

SISTEM MONITORING KEBUN SAWI VIA TELEGRAM

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

AFIF

NIRM : 0031531

FERDIANSYAH

NIRM : 0031542

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG**

2018

LEMBAR PENGESAHAN

SISTEM *MONITORING* KEBUN SAWI *VIA TELEGRAM*

Oleh :

Afif

NIRM : 0031531

Ferdiansyah

NIRM : 0031542

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



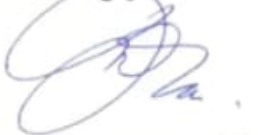
Eko Sulistyono, M.T.

Pembimbing 2



Riki Ariansyah, M.T.

Penguji 1



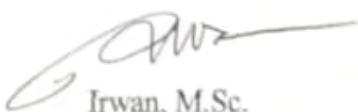
Indra Dwisaputra, M.T.

Penguji 2



Ocsirendi, M.T.

Penguji 3



Irwan, M.Sc.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini :

Nama Mahasiswa 1 : Afif

NIRM : 0031531

Nama Mahasiswa 2 : Ferdiansyah

NIRM : 0031542

Dengan Judul: *SISTEM MONITORING KEBUN SAWI VIA TELEGRAM*

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia diberikan sanksi yang berlaku.

Sungailiat, 16 Juli 2018

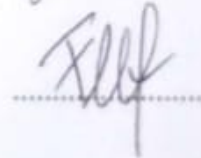
Nama Mahasiswa

Tanda Tangan

1. Afif



2. Ferdiansyah



ABSTRAK

Sawi Hijau (Brassica rapa L) merupakan salah satu komoditas tanaman hortikultura dari jenis sayuran yang memiliki macam – macam manfaat dan kegunaan dalam kehidupan masyarakat sehari – hari dan mudah dalam perawatan sehingga banyak dibudidayakan. Dengan aplikasi Telegram yang digunakan sebagai media komunikasi antara petani dan kebunnya. Dan menggunakan Mikrokontroller jenis Arduino Mega 2560 sebagai pengolahan data sensor petani dapat memantau kelembaban tanah, kelembaban udara, serta suhu kapan pun dan dimanapun, serta menggunakan Mikrokontroller jenis NodeMCU sebagai jembatan penghubung ke jaringan internet. Diharapkan dari sistem monitoring yang dibuat dapat memudahkan pekerjaan petani dalam memantau penyiraman kebun yang dimilikinya.

Kata kunci : Sawi Hijau, Monitoring, Jarak dan Waktu, Telegram, Mikrokontroller.

ABSTRACT

Green mustard (Brassica rapa L) is one of the commodities of horticulture from the type of vegetable which has various kinds of benefits and uses in the daily life of the community and is easy to care so that it is widely cultivated. With the Telegram application that is used as a medium of communication between farmers and their gardens. And using Arduino Mega 2560 type microcontroller as sensor data processing farmers can monitor soil moisture, humidity, and temperature anytime and anywhere, and use the NodeMCU microcontroller as a connecting bridge to the internet network. It is expected that the monitoring system can facilitate the work of farmers in monitoring the watering of their gardens.

Keywords: Green Mustard, Monitoring, Time and Distance, Telegram, Microcontroller.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan laporan Proyek Akhir ini dengan baik. Adapun laporan Proyek Akhir ini disusun sebagai salah satu persyaratan dan kewajiban mahasiswa untuk menyelesaikan kurikulum program pendidikan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung.

Dalam Proyek Akhir ini penulis membuat sebuah sistem pengontrolan pada kebun sawi dengan menggunakan *Telegram*. Penulis mengakui bahwa selesainya Proyek Akhir ini tidak lepas dari bantuan banyak pihak yang telah membantu dan memberi dukungan dalam membuat alat maupun dalam menyelesaikan laporan Proyek Akhir ini. Untuk itu penulis mengucapkan terima kasih kepada :

1. Orang tua serta keluarga yang selalu memberikan kasih sayang, doa serta dukungan.
2. Bapak Sugeng Ariyono, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Aan Febriansyah, M.T, selaku Kepala Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Eko Sulisty, M.T, selaku Kepala Prodi D III Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung sekaligus pembimbing 1 dalam Proyek Akhir ini.
5. Bapak Riki Afriansyah, M.T, selaku pembimbing 2 dalam Proyek Akhir ini.
6. Bapak Dr. Parulian Silalahi, M.Pd selaku wali kelas 3 EB.
7. Seluruh staf pengajar dan karyawan di Politeknik Manufaktur Negeri Bangka Belitung.
8. Rekan-rekan mahasiswa tingkat akhir Politeknik Manufaktur Negeri Bangka Belitung.
9. Teman-teman yang telah ikut mendukung dan memberikan bantuan serta masukan dalam pembuatan Proyek Akhir ini.

10. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa dalam laporan Proyek Akhir ini masih jauh dari sempurna, untuk itu penulis sangat mengharapkan semua jenis saran, kritik dan masukan yang bersifat membangun dalam rangka perbaikan laporan ini. Demikian laporan ini dibuat dan semoga laporan ini dapat bermanfaat dan menambah wawasan bagi pembaca. Akhir kata penulis mengucapkan terima kasih.

Sungailiat, 16 Juli 2018

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah	1
1.2. Perumusan Masalah	2
1.2.1. Rumusan Masalah.....	2
1.2.2. Batasan Masalah	2
1.3. Tujuan Proyek Akhir.....	2
BAB II DASAR TEORI.....	3
2.1. Sawi hijau (<i>Brassica rapa L</i>)	3
2.1.1. Teknik Budidaya Sawi Hijau (<i>Brassica rapa L</i>	4
2.2. <i>Telegram</i>	6
2.3. Arduino Mega 2560	7
2.4. NodeMCU ESP8266 12E	9
2.5. LCD (<i>Liquid Crystal Display</i>)	10
2.6. YL-69 (<i>Soil Moisture Sensor</i>).....	11
2.7 DHT 11 (<i>Temperature and Humidity Sensor</i>)	12
2.8. Real Time Clock Module (DS3231)	14
2.9. <i>SD Card Module</i>	15

2.10.	<i>Relay Module 10A</i>	17
2.11.	<i>Ubec (Universal Battery Elimination Circuit)</i>	18
2.12.	Pompa DP-537.....	18
2.13.	<i>ThingSpeak</i>	19
BAB III METODE PELAKSANAAN		21
3.1.	Survei, Pengumpulan Data dan Pengolahan Data.....	22
3.2.	Perancangan Sistem <i>Monitoring</i> dan Kebun via Telegram	22
3.3.	Pembuatan Sistem <i>Monitoring</i> dan Kebun via Telegram	23
3.3.1.	Langkah – langkah Pembuatan Kontruksi Kebun.	23
3.3.2.	Langkah – langkah Pembuatan <i>Sistem Monitoring</i>	23
3.4.	Uji Coba Alat.....	23
3.4.1.	Uji Coba Sistem <i>Monitoring</i>	23
3.4.2.	Uji Coba Kontruksi Kebun	24
3.4.3.	Uji coba Keseluruhan.....	24
3.5.	Analisis Data.....	24
3.6.	Pembuatan Laporan	24
BAB IV PEMBAHASAN		25
4.1	Diagram Blok.....	25
4.2	Perancangan Sistem <i>Monitoring</i>	26
4.2.1	Perancangan Rangkain <i>Soil Moisture Sensor (YL-69)</i>	26
4.2.2	Perancangan Rangkaian <i>Temp and Humd Sensor (DHT11)</i>	27
4.2.3	Perancangan Rangkain LCD (<i>Liquid Cristal Display</i>).	27
4.2.4	Perancangan Rangkaian <i>Real Time Clock Module DS3231</i>	29
4.2.5	Perancangan Rangkaian <i>SD Card Module</i>	29
4.2.6	Perancangan Rangkaian NodeMCU ESP8286 12E.....	30
4.2.7	Perancangan Rangkaian Keseluruhan.....	31
4.3	Perancangan Rangkaian <i>Driver Pompa</i>	32
4.4	Perancangan Rangkaian Catu Daya	33
4.5	Perancangan Kontruksi Kebun	33
4.5.1	Perancangan Rangka Kebun	33
4.5.2	Perancangan Pemipaan Penyiraman Kebun.....	35

4.6	Pembuatan Sistem <i>Monitoring</i>	35
4.6.1	Pengaturan <i>Bot User Telegram</i>	35
4.6.2	Pembuatan Rangkaian <i>Soil Moisture Sensor</i> (YL-69).....	36
4.6.3	Pembuatan Rangkaian <i>Temp and Humd Sensor</i> (DHT11)	44
4.6.4	Pembuatan Rangkaian LCD (<i>Liquid Cristal Display</i>).....	47
4.6.5	Pembuatan Rangkaian <i>Real Time Clock</i> (DS3231)	48
4.6.6	Pembuatan Rangkaian <i>SD Card Module</i>	50
4.6.7	Perancangan Rangkaian NodeMCU ESP8266 12E.....	52
4.6.8	Perakitan Rangkaian Keseluruhan	53
4.7	Pembuatan Rangkaian <i>Catu Daya</i>	57
4.8	Pembuatan Rangkaian <i>Driver Pompa</i>	58
4.9	Pembuatan Kontruksi Kebun	59
4.9.1	Pembuatan Rangka Kebun	60
4.9.2	Pembuatan Pemipaan Penyiraman Kebun	61
	BAB V KESIMPULAN DAN SARAN	63
5.1	Kesimpulan	63
5.2	Saran	63
	DAFTAR PUSTAKA	64
	LAMPIRAN	

DAFTAR TABEL

Tabel	Halaman
2.1 Gizi Tanaman Sawi Hijau.....	6
2.2 Spesifikasi Arduino Mega 2560	8
2.3 Spesifikasi NodeMCU ESP8266 12E.....	9
2.4 Spesifikasi <i>Soil Moisture Sensor</i> YL-69	12
2.5 Konfigurasi Pin <i>Soil Moisture Sensor</i> YL-69.....	12
2.6 Spesifikasi DHT 11 (<i>Temperature and Humidity Sensor</i>)	13
2.7 Konfigurasi DHT 11 (<i>Temperature and Humidity Sensor</i>).....	14
2.8 Spesifikasi DS3231 <i>Real Time Clock (RTC) Module</i>	15
2.9 Konfigurasi Pin DS3231 <i>Real Time Clock (RTC) Module</i>	15
2.10 Spesifikasi <i>SD Card Module</i>	16
2.11 Konfigurasi Pin <i>SD Card Module</i>	16
2.12 Spesifikasi <i>Relay Module</i> 10A	17
2.13 Konfigurasi Pin <i>Relay Module</i> 10A.....	17
4.1 Hasil Pengukuran <i>Soil Moisture Sensor</i> 1	38
4.2 Hasil Pengukuran <i>Soil Moisture Sensor</i> 2	41
4.3 Hasil Pengukuran <i>Temperature</i> (DHT11)	45
4.4 Hasil Pengukuran <i>Humidity</i> (DHT11)	45
4.5 Pengujian Pencacah Waktu RTC.....	49
4.6 Pengujian Waktu Pengiriman Pesan.....	55
4.7 Pengujian Rangkain Catu Daya	57
4.8 Pengujian Rangkain <i>Driver</i> Pompa	59

DAFTAR GAMBAR

Gambar	Halaman
2.1 Sawi Hijau (<i>Brassica rapa</i>)	3
2.2 Pembenihan.....	4
2.3 Pengolahan Tanah.....	5
2.4 Penyiraman	5
2.5 <i>Telegram</i>	7
2.6 Arduino Mega 2560	8
2.7 NodeMCU ESP8266 12E	9
2.8 LCD (<i>Liquid Crystal Display</i>) 2x16	10
2.9 YL-69 (<i>Soil Moisture Sensor</i>)	11
2.10 DHT 11 (<i>Temperature and Humidity Sensor</i>)	13
2.11 DS3231 <i>Real Time Clock (RTC) Module</i>	14
2.12 <i>SD Card Module</i>	16
2.13 <i>Relay Module</i> 10A	17
2.14 UBEC 3A.....	18
2.15 Pompa DP-537.....	19
2.16 Logo <i>ThingSpeak</i>	19
2.17 <i>API Key</i> dan <i>Channel ID</i>	20
3.1 <i>Flow Chart</i> Proses Pengerjaan Proyek Akhir.....	21
3.2 Proses Kerja Sistem <i>Monitoring Kebun Via Telegram</i>	22
4.1 Diagram Blok <i>Monitoring Kebun Via Telegram</i>	25
4.2 Skematik <i>Soil Moisture Sensor</i> (YL-69)	26
4.3 Hubungan Pin <i>Soil Moisture Sensor</i> (YL-69).	26
4.4 Skematik <i>Temperature and Humidity Sensor</i> (DHT 11).....	27
4.5 Hubungan Pin <i>Temperature and Humidity Sensor</i> (DHT 11).....	27
4.6 Skematik LCD (<i>Liquid Crystal Display</i>).....	28
4.7 Hubungan Pin LCD (<i>Liquid Crystal Display</i>).....	28

