhost.com/pengertian-web-server-dan-fungsinya/>
- Jamzuri, E. R. (2016, Agustus 5). *Cara Memprogram Wemos D1 R2 Mini ESP8266 Dengan Arduino*. Retrieved from Eko - Rudiawan.com: <http://eko-rudiawan.com/cara-memprogram-wemos-esp8266-dengan-arduino/>
- Langi, S. I., Wuwung, J., & Lumenta, A. S. (2014). Kipas Angin Otomatis. *E-Journal Teknik Elektro dan Komputer*, 45.
- Mahali, M. I. (2017). *Menghubungkan ESP8266 dengan BLYNK*. Yogyakarta.
- Oktavianus, R. (2017). Desain dan Implementasi Sistem Monitoring Kelembaban Tanah Berbasis Android. *Jurnal Teknik Infomatikan Universitas Halu Oleo Kendari*.
- P., U. C. (2018, September 14). *Pengetian Solenoid Valve*. Retrieved from valvejual: <http://www.valvejual.com/pengertian-solenoid-valve/>

Pambudi, K. J. (2014). Rancang Bangun Wireless Sensor Network untuk Monitoring Suhu dan Kelembaban pada Lahan Tanaman Jarak. *Journal of Control and Network System*, Vol. 3 No. 2, STIKOM Surabaya, hal 09-17.

sH.N.S. (2012). Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis. *Informatika Bandung*.

Wahyu Adi Prayitno, A. M. (2017). Sistem Monitoring Suhu, Kelembaban, dan Pengendali Penyiraman Tanaman Hidroponik menggunakan Blynk Android.

Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 1, hal 292-297.