4.8	Skematik RTC DS3231	29
4.9	Hubungan Pin Real Time Clock Module (DS3231).....	29
4.10	Skematik <i>SD Card Module</i>	30
4.11	Hubungan Pin <i>SD Card Module</i>	30
4.12	Skematik NodeMCU ESP8286 12E	31
4.13	Hubungan Pin NodeMCU ESP8266 12E.	31
4.14	Skematik Keseluruhan Rangkaian.....	32
4.15	Skematik <i>Driver Pompa</i>	32
4.16	Skematik Catu Daya	33
4.17	Tampak Samping Rangka Kebun	34
4.18	Tampak Atas Rangka Kebun	34
4.19	Tampak Depan Rangka Kebun	34
4.20	Pemipaan Penyiraman Kebun.....	35
4.21	Pesan <i>Token API</i>	36
4.22	Rangkaian <i>Soil Moisture Sensor (YL-69)</i>	36
4.23	Pembacaan <i>Soil Moisture Sensor (YL-69)</i>	37
4.24	<i>Mapping</i> Peletakan <i>Soil Moisture Sensor (YL-69)</i>	37
4.25	Pengukuran S1 Pertama.....	38
4.26	Pengukuran S2 Pertama.....	39
4.27	Pengukuran S3 Pertama.....	39
4.28	Pengukuran S4 Pertama.....	39
4.29	Pengukuran S5 Pertama.....	40
4.30	Pengukuran S6 Pertama.....	40
4.31	Pengukuran S1 Kedua.....	41
4.32	Pengukuran S2 Kedua.....	42
4.33	Pengukuran S3 Kedua.....	42
4.34	Pengukuran S4 Kedua.....	42
4.35	Pengukuran S5 Kedua.....	43
4.36	Pengukuran S6 Kedua.....	43
4.37	Peletakan Sensor DHT11	44
4.38	Pembacaan <i>Temperature and Humidity Sensor (DHT11)</i>	45

4.39	Pengujian Sensor DHT11 Pertama.....	46
4.40	Pengujian Sensor DHT11 kedua	46
4.41	Pengujian Sensor DHT11 Ketiga	46
4.42	Rangkaian LCD (<i>Liquid Crystal Display</i>)	47
4.43	Pengujian Tampilan LCD 16x2	48
4.44	Perangkain Real Time Clock DS3231	48
4.45	Tampilan Tanggal dan Waktu RTC.....	49
4.46	Pengujian RTC Pertama	49
4.47	Pengujian RTC Kedua	50
4.48	Pengujian RTC Ketiga.....	50
4.49	Perangkaian SD Card Module	50
4.50	Hasil Penyimpanan di SD Card Module	51
4.51	Perangkaian NodeMCU ESP8266 12E	52
4.52	Pengujian <i>Chatting Telegram</i> NodeMCU	53
4.53	Rangkaian Sistem <i>Monitoring</i>	53
4.54	Pengiriman Data Kelembaban Tanah Ke <i>ThingSpeak</i>	54
4.55	Pengiriman Data Suhu Ke <i>ThingSpeak</i>	54
4.56	Pengiriman Data kelembaban Udara Ke <i>ThingSpeak</i>	54
4.57	Perintah Perintah “ <i>Manual</i> ”	55
4.58	Perintah Perintah “ <i>Pump Off</i> ”	55
4.59	Perintah Perintah “ <i>Pump On</i> ”	56
4.60	Pengujian Perintah “ <i>Status</i> ”	56
4.61	Pengujian Perintah “ <i>Save Data</i> ”	56
4.62	Pengujian Perintah “ <i>Auto</i> ”	56
4.63	Rangkaian Catu Daya	57
4.64	Hasil Pengujian Tenggangan Catu Daya 5 V	58
4.65	Hasil Pengujian Tenggangan Catu Daya 12 V	58
4.66	Rangkaian <i>Driver</i> Pompa	59
4.67	Hasil Pembuatan Rangka Kebun	60
4.68	Pengujian Lebar kebun	60
4.69	Pengujian Panjang kebun.....	61

4.70 Pemipaan Penyiraman Kebun.....	61
4.71 Pengujian Sistem Penyiraman Kebun.....	62

DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup.

Lampiran 2 : Program Arduino Mega 2560 : Sistem *Monitoring* Kebun Sawi *Via Telegram*.

Lampiran 3 : Program NodeMCU ESP8266 12E : Sistem *Monitoring* Kebun Sawi *Via Telegram*.

Lampiran 4 : Spesifikasi Alat Ukur Suhu dan Kelembaban Udara : *Hygrometer* HTC – 2.

Lampiran 5 : Spesifikasi Alat Ukur Kelembaban Tanah : *Tester Soil Water 3 in 1*.

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Sawi Hijau (*Brassica rapa L*) merupakan salah satu komoditas tanaman *hortikultura* dari jenis sayur – sayuran yang dimanfaatkan daun – daun yang masih muda. Daun sawi sebagai makanan sayuran memiliki macam – macam manfaat dan kegunaan dalam kehidupan masyarakat sehari – hari. Hariyadi, Ali, dan Nurlina [1] menyatakan Sawi hijau biasanya dibudidayakan pada daerah yang mempunyai ketinggian 100 m sampai 500 m di atas permukaan laut dan tanah yang cocok untuk ditanami sawi adalah tanah gembur, banyak mengandung humus, subur, serta pembuangan airnya baik. Dan Anonim [2] menyatakan bahwa sawi hijau juga tahan terhadap air hujan, sehingga dapat ditanam sepanjang tahun. Pada musim kemarau yang perlu diperhatikan adalah penyiraman secara teratur. Akan tetapi terdapat berbagai permasalahan yang dihadapi para petani dalam merawat sawi hijaunya. Berdasarkan observasi langsung yang dilakukan di Kp. Pasir Kelurahan Matras, Sungailiat beberapa kendala yang terjadi yaitu penyiraman sawi didasari pada kondisi cuaca, apabila hujan pada malam hari maka penyiraman tidak dilakukan pada pagi harinya dan seterusnya, sehingga kelembaban tanah tergantung daripada curah hujan dan kelembaban tanah yang ideal belum tentu tercapai, serta metode penyiraman yang masih menggunakan tenaga manusia terkesan kurang efisien dari segi waktu, dan tenaga.

Di era modern sekarang ini, perkembangan teknologi sangat cepat, salah satu aplikasi dibidang teknologi yang dapat diterapkan untuk membantu permasalahan para petani sawi yang ada ialah dengan membangun sebuah sistem *monitoring* kelembaban dan penyiraman kebun sawi dengan memanfaatkan aplikasi *Telegram*. Sehingga para petani sawi dapat kapan saja memantau kelembaban tanah, kelembaban udara, serta suhu lingkungan dan penyiraman kebun sawi yang dimilikinya tanpa batasan jarak, waktu, dan tenaga, melalui

konektifitas internet yang dapat berkomunikasi secara dua arah antara *user* (petani) dan *object* (kebun sawi).

1.2. Perumusan Masalah

1.2.1. Rumusan Masalah

Adapun rumusan masalah yang diangkat berdasarkan latar belakang Proyek Akhir ini adalah sebagai berikut :

1. Bagaimana merancang sebuah sistem menggunakan aplikasi *Telegram* melalui *Smartphone* yang dapat memantau kelembaban tanah, kelembaban udara, serta suhu lingkungan dan penyiraman kebun sawi yang dimiliki?

1.2.2. Batasan Masalah

Adapun batasan masalah dari Proyek Akhir ini adalah sebagai berikut :

1. Ukuran dari kebun sawi dibatasi dengan panjang = 2 m dan lebar = 1 m.
2. *Monitoring* dibatasi pada kelembaban tanah, kelembaban udara, serta suhu lingkungan dan penyiraman kebun sawi.

1.3. Tujuan Proyek Akhir

Tujuan yang ingin dicapai pada pembuatan Proyek Akhir ini adalah sebagai berikut :

1. Mempermudah pekerjaan petani dalam memonitor kelembaban tanah, kelembaban udara, serta suhu lingkungan dan penyiraman kebun sawi yang dimilikinya.
2. Meningkatkan efisiensi pekerjaan dari segi waktu dan tenaga.

BAB II

DASAR TEORI

2.1. Sawi Hijau (*Brassica rapa L*)

Sawi hijau (*Brassica rapa L*) merupakan salah satu jenis sayuran yang banyak digemari oleh masyarakat Indonesia, selain karena mudah didapatkan, sawi hijau juga memiliki banyak sekali khasiat serta manfaatnya seperti pengobatan berbagai macam penyakit sehingga sawi hijau termasuk golongan sayuran yang mempunyai peran penting untuk memenuhi kebutuhan pangan, gizi, dan obat bagi masyarakat[2]. Berikut sawi hijau ditunjukkan pada gambar pada gambar 2.1 :



Gambar 2.1 Sawi Hijau (*Brassica rapa L*)

Tanaman sawi memiliki perawatan yang tidak begitu sulit dan pertumbuhan tanaman cepat, sehingga budidaya tanaman sayuran seperti sawi hijau ini sering diterapkan oleh para petani untuk mendapatkan hasil yang cepat. Pada perawatan tanaman sawi hijau hal yang biasa dilakukan adalah penyiangan tanaman, pemupukan, dan penyemprotan yang dilakukan secara teratur untuk mencegah adanya hama atau penyakit yang tidak diinginkan. Syarat – syarat penting untuk bertanam sawi ialah tanahnya gembur, banyak mengandung humus (subur), dan keadaan pembuangan airnya (*drainase*) baik. Derajat keasamaan tanah (pH) antara 6 – 7[1].

2.1.1. Teknik Budidaya Sawi Hijau (*Brassica rapa L*)

- **Pembenihan**

Satu faktor penentu keberhasilan budidaya sawi hijau adalah faktor pembenihan, karena benih yang baik dapat menghasilkan tanaman yang memiliki pertumbuhan bagus. Pada umumnya benih sawi yang baik memiliki bentuk bulat, kecil, warna kulit coklat kehitaman, agak keras, dan permukaannya licin mengkilap. Benih sawi yang akan digunakan untuk bercocok tanam harus memiliki kualitas yang baik. Jika benih tersebut membeli di toko, maka saat membeli harus diperhatikan lamanya penyimpanan, kadar air, varietas, suhu dan tempat untuk menyimpan[1]. Berikut contoh benih yang digunakan ditunjukkan pada gambar 2.2 :



Gambar 2.2 Pembenihan

- **Pengolahan Tanah**

Pengolahan tanah secara umum proses pengolahan tanah untuk budidaya sawi hijau yang dimaksud adalah melakukan penggemburan tanah di gambus ukuran ± 20 cm. Pemberian pupuk dasar pada tanah yang tujuannya untuk menambah kesuburan pada tanaman. Maka diberi pupuk organik, seperti pupuk kandang atau kompos jerami diberikan saat berlangsungnya penggemburan tanah. Agar pupuk organik tersebut dapat cepat merata dan bercampur dengan tanah yang akan digunakan, maka digunakan perbandingan antara pupuk organik dan tanah sebesar 2 : 2[1].

Berikut contoh pengolahan tanah yang ditunjukkan pada gambar 2.3 :



Gambar 2.3 Pengolahan Tanah

- **Penyiraman**

Pertama yang harus diperhatikan dalam perawatan adalah penyiraman. Penyiraman tergantung pada musim. Jika musim penghujan datang dan curah hujan berlebihan, maka pengurangan air harus dilakukan. Tetapi jika sebaliknya, yakni jika air kurang karena datangnya musim kemarau, maka harus dilakukan penambahan air, agar kecukupan bagi tanaman sawi. Berikut contoh penyiraman ditunjukkan pada gambar 2.4 :



Gambar 2.4 Penyiraman

Berikut merupakan syarat pertumbuhan tanaman sawi menurut survei sekunder yang telah dikumpulkan :

1. Pertumbuhan optimal pada ketinggian 5 – 1200 mdpl[1].
2. Suhu pertumbuhan optimal antara 27 °C – 32 °C[3].

3. Kelembaban udara yang ideal antara 80 %RH – 90 %RH[3].
4. Kelembaban tanah yang ideal antara 60 % - 87 % [3].

- **Pemupukan**

Pemupukan membantu tanaman memperoleh hara yang dibutuhkannya. Unsur hara yang pokok dibutuhkan tanaman adalah unsur *Nitrogen* (N), *Fosfor* (P), dan *Kalium* (K). Itulah sebabnya ketiga unsur ini (NPK) merupakan pupuk utama yang dibutuhkan oleh tanaman. Pupuk organik juga dibutuhkan oleh tanaman, memang kandungan harany jauh dibawah pupuk kimia, tetapi pupuk organik memiliki kelebihan membantu menggemburkan tanah dan menyatu secara alami, menambahkan unsur hara dan memperbaiki tanah[2]. Adapun tabel kandungan gizi sawi hijau ditunjukkan pada tabel 2.1 berikut :

Tabel 2.1 Gizi Tanaman Sawi Hijau[1]

Kalori	63
Gram Protein	5
Gram Lemak	1
Gram Karbohidrat	11
Dari Kebutuhan Harian Untuk Vitamin A	25%
Dari Kebutuhan Harian Untuk Vitamin C	5%

2.2. *Telegram*

Telegram adalah sebuah aplikasi layanan pengirim pesan instan *multiplatform* berbasis *cloud* yang gratis dan nirkabel. *Telegram* juga memungkinkan para pengguna untuk mengirimkan pesan secara rahasia yang dienkripsi sebagai tambahan keamanan. *Telegram* juga bersifat *open source*, yang berarti para pengguna secara umum dapat membuat fitur – fitur tambahan yang dapat disertakan pada aplikasi ini, Salah satu fitur yang dimiliki ialah tersedianya fitur *bot chatting* yang dapat diatur dan dikomunikasikan dengan mikrokontroller melalui konektifitas internet yang tersedia, sehingga dapat diaplikasikan pada

berbagai macam kegunaan dan memiliki cakupan yang sangat luas[4]. Berikut logo *Telegram* ditunjukkan gambar 2.5 :



Gambar 2.5 Logo *Telegram*[4]

Berikut merupakan langkah – langkah penggunaan aplikasi *Telegram* :

- *Install* aplikasi *Telegram*. Buka *App store (IOS)* ata *Google play / Playstore (Android)* dan cari *Telegram* sesuai dengan logo diatas.
- Setelah di *Install*, buka aplikasinya ikuti langkah – langkahnya, masukan nomor telepon, lalu telegram akan mengirimkan pesan teks seperti kode,lalu masukan kode tersebut ke aplikasi telegram untuk melanjutkan
- Tambahkan nama dan foto untuk keterangan dari profil anda sendiri lalu klik *next*.
- *Telegram* secara otomatis dapat menemukan kontak teman – teman yang menggunakan *Telegram* juga, jika kita memberikan izin aplikasi ini untuk mengakses kontak di *smarphone*.
- *Telegram* sudah bisa digunakan untuk *chat* atau kirim foto maupun *file* apapun.

2.3. **Arduino Mega 2560**

Arduino Mega 2560 adalah papan Mikrokontroler berdasarkan Atmega 2560. Mikrokontroler itu sendiri adalah *chip* atau IC (*integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada Mikrokontroler bertugas sebagai otak yang mengendalikan *input*, proses dan

output sebuah rangkaian elektronik. Berikut Arduino Mega 2560 ditunjukkan pada gambar 2.6 :



Gambar 2.6 Arduino Mega 2560[5]

- **Spesifikasi Arduino Mega 2560**

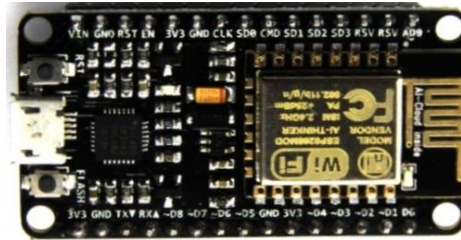
Spesifikasi dari Arduino Mega 2560 dapat dilihat pada tabel 2.2 berikut :

Tabel 2.2 Spesifikasi Arduino Mega 2560[5]

No.	Spesifikasi	Keterangan
1.	Mikrokontroler	Atmega 2560
2.	Tegangan Operasi	5V
3.	<i>Input Voltage</i> (disarankan)	7-12V
4.	<i>Input Voltage</i> (batas)	6-20V
5.	Digital I/O Pin	54 (15 <i>output</i> PWM)
6.	Pin Masukan Analog	16
7.	DC <i>Current</i> per I/O Pin	20mA
8.	DC saat ini untuk 3.3V Pin	50mA
9.	<i>Flash Memory</i>	256 KB , 8 KB digunakan oleh <i>bootloader</i>
10.	SRAM	8 KB
11.	EEPROM	4 KB
12.	Kecepatan <i>Clock</i>	16 Mhz
13.	LED_BUILTIN	13
14.	Panjang	10,52 mm
15.	Lebar	53,3 mm
16.	Berat	37

2.4. NodeMCU ESP8266 12E

NodeMCU adalah sebuah platform *IoT* yang bersifat opensource. Terdiri dari perangkat keras berupa *System On Chip* ESP8266 dari ESP8266 buatan Espressif System. Berikut NodeMCU ESP8266 12E ditunjukkan pada gambar 2.7 :



Gambar 2.7 NodeMCU ESP8266 12E[6]

NodeMCU bisa dianalogikan sebagai board arduino yang terkoneksi dengan ESP8622. NodeMCU telah *me-package* ESP8266 ke dalam sebuah board yang sudah terintegrasi dengan berbagai *feature* selayaknya *microcontroller*. Maka fitur – fitur yang dimiliki oleh NodeMCU akan lebih kurang serupa dengan ESP-12E.

- **Spesifikasi NodeMCU ESP8266 12E**

Spesifikasi dari NodeMCU ESP8266 12E dapat dilihat pada tabel 2.3 berikut :

Tabel 2.3 Spesifikasi NodeMCU ESP8266 12E[6]

No.	Spesifikasi	Keterangan
2.	Tegangan Operasi	3.3V
3.	<i>GPIO</i>	17 (multiplexing dengan fungsi lain)
4.	Arus	10uA ~ 170mA
5.	<i>Wifi Direct (Peer to Peer)</i>	<i>Soft-aP</i>
7.	<i>Processor</i>	<i>Tensilica L106 32-bit</i>
8.	<i>Processor Speed</i>	80~160 MHz
10.	802.11 <i>support</i>	b/g/n
12.	<i>Flash Memory</i>	16MB <i>max</i> (512K normal)

2.5. LCD (*Liquid Crystal Display*)

LCD (*Liquid Crystal Display*) adalah salah satu jenis *display* elektronik yang dibuat dengan teknologi *CMOS logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. LCD berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik. Dalam modul LCD terdapat Mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD. Mikrokontroler pada suatu LCD dilengkapi dengan memori dan register. Berikut LCD 2x16 ditunjukkan pada gambar 2.8 :



Gambar 2.8 LCD (*Liquid Crystal Display*) 2x16[7]

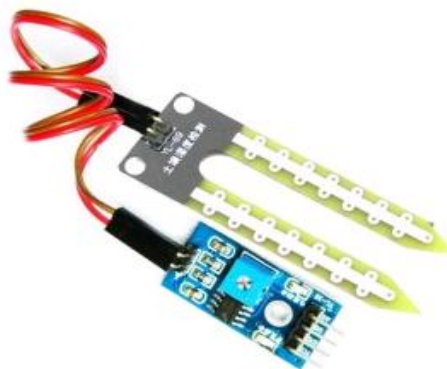
Fungsi dari setiap kaki yang ada pada LCD 2x16 adalah[7]:

1. Kaki 1 (GND): kaki ini berhubungan dengan tegangan 0 volt (*Ground*).
2. Kaki 2 (VCC): kaki ini berhubungan dengan tegangan +5 Volt.
3. Kaki 3 (VEE/VLCD): tegangan pengatur kontras LCD, kaki ini terhubung pada *cermet*. Kontras mencapai nilai maksimum pada saat kondisi kaki ini pada tegangan 0 volt.
4. Kaki 4 (RS): *Register Select*, kaki pemilih *register* yang akan diakses. Untuk akses ke *register* data, logika dari kaki ini adalah 1 dan untuk akses ke *register* perintah, logika dari kaki ini adalah 0.
5. Kaki 5 (R/W): logika 1 pada kaki ini menunjukkan bahwa modul LCD sedang pada mode pembacaan dan logika 0 menunjukkan bahwa modul LCD sedang pada mode penulisan. Untuk aplikasi yang tidak memerlukan pembacaan data pada modul LCD, kaki ini dapat dihubungkan langsung ke *Ground*

6. Kaki 6 (E): *Enable Clock LCD*, kaki mengaktifkan clock LCD. Logika 1 pada kaki ini diberikan pada saat penulisan atau membacakan data.
7. Kaki 7–14 (D0–D7): data bus, kedelapan kaki LCD ini adalah bagian di mana aliran data sebanyak 4 bit ataupun 8 bit mengalir saat proses penulisan maupun pembacaan data.
8. Kaki 15 (Anoda): berfungsi untuk tegangan positif dari *backlight* LCD sekitar 4,5 volt (hanya terdapat untuk LCD yang memiliki *backlight*)
9. Kaki 16 (Katoda): tegangan negatif *backlight* LCD sebesar 0 volt (hanya terdapat pada LCD yang memiliki *backlight*).

2.6. YL-69 (*Soil Moisture Sensor*)

Soil moisture sensor merupakan sensor untuk mengukur tingkat air dalam tanah atau kelembaban tanah, yang mempunyai dua *output* yaitu, keluaran analog yang dikonversikan kedalam bentuk persen, dan keluaran digital (*High / Low*) yang dapat diatur sensitivitas keluarannya dengan memutar *knob* pada *trimpot*. Penggunaan YL-69 (*Soil Moisture Sensor*) dengan cara menancapkan *probe* sensor pada suatu bidang tanah atau yang dapat mewakili suatu daerah yang ingin kita ukur. Dan pada Proyek Akhir ini menggunakan keluaran analog yang dikonversikan kedalam bentuk persen. Berikut YL-69 (*Soil Moisture Sensor*) ditunjukkan pada gambar 2.9 :



Gambar 2.9 YL-69 (*Soil Moisture Sensor*) [8]

- **Spesifikasi Soil Moisture Sensor YL-69**

Spesifikasi *soil moisture sensor* YL-69 dapat dilihat pada tabel 2.4 berikut :

Tabel 2.4 Spesifikasi *soil moisture sensor* YL-69[8]

No.	Spesifikasi	Keterangan
1.	Tegangan Operasi	3.3 – 5V
2.	Tegangan Keluaran	0 – 4.2V
3.	Arus	35 mA
4.	Indikator	Led
5.	Dimensi	60 x 20 x 5 mm

- **Konfigurasi Pin Soil Moisture Sensor YL-69**

Konfigurasi pin *soil moisture sensor* YL-69 dapat dilihat pada tabel 2.5 berikut :

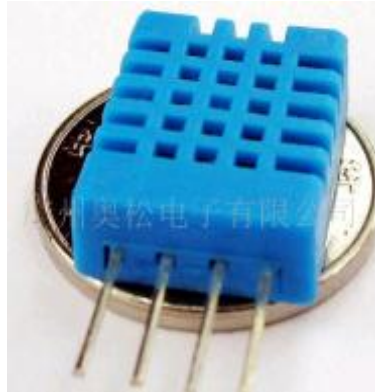
Tabel 2.5 Konfigurasi pin *soil moisture sensor* YL-69

No.	Pin	Keterangan
1.	VCC	Tegangan 3.3 – 5V.
2.	GND	Tegangan 0V (<i>Ground</i>).
3.	A0	Mengeluarkan nilai analog 0 – 1023
4.	D0	Mengeluarkan nilai digital (<i>HIGH/LOW</i>) pada kelembapan tertentu yang dapat diatur sensitivitasnya dengan knob pengatur sensitivitas.

2.7. DHT 11 (*Temperature and Humidity Sensor*)

DHT 11 (*Temperature and Humidity Sensor*) merupakan sensor yang berfungsi untuk mengukur suhu dan kelembaban udara pada daerah tersebut. DHT 11 (*Temperature and Humidity Sensor*) yang memiliki keluaran digital yang dikirimkan ke mikrokontroler dan langsung dikonversikan menggunakan *library* DHT 11 yang telah tersedia, kedalam satuan *Celcius* untuk suhu dan *Relative*

Humidity (RH) untuk kelembaban udara. Berikut DHT 11 ditunjukkan pada gambar 2.10 :



Gambar 2.10 DHT 11 (*Temperature and Humidity Sensor*)[9]

- **Spesifikasi DHT 11 (*Temperature and Humidity Sensor*)**

Spesifikasi DHT 11 (*Temperature and Humidity Sensor*) dapat dilihat pada tabel 2.6 berikut :

Tabel 2.6 Spesifikasi DHT 11 (*Temperature and Humidity Sensor*)[9]

No.	Spesifikasi	Keterangan
1.	Tegangan Operasi	3.3 – 5V
2.	Tegangan Keluaran	Digital signal via <i>single-bus</i>
3.	Elemen Sensor	<i>Polymer Resistor</i>
4.	Batas Pengukuran	Kelembapan : 20 – 95 % RH Suhu : 0 – 50 °Celcius
5.	Akurasi	Kelembapan : ± 4% RH (Max ± 5% RH) Suhu : ± 2.0 °Celcius
6.	Resolusi / Sensitivitas	Kelembapan 1% RH Suhu ± 0.1 °Celcius
7.	<i>Humidity Hysteresis</i>	± 1% RH
8.	<i>Long-term stability</i>	± 0.5 % RH / year
9.	<i>Sensing period</i>	2S
10.	Dimensi	12 x 15.5 x 5.5 mm

- **Konfigurasi Pin DHT 11 (*Temperature and Humidity Sensor*)**

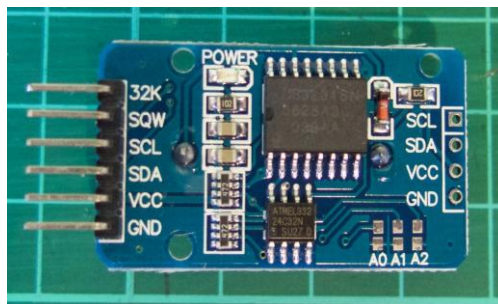
Konfigurasi pin DHT 11 (*Temperature and Humidity Sensor*) dapat dilihat pada tabel 2.7 berikut :

Tabel 2.7 Konfigurasi pin DHT 11 (*Temperature and Humidity Sensor*)

No.	Pin	Keterangan
1.	VCC	Tegangan 3.3 – 5V.
2.	GND	Tegangan 0V (<i>Ground</i>).
3.	NC	<i>Not Connect.</i>
4.	Data	<i>Digital signal via single-bus.</i>

2.8. *Real Time Clock Module (DS3231)*

Real Time Clock merupakan sebuah *module* yang berfungsi untuk mencacah waktu, khususnya jam, menit, detik, bahkan tanggal, bulan, tahun dan dapat membaca temperatur disekitar dan menjaga atau menyimpan secara *real time*. Maka setelah proses hitung waktu dilakukan data langsung disimpan atau dikirimkan menggunakan komunikasi I2C sebagai jalur komunikasi ke Arduino Mega 2560. RTC dilengkapi dengan baterai sebagai pensuplai daya pada *chip* sehingga waktu akan tetap *up to date* dan sering dijumpai pada *motherboard* PC yang biasa terletak dekat *chip* BIOS. Ada berbagai macam jenis – jenis *Real Time Clock*, salah satunya DS3231 yang mempunyai akurasi pencacahan yang baik. Berikut DS3231 *Real Time Clock* ditunjukkan pada gambar 2.11 berikut :



Gambar 2.11 *Real Time Clock Module (DS3231)*[10]

- **Spesifikasi *Real Time Clock Module* (DS3231)**

Spesifikasi *Real Time Clock Module* (DS3231) *Module* dapat dilihat pada tabel 2.8 berikut :

Tabel 2.8 Spesifikasi *Real Time Clock Module* (DS3231) [11]

No.	Spesifikasi	Keterangan
1.	Tegangan Operasi	3.3 – 5V
2.	<i>Accuracy</i>	$\pm 2 \text{ minutes / year from } -40^{\circ}\text{C to } +85^{\circ}\text{C}$
3.	<i>Memory Chip</i>	AT24C32 (<i>storage capacity</i> 32K)
4.	<i>Backup Battery</i>	<i>Internal Timekeeping</i>
6.	<i>Frequency</i>	32KHZ pin

- **Konfigurasi Pin *Real Time Clock Module* (DS3231)**

Konfigurasi pin *Real Time Clock Module* (DS3231) dapat dilihat pada tabel 2.9 berikut :

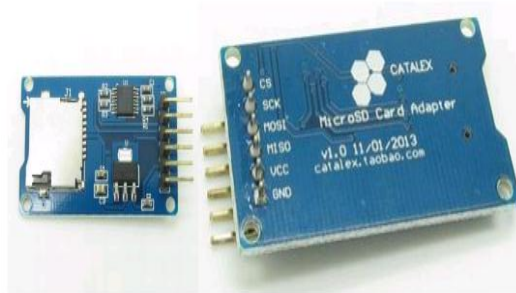
Tabel 2.9 Konfigurasi pin *Real Time Clock Module* (DS3231)

No.	Pin	Keterangan
1.	VCC	Tegangan 3.3 – 5V.
2.	GND	Tegangan 0V (<i>Ground</i>).
3.	SDA	Berhubungan dengan pin SDA master.
4.	SCL	Berhubungan dengan pin SCL master.
5.	SQW	<i>Not Connect.</i>
6.	32K	<i>Not Connect.</i>

2.9. SD Card Module

SD Card Module merupakan sebuah adapter untuk menyimpan data dari mikrokontroler ke dalam sebuah *memory card* yang *erasable* dan *renewable* data penyimpanannya, dan modul ini menggunakan komunikasi SPI sebagai jalur *interface* mikrokontroler. Dan modul ini bersifat *non – volatile* yang berarti data

akan tetap tersimpan walaupun tidak mendapat suplai tegangan. Berikut *SD Card Module* ditunjukkan pada gambar 2.12 berikut :



Gambar 2.12 *SD Card Module*[12]

- **Spesifikasi *SD Card Module***

Spesifikasi *SD Card Module* dapat dilihat pada tabel 2.10 berikut :

Tabel 2.10 Spesifikasi *SD Card Module*[12]

No.	Spesifikasi	Keterangan
1.	Tegangan Operasi	4.5 – 5.5V.
2.	<i>Level Conversion Circuit</i>	<i>LDO regulator output 3.3V for level conversion chip, Micro SD card supply</i>
3.	Komunikasi	<i>SPI Interface.</i>
4.	<i>Support For</i>	<i>SD Card, SDHC Card.</i>
5.	<i>Micro SD Card Connector</i>	<i>Self Bomb Deck.</i>

- **Konfigurasi Pin *SD Card Module***

Konfigurasi pin *SD Card Module* dapat dilihat pada tabel 2.11 berikut :

Tabel 2.11 Konfigurasi pin *SD Card Module*

No.	Pin	Keterangan
1.	VCC	Tegangan 4.5 – 5.5V.
2.	GND	Tegangan 0V (<i>Ground</i>).
3.	<i>SPI Bus</i>	<i>MOSI, MISO, SCK, SS.</i>

2.10. Relay Module 10A

Relay module merupakan saklar elektrik yang dapat dikontrol oleh mikrokontroler dengan menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil dapat menghantarkan listrik yang bertegangan yang lebih tinggi. Berikut *Relay Module 10A* ditunjukkan pada gambar 2.13 :



Gambar 2.13 *Relay Module 10A*[13]

- **Spesifikasi *Relay Module 10A***

Spesifikasi *SD Card Module* dapat dilihat pada tabel 2.12 berikut ini :

Tabel 2.12 Spesifikasi *Relay Module 10A*[13]

No.	Spesifikasi	Keterangan
1.	Dimensi	45.8mm x 32.4mm.
2.	<i>Supply Coils</i>	5V.
3.	<i>Switching Capability</i>	10A / 250VAC, 10A / 30VDC
4.	Komunikasi	<i>MCU I/O Link</i>

- **Konfigurasi Pin *Relay Module 10A***

Konfigurasi pin *Relay Module 10A* dapat dilihat pada tabel 2.13 berikut :

Tabel 2.13 Konfigurasi pin *Relay Module 10A*

No.	Pin	Keterangan
1.	VCC	Tegangan 5V.
2.	GND	Tegangan 0V (<i>Ground</i>).
3.	IN1	<i>High / Low Output from MCU</i>
4.	IN2	<i>High / Low Output from MCU</i>

2.11. UBEC (*Universal Battery Elimination Circuit*)

UBEC merupakan *regulator switching* yang bekerja pada tingkat tegangan 5V atau 6V dengan mengambil daya dari *battery pack* atau sumber DC lainnya, dan menurunkannya ke level tegangan 5V atau 6V. Adapun UBEC yang digunakan ditunjukkan pada gambar 2.14. Berikut adalah deskripsi serta spesifikasi dari jenis UBEC yang digunakan[14] :

Nama Merek : Hobbywing

Nama Item : 3A-UBEC

Tegangan Output : 5V 3A atau 6V 3A (dipilih menggunakan konektor)

Arus keluaran : 3 A

Masukan : 5.5V-26V (2-6S Lipo atau 5-18 sel NiMH/NiCd)

Ukuran : 51mm x 16.6mm x 8.5mm

Berat : 11.5g

Fitur :

1. Menggunakan mode saklar DC-DC *switch* lanjutan.
2. Outputnya cukup kuat meski bekerja dengan baterai Lipo 4S sampai 6S.
3. Perlindungan pembalikan polaritas baterai.
4. Perisai logam dan saringan secara signifikan mengurangi interferensi elektromagnetik.
5. Status kerja ditunjukkan oleh LED, lampu menyala saat UBEC bekerja normal.



Gambar 2.14 UBEC 3A[14]

2.12. Pompa DP-537

Pompa Air DC digunakan untuk melakukan penyiraman secara elektrik dengan *suplly* 12V yang dapat dikontrol secara mudah. Adapun pompa yang

digunakan ditunjukkan pada gambar 2.15. Berikut merupakan spesifikasi dari pompa air yang digunakan[15] :

Model : DP-537
Dimension : 170 x 100 x 67 mm
Daya / Arus : 60 – 65 Watt (5A)
Voltage : DC 12V (maks 15 Volt).
Max Pressure : 100 Psi.
Max Flow : 4.0L / Min.



Gambar 2.15 Pompa DP-537[15]

2.13. *ThingSpeak*

ThingSpeak merupakan adalah layanan *web* gratis yang memungkinkan kita mengumpulkan dan menyimpan data sensor di *cloud* dan mengembangkan aplikasi *Internet of Things*. Layanan *ThingSpeak* menyediakan aplikasi yang memungkinkan kita menganalisis dan memvisualisasikan data di MATLAB, dan kemudian bertindak berdasarkan data. Data sensor dapat dikirim ke *ThingSpeak* dari Arduino®, Raspberry Pi™, BeagleBone Black, dan perangkat keras lainnya[16]. Berikut logo *Thingspeak* ditunjukkan pada gambar 2.16 :



Gambar 2.16 Logo *ThingSpeak*[16]

- **Pengaturan Saluran *ThingSpeak***

ThingSpeak membutuhkan akun pengguna dan saluran. Saluran adalah tempat mengirim data dan tempat menyimpan data. Setiap saluran memiliki 8 bidang data, bidang lokasi, dan bidang status. Kita dapat mengirim data setiap 15 detik ke *ThingSpeak*, tetapi sebagian besar aplikasi bekerja dengan baik setiap menit. Hal yang dilakukan untuk memulai pengiriman data ke *ThingSpeak* ialah[16] :

1. Mendaftar akun pengguna baru *ThingSpeak* di :
https://thingspeak.com/users/sign_up
2. Buat *Channel* baru dengan pilih *Channels*, *My Channels*, and then *New Channel*.
3. Simpan dan catat *Write API Key* dan *Channel ID*. Berikut *API Key* dan *Channel ID*.

```
// replace with your channel's thingspeak API key
unsigned long myChannelNumber = 412985;
const char * myWriteAPIKey = "66TKCZMVFMWZ70**";
```

Gambar 2.17 *API Key* dan *Channel ID*

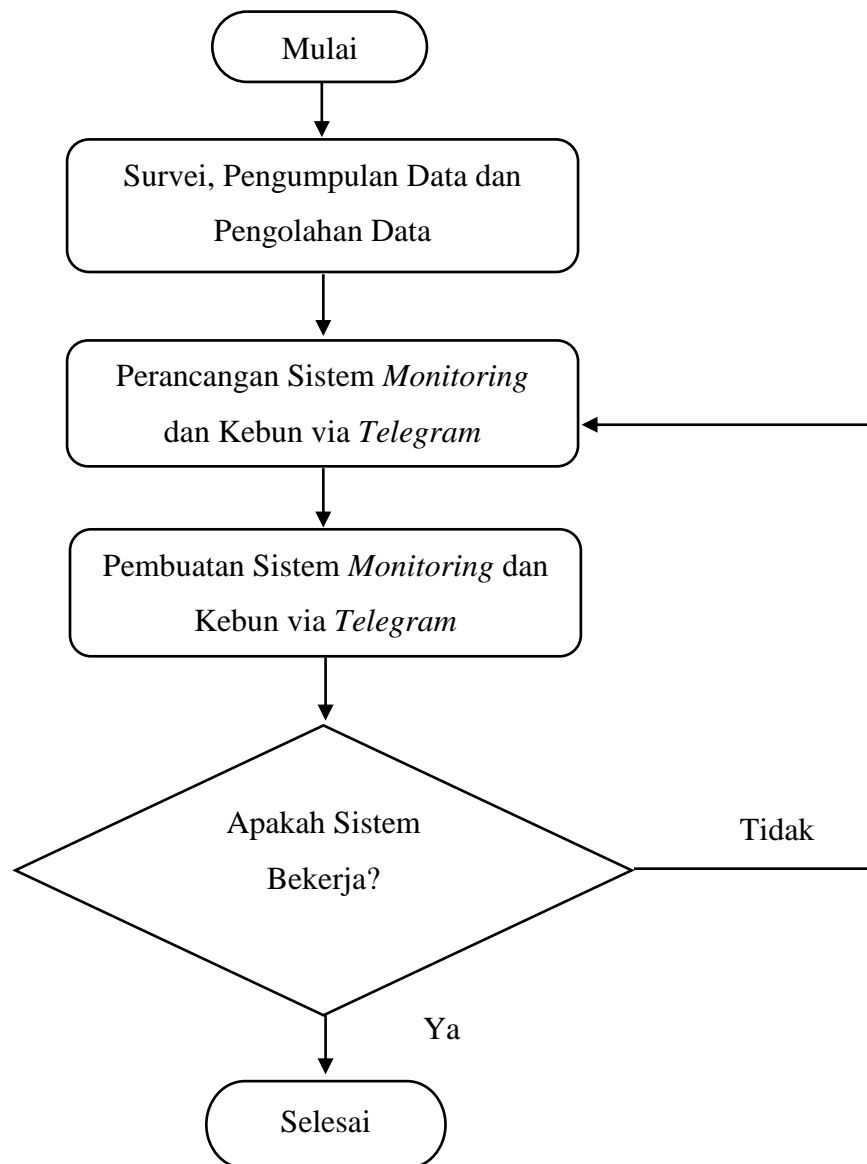
- **Pengiriman Data Ke *ThingSpeak***

Setelah pengaturan saluran dan pemasangan *library ThingSpeak* selesai, berikutnya data sudah bisa dikirim ke *cloud ThingSpeak*. Berikut merupakan langkah pengiriman data ke saluran *ThingSpeak*[17] :

1. Buka *sketch example ThingSpeak "WriteVoltage.ino"* .
2. Masukkan *Write API Keys* dan *Channel ID* kita ke *Sketch Arduino*.
3. Pastikan jenis *board Arduino* yang dipakai dan *port* yang digunakan. Lalu *upload sketch* tadi ke *Arduino*, dan lihat hasilnya di saluran yang kita buat.

BAB III METODE PELAKSANAAN

Dalam pembuatan Proyek Akhir ini dilakukan dalam beberapa tahap proses pengerjaan yang bertujuan untuk mempermudah dalam pencapaian target penyelesaian. Berikut Flow Chart *Proses Pengerjaan Proyek Akhir* ditunjukkan pada gambar 3.1 :



Gambar 3.1 *Flow Chart* Proses Pengerjaan Proyek Akhir

Berdasarkan gambar *Flow Chart* diatas, proses pengerjaan Proyek Akhir dilakukan melalui beberapa metode dalam penyelesaian pembuatan Proyek Akhir ini yaitu sebagai berikut :

3.1. Survei, Pengumpulan Data dan Pengolahan Data

Langkah pertama yang dilakukan sebelum memulai Proyek Akhir ialah, melakukan survei dan pengumpulan data yang sesuai dengan judul proyek akhir. Dimana, survei dan pengumpulan merujuk kepada survei primer dan sekunder yang dilakukan.

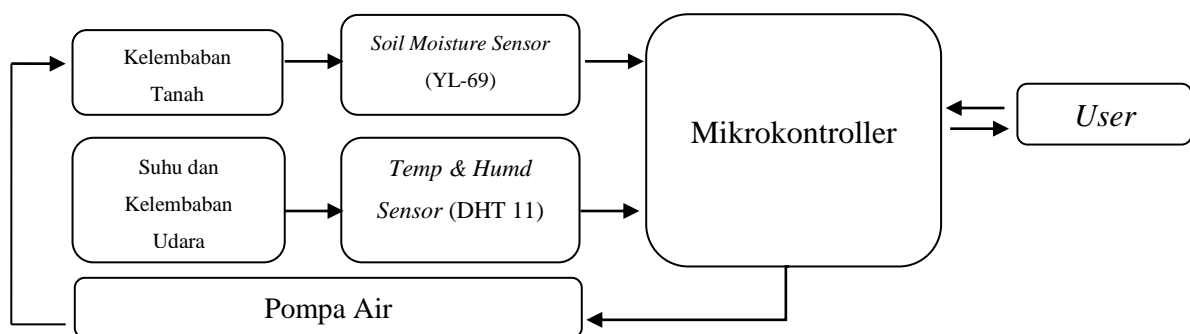
Berikut merupakan beberapa langkah survei primer yang dilakukan untuk pengumpulan data :

1. Konsultasi ke dosen pembimbing.
2. Observasi ke kebun sawi langsung.

Dan survei sekunder yang dilakukan merujuk kepada beberapa jurnal, paper dan website yang terdapat pada daftar pustaka.

3.2. Perancangan Sistem *Monitoring* dan Kebun via Telegram

Setelah melakukan pengumpulan dan pengolahan data, langkah selanjutnya, melakukan perancangan sistem *monitoring* dan konsep kebun. Dimana, pada tahap ini dibuat blok diagram kerja daripada sistem *monitoring* yang akan dilakukan, yang nantinya disesuaikan dengan kebutuhan daripada konsep kebun dan jenis tanaman yang dipilih sebagai *object* uji coba. Berikut Proses Kerja Sistem *Monitoring* Kebun Via Telegram ditunjukkan pada gambar 3.2 :



Gambar 3.2 Proses Kerja Sistem *Monitoring* Kebun Via Telegram

3.3. Pembuatan Sistem *Monitoring* dan Kebun via *Telegram*

Setelah selesai melakukan perancangan sistem dan konsep kebun yang akan dibuat, selanjutnya masuk dalam tahap pengerjaan dan pembuatan sistem dan konsep kebun yang disesuaikan dengan blok diagram kerja, yang dimana meliputi pembuatan konstruksi kebun seperti pemilihan bahan, penentuan ukuran dan perakitan komponen – komponen dan *coding* program daripada sistem *monitoring* sendiri.

3.3.1. Langkah – langkah Pembuatan Konstruksi Kebun

Berikut langkah – langkah pembuatan konstruksi kebun :

1. Pembuatan rangka kebun dengan ukuran 2 x 1 meter.
2. Pembuatan pemipaan penyiraman kebun.
3. Pemilihan media tanam yang akan digunakan.
4. Pemilihan dan penyemaian bibit sawi yang akan ditanam.

3.3.2. Langkah – langkah Pembuatan Sistem *Monitoring*

Berikut merupakan langkah – langkah pembuatan sistem *monitoring* :

1. Pembuatan rangkaian sistem monitoring kebun via *Telegram*.
2. Pembuatan rangkaian catu daya.
3. Pembuatan rangkaian *driver* pompa.
4. Pemasangan sensor.

3.4. Uji Coba Alat

Uji coba dilakukan pada masing – masing bagian sistem *monitoring* dan konstruksi kebun. Uji coba pada bagian ini dilakukan untuk memastikan apakah sistem *monitoring* atau konstruksi kebun yang dibuat sudah bekerja dengan optimal serta sesuai dengan fungsi yang diinginkan.

3.4.1. Uji Coba Sistem *Monitoring*

Uji coba uji coba sistem *monitoring* yang dilakukan antara lain :

1. Uji coba rangkaian sistem monitoring kebun via *Telegram*..
2. Uji coba rangkaian catu daya.

3. Uji coba rangkaian *driver* pompa.
4. Uji coba sensor.

3.4.2 Uji Coba Konstruksi Kebun

Uji coba konstruksi kebun yang dilakukan meliputi pengujian terhadap kekuatan kebun kemampuan tanah dalam menyerap curah hujan. Pengujian dilakukan untuk memastikan apakah kebun dapat menahan curah hujan yang berlebih dan mendukung sistem *monitoring* yang telah dibuat agar dapat mencapai target yang diinginkan.

3.4.3 Uji Coba Keseluruhan

Uji coba ini dilakukan untuk melihat hasil pengujian terhadap keseluruhan sistem yang dibuat. Uji coba yang dilakukan antara lain :

1. Uji coba *monitoring* kebun via *Telegram*.
2. Uji coba membaca data *recording* hasil *monitoring*.

3.5. Analisis Data

Pada tahap ini dilakukan analisis pada hasil uji coba untuk melihat kekurangan terhadap alat yang dibuat, baik itu terhadap konstruksi, rangkaian sistem, maupun pemrograman yang dibuat. Apabila terdapat kesalahan pada sistem yang dibuat maka akan dilakukan perbaikan terhadap sistem yang mengalami permasalahan tersebut baik pada rangkaian maupun program yang telah dibuat agar sesuai dengan target yang diinginkan.

3.6. Pembuatan Laporan

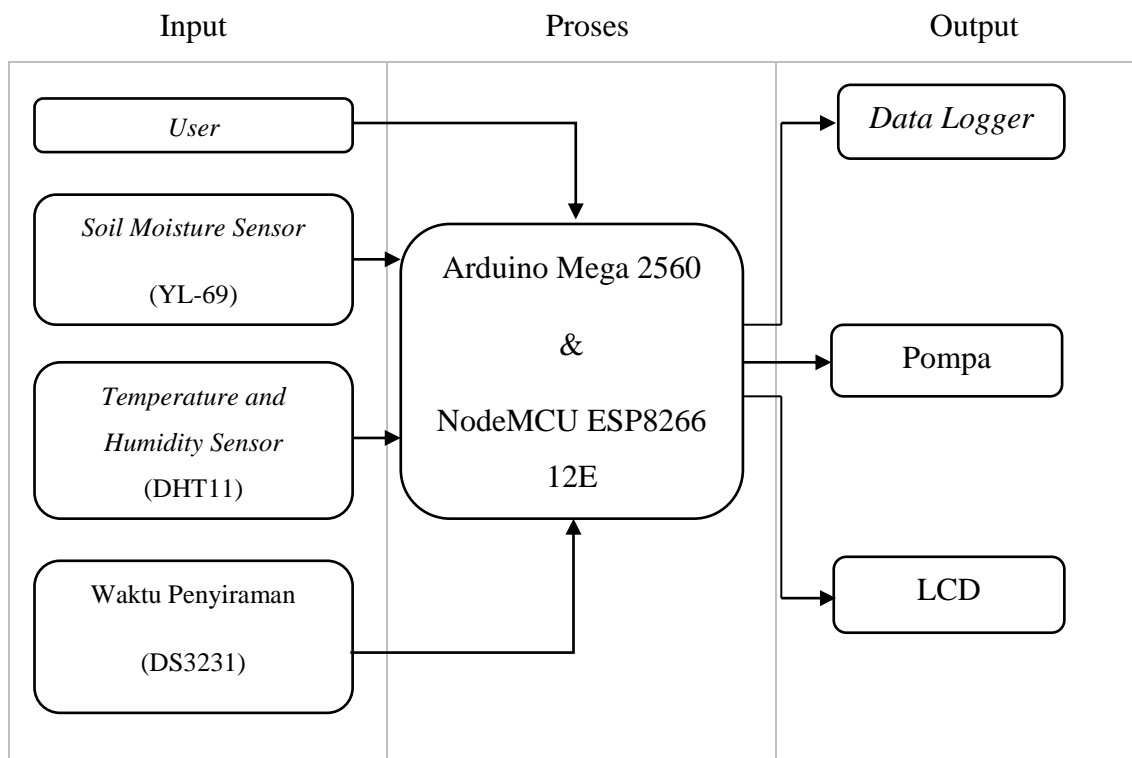
Tahap ini merupakan tahap terakhir dalam pembuatan Proyek Akhir yang bertujuan untuk merangkum keseluruhan detail mengenai alat yang dibuat, yang meliputi latar belakang, tujuan, rumusan masalah, batasan masalah, landasan teori, metode pelaksanaan, pembahasan, serta kesimpulan dan saran.

BAB IV PEMBAHASAN

Pada bab ini akan dibahas mengenai proses serta metode yang digunakan dalam pembuatan Proyek Akhir dengan judul “*Monitoring Kebun Sawi Via Telegram*” yaitu sebagai berikut :

4.1. Diagram Blok

Berikut Diagram blok dari *Monitoring Kebun Via Telegram* ditunjukkan pada gambar 4.1 :



Gambar 4.1 Diagram Blok *Monitoring Kebun Via Telegram*

Monitoring dilakukan berdasarkan waktu penyiraman, kelembaban tanah, serta suhu lingkungan, berdasarkan inputan dari sensor – sensor yang digunakan. Serta instruksi langsung dari *user* (petani) dengan memanfaatkan jaringan internet

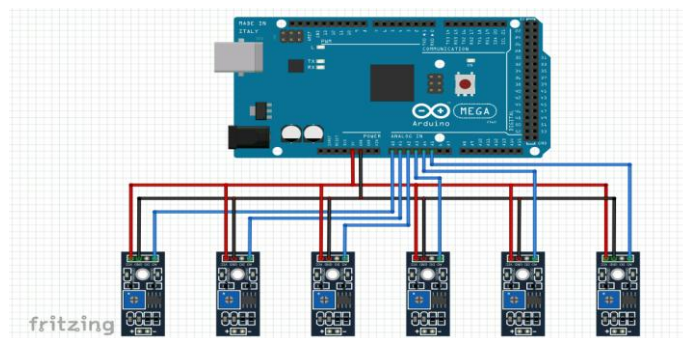
yang ada, yang nantinya data akan diolah apakah perlu melakukan penyiraman atau hanya sekedar memantau dan merekam kondisi kebun pada saat itu.

4.2 Perancangan Sistem *Monitoring*

Perancangan sistem *monitoring* terbagi menjadi beberapa bagian komponen yang digunakan, masing – masing komponen memiliki bermacam karakteristik dan skematik rangkaian yang berbeda pula.

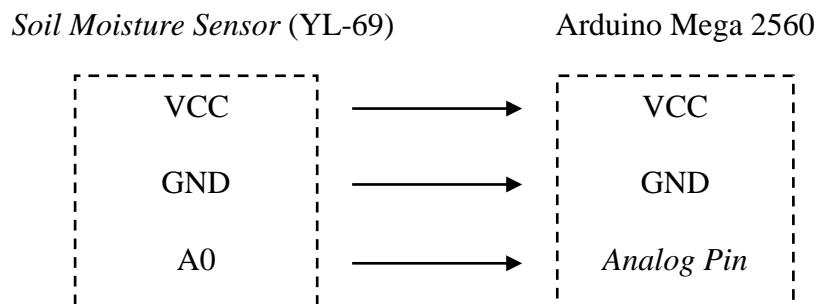
4.2.1. Perancangan Rangkaian *Soil Moisture Sensor (YL-69)*

Soil Moisture Sensor memiliki pin A0 atau pin yang menghasilkan tegangan analog sesuai dengan kelembaban tanah yang dibacanya. Untuk mendapatkan tegangan analog yang sesuai dengan kelembaban tanah diperlukan sebuah skema perangkaian sensor yang tepat. Berikut skematik *Soil Moisture Sensor (YL-69)* ditunjukkan pada gambar 4.2 :



Gambar 4.2 Skematik *Soil Moisture Sensor (YL-69)*

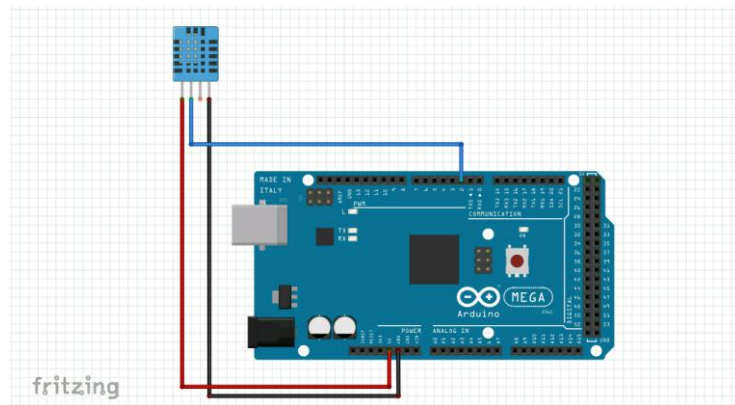
Berikut keterangan hubungan pin *Soil Moisture Sensor (YL-69)* ke Arduino Mega 2560 :



Gambar 4.3 Hubungan Pin *Soil Moisture Sensor (YL-69)*

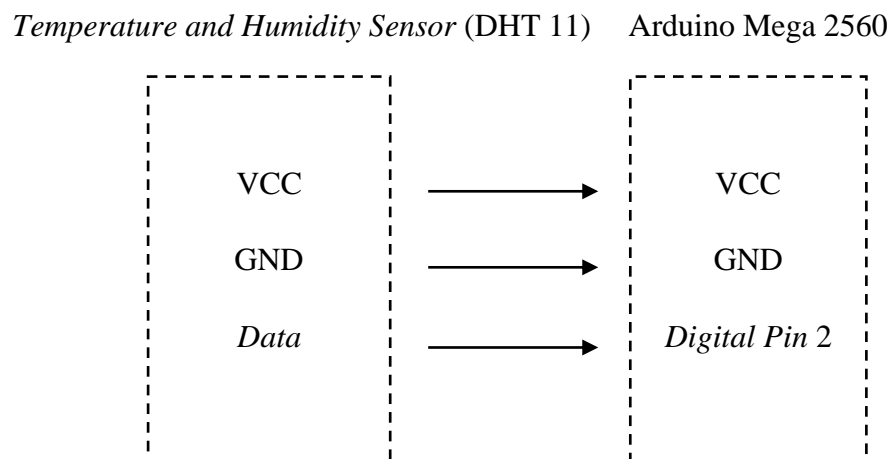
4.2.2. Perancangan Rangkaian *Temperature and Humidity Sensor (DHT 11)*

DHT 11 merupakan sebuah sensor temperatur dan kelembaban udara memerlukan sebuah skema perangkaian sensor yang tepat agar dapat bekerja seperti pada skematik rangkaian. Berikut skematik *Temperature and Humidity Sensor (DHT 11)* ditunjukkan pada gambar 4.4 :



Gambar 4.4 Skematik *Temperature and Humidity Sensor (DHT 11)*

Berikut keterangan hubungan pin *Temperature and Humidity Sensor (DHT 11)* ke Arduino Mega 2560 :

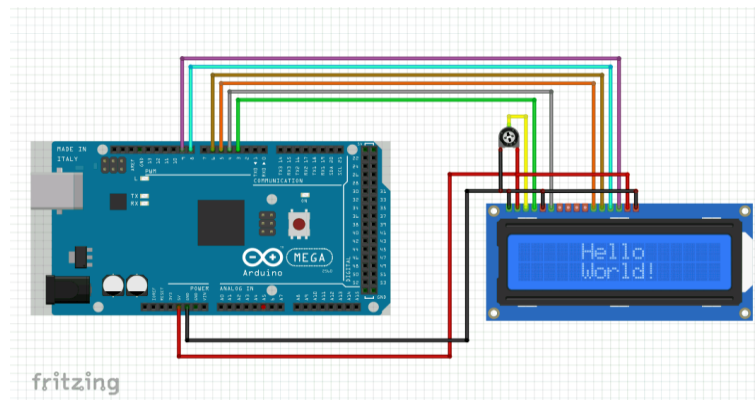


Gambar 4.5 Hubungan Pin *Temperature and Humidity Sensor (DHT 11)*

4.2.3. Perancangan Rangkaian LCD (*Liquid Crystal Display*)

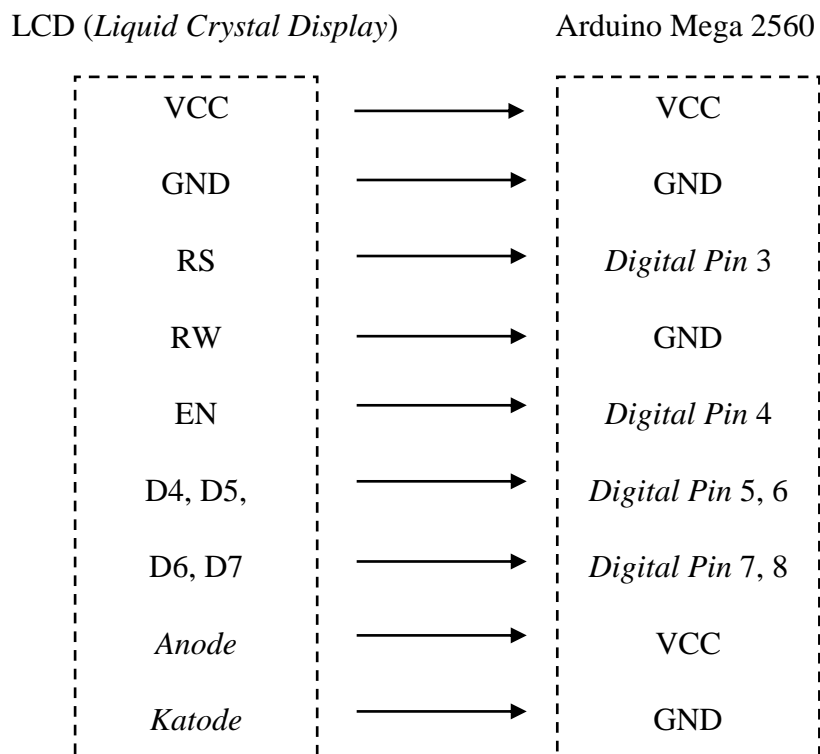
LCD (*Liquid Crystal Display*) digunakan sebagai *display* elektronik yang berfungsi untuk menampilkan data – data sensor, waktu, dan suhu yang terbaca

pada saat itu, maka dari itu agar LCD dapat menampilkan data – data seperti seharusnya maka dirancanglah sebuah skema yang tepat seperti pada skematik rangkaian. Berikut skematik LCD (*Liquid Crystal Display*) ditunjukkan pada gambar 4.6 :



Gambar 4.6 Skematik LCD (*Liquid Crystal Display*)

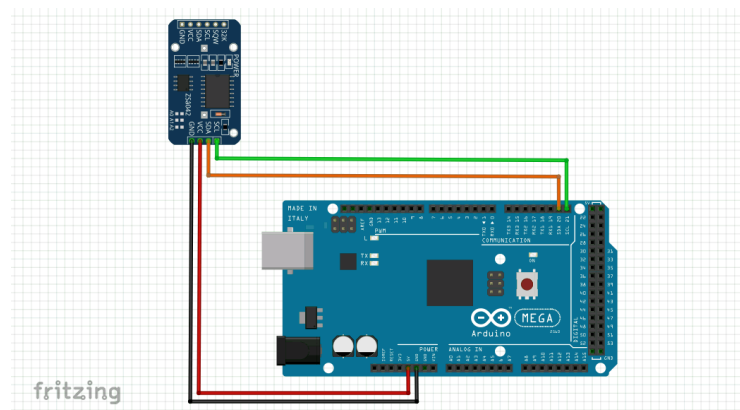
Berikut keterangan hubungan pin LCD (*Liquid Crystal Display*) ke Arduino Mega 2560 :



Gambar 4.7 Hubungan Pin LCD (*Liquid Crystal Display*)

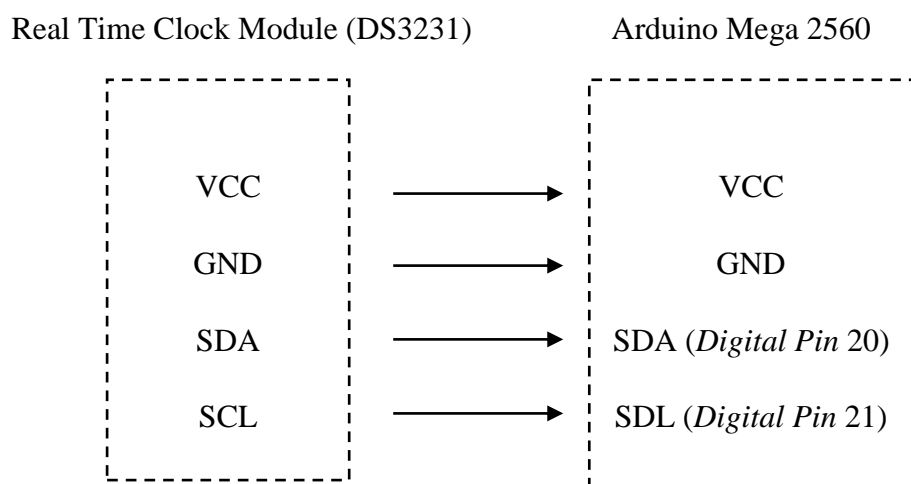
4.2.4. Perancangan Rangkaian *Real Time Clock Module (DS3231)*

RTC merupakan sebuah modul yang berfungsi untuk mencacah waktu dan tanggal secara langsung, pada modul DS3231 ini menggunakan komunikasi TWI dengan pin SDA, dan SCL. Agar modul dapat bekerja maka dirancanglah sebuah skema yang tepat seperti pada skematik rangkaian. Berikut skematik RTC Module DS3231 ditunjukan pada gambar 4.8 :



Gambar 4.8 Skematik RTC Module (DS3231)

Berikut keterangan hubungan pin Real Time Clock Module (DS3231) ke Arduino Mega 2560 :

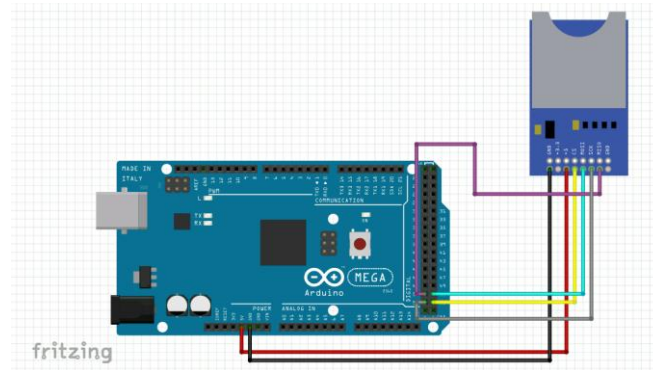


Gambar 4.9 Hubungan Pin Real Time Clock Module (DS3231)

4.2.5. Perancangan Rangkaian *SD Card Module*

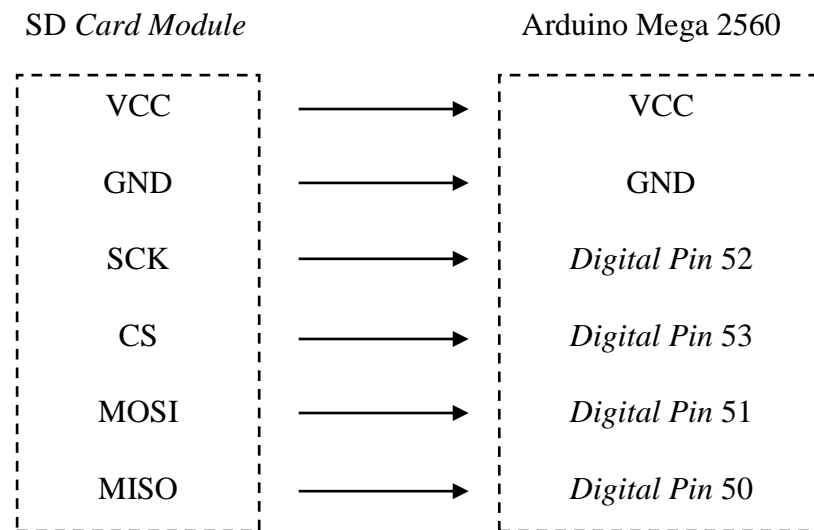
SD Card Module berfungsi adapter yang dapat menyimpan data – data hasil *monitoring* didalam sebuah *micro SD*. Agar modul dapat bekerja seperti

seharusnya maka dirancanglah sebuah skema yang tepat pada skematik rangkaian. Berikut skematik SD *Card Module* ditunjukkan pada gambar 4.10 :



Gambar 4.10 Skematik SD *Card Module*

Berikut keterangan hubungan pin SD *Card Module* ke Arduino Mega 2560 :

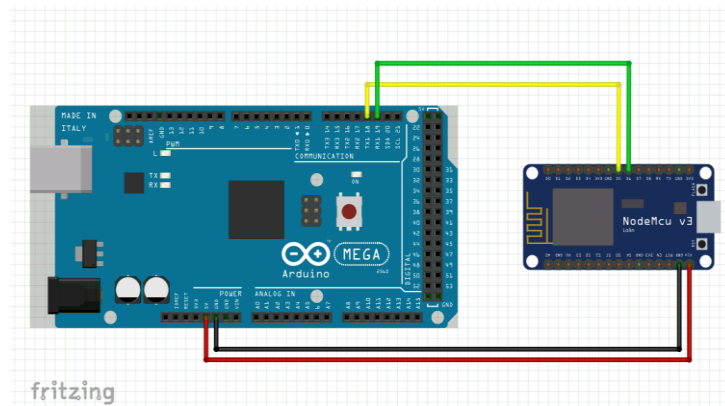


Gambar 4.11 Hubungan Pin SD *Card Module*

4.2.6. Perancangan Rangkaian NodeMCU ESP8266 12E

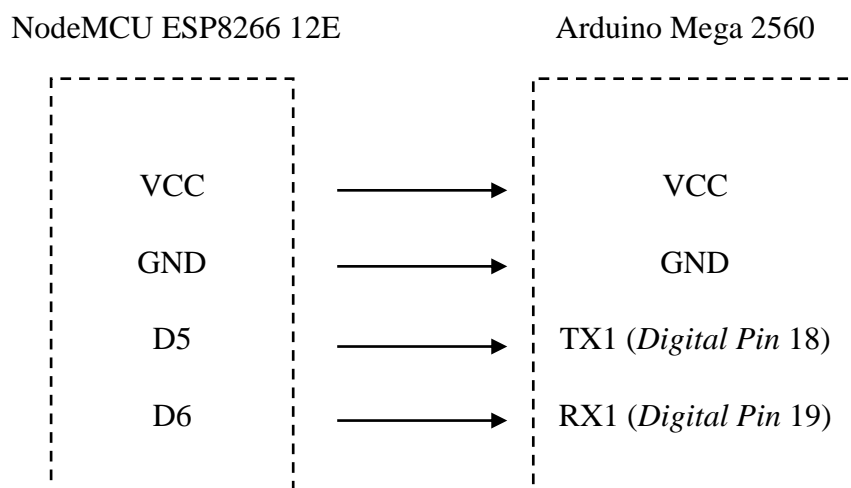
NodeMCU ESP8266 12E berfungsi sebagai modul *wifi* yang digunakan sebagai jalur komunikasi dengan user melalui jaringan internet. Sehingga *user* dapat memantau kebunnya dari mana dan kapan saja. Agar modul dapat bekerja seperti seharusnya maka dirancanglah sebuah skema yang tepat pada skematik rangkaian.

Berikut skematik NodeMCU ESP8266 12E ditunjukkan pada gambar 4.12 :



Gambar 4.12 Skematik NodeMCU ESP8266 12E

Berikut keterangan hubungan pin NodeMCU ESP8266 12E ke Arduino Mega 2560 :

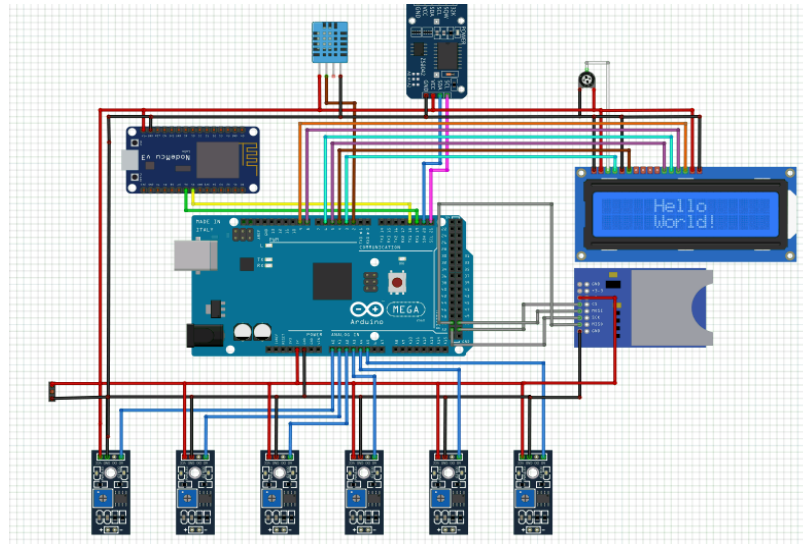


Gambar 4.13 Hubungan Pin NodeMCU ESP8266 12E

4.2.7. Perancangan Rangkaian Keseluruhan

Setelah mengetahui macam – macam dari skematik setiap komponen yang digunakan, maka terbentuklah sebuah skematik rangkaian sistem *Monitoring* yang dapat memantau kondisi kebun kapanpun dan dimanapun, secara *real time* dan dapat menyimpan data – data kondisi kebun pada saat itu dalam *SD card* ataupun dalam bentuk grafik didalam *cloud Thinkspeak*.

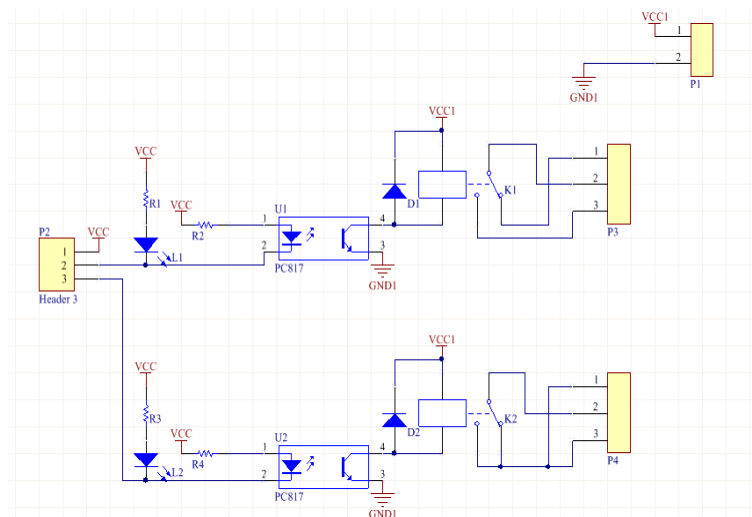
Berikut skematik keseluruhan rangkaian ditunjukkan pada gambar 4.14 :



Gambar 4.14 Skematik Keseluruhan Rangkaian

4.3. Perancangan Rangkaian *Driver* Pompa

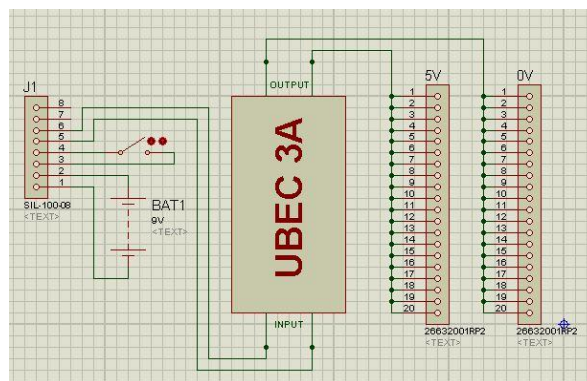
Driver pompa digunakan untuk mematikan dan menghidupkan pompa melalui perintah sinyal digital yang dikeluarkan oleh mikrocontroller. Berikut skematik *driver* pompa ditunjukkan pada gambar 4.9 :



Gambar 4.15 Skematik *Driver* Pompa

4.4. Perancangan Rangkaian Catu Daya

Rangkaian catu daya berfungsi sebagai rangkaian yang membagi tegangan dari *Power Supply* menggunakan UBEC (*Universal Battery Elimination Circuit*) 3A untuk menghasilkan tegangan yang diinginkan yaitu 5 VDC untuk Mikrokontroler Arduino Mega 2560, NodeMCU ESP8266 12E, DHT 11, RTC, DS3231, LCD 16x2, SD Card Module, Soil Moisture Sensor (YL-69), serta 12 VDC sebagai *supply* pompa yang digunakan. Berikut skematik catu daya ditunjukkan pada gambar 4.10 :



Gambar 4.16 Skematik Catu Daya

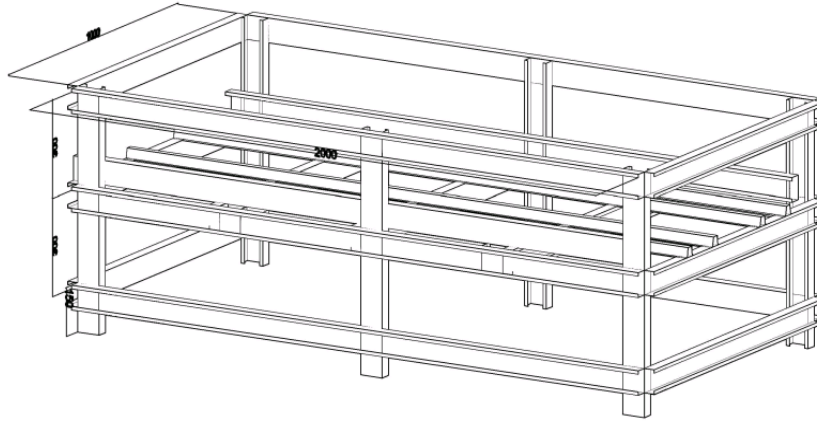
4.5. Perancangan Konstruksi Kebun

Dalam perancangan konstruksi kebun yang dibuat fungsi kebun harus dapat mendukung sistem *monitoring* yang dibuat agar mendapatkan hasil yang maksimal, oleh karena itu perancangan kebun terbagi menjadi dua bagian, seperti berikut :

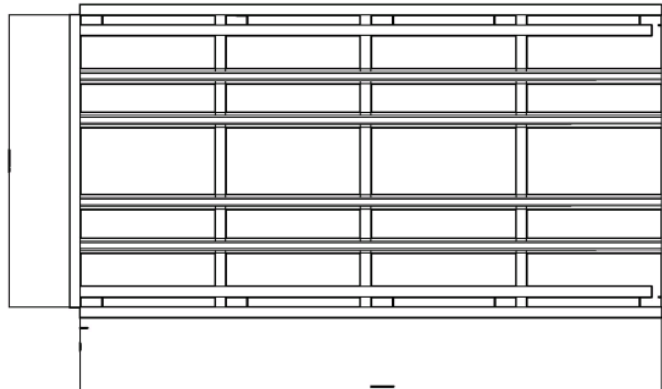
4.5.1. Perancangan Rangka Kebun

Rancangan rangka kebun dibatasi dengan ukuran panjang 2m, lebar 1m, dan tinggi 0,75m dengan menggunakan material baja ringan yang dilapisi papan dan *polyback* sebagai wadah dari media tanam. Serta sistem saluran pengairan kebun untuk menghindari kelebihan air didalam rangka.

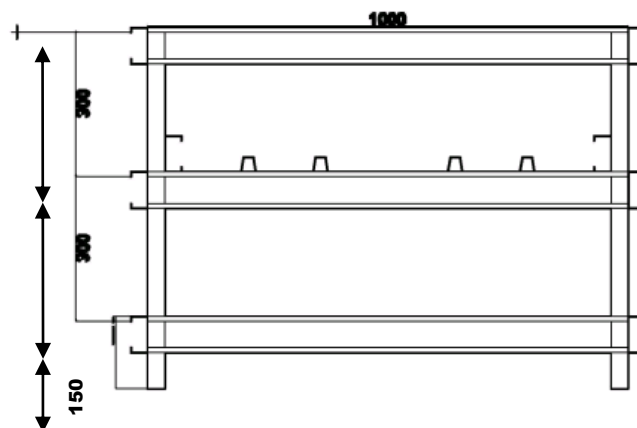
Berikut rancangan rangka kebun ditunjukkan pada gambar 4.17, 4.18, 4.19 :



Gambar 4.17 Tampak Samping Rangka Kebun



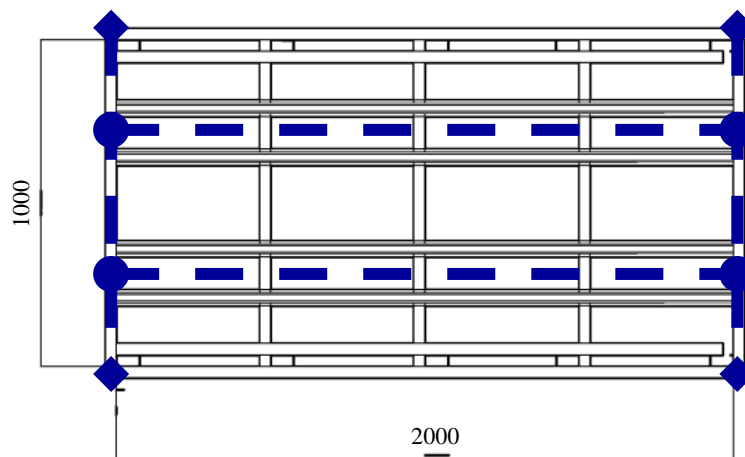
Gambar 4.18 Tampak Atas Rangka Kebun



Gambar 4.19 Tampak Depan Rangka Kebun

4.5.2. Perancangan Pemipaan Penyiraman Kebun

Sistem penyiraman menggunakan pompa air *DC* yang dapat dikontrol secara otomatis ataupun manual sehingga dapat mendukung sistem *monitoring*, agar mudah mendapatkan kelembaban tanah yang sesuai dengan jenis tanaman. Berikut pemipaan penyiraman kebun ditunjukkan pada gambar 4.14 :



Gambar 4.20 Pemipaan Penyiraman Kebun

4.6. Pembuatan Sistem *Monitoring*

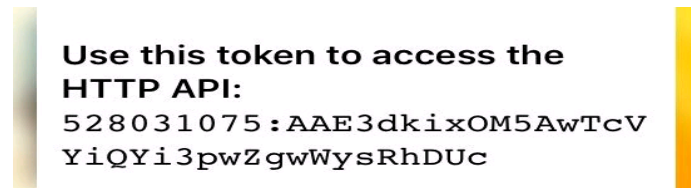
Setelah didapatkan rancangan dari sistem *monitoring*. Selanjutnya masuk ke dalam tahap pembuatan dan pengujian komponen – komponen apakah dapat bekerja sesuai dengan yang diinginkan. Adapun tahap – tahap pembuatan sistem *monitoring* terbagi menjadi beberapa bagian, seperti berikut :

4.6.1. Pengaturan *Bot User Telegram*

Berikut merupakan langkah – langkah membuat *bot user* :

- Buka <https://t.me/Manybot>
- Klik “*Create a new Bot*”.
- Kemudian, klik **@BotFather** untuk membuat nama *bot* dan mendapatkan token API. *Bot Father* adalah bot resmi dari telegram yang tugasnya membuat *bot* baru.

- Tuliskan nama Bot.
- Kemudian tuliskan username untuk bot.
- Kita akan mendapatkan sebuah token API, sebagai alamat komunikasi antara *user* dan *object*.

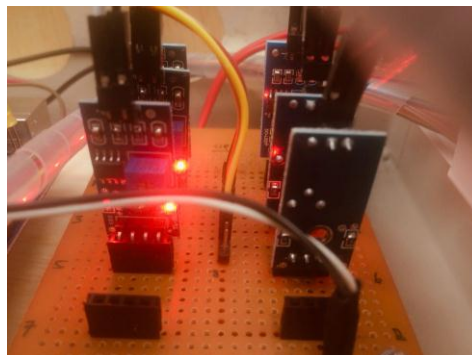


Gambar 4.21 Pesan *Token API*

4.6.2. Pembuatan Rangkaian *Soil Moisture Sensor (YL-69)*

Pembuatan rangkaian berdasarkan dengan rancangan skematik yang telah dibuat.

Berikut Rangkaian *Soil Moisture Sensor (YL-69)* pada ditunjukkan gambar 4.16 :



Gambar 4.22 Rangkaian *Soil Moisture Sensor (YL-69)*

Setelah pembuatan rangkaian selesai, lalu masuk ke tahap *coding* program dengan memanfaatkan keluaran analog dari sensor yang dikonversikan kedalam satuan kelembaban tanah (persen).

```
#define analog_max 1023
// read the input on analog pin 0:
int sensorValue1 = analogRead(A0);
int sensorValue2 = analogRead(A1);
int sensorValue3 = analogRead(A2);
int sensorValue4 = analogRead(A3);
```

Mendefinisikan nilai maksimum adc 10 bit, dan mendeklarasikan pembacaan nilai analog ke dalam variabel.

```

int sensorValue5 = analogRead(A4);
int sensorValue6 = analogRead(A5);
// Converting data
int S1 = (0.12) * (analog_max - sensorValue1);
int S2 = (0.12) * (analog_max - sensorValue2);
int S3 = (0.12) * (analog_max - sensorValue3);
int S4 = (0.12) * (analog_max - sensorValue4);
int S5 = (0.12) * (analog_max - sensorValue5);
int S6 = (0.12) * (analog_max - sensorValue6);

```

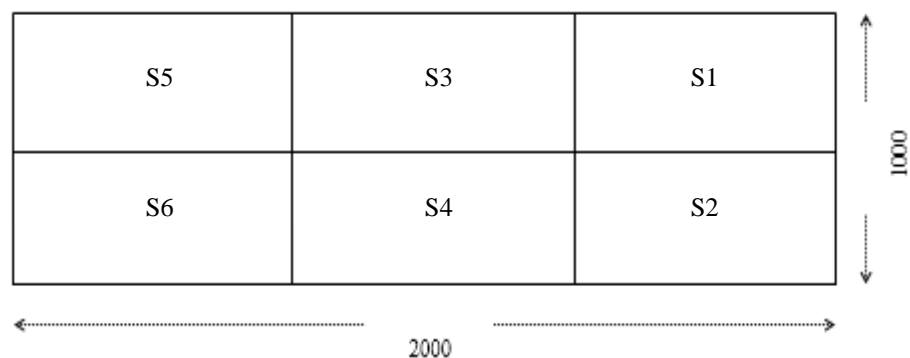
Merubah data analog kedalam bentuk satuan kelembaban tanah (persen).

Berikut Pembacaan *Soil Moisture Sensor* (YL-69) ditunjukkan pada gambar 4.17 :



Gambar 4.23 Pembacaan *Soil Moisture Sensor* (YL-69)

Untuk mengetahui tingkat keakuratan pembacaan sensor, maka dilakukan perbandingan hasil pembacaan sensor dan alat ukur kelembaban tanah *Soil Water 3 in 1*. Berikut *Mapping* peletakan *Soil Moisture sensor* (YL-69) ditunjukkan pada gambar 4.18 :



Gambar 4.24 *Mapping* Peletakan *Soil Moisture Sensor* (YL-69)

Adapun hasil pengukuran pertama yang dilakukan ditunjukkan pada tabel 4.1 berikut :

Tabel 4.1 Hasil Pengukuran *Soil Moisture Sensor* 1

No.	P. Sensor (%)			P. Alat Ukur (%)			Error (%)			Rata – rata Error (%)
S1	71	72	73	65	70	70	6	2	3	3.66
S2	71	70	72	70	70	69	1	0	3	1.33
S3	72	72	72	69	70	69	3	2	3	2.66
S4	69	69	69	70	69	69	1	0	0	0.33
S5	69	67	68	68	70	69	1	3	1	1.66
S6	67	65	66	65	65	68	2	0	2	1.33

Ket : Pengukuran dilakukan pada pagi hari pada pukul 06.35 – 07.00 dengan kondisi cuaca gerimis.

- Pengukuran pada S1 :



Gambar 4.25 Pengukuran S1 Pertama

- Pengukuran pada S2 :



Gambar 4.26 Pengukuran S2 Pertama

- Pengukuran pada S3 :



Gambar 4.27 Pengukuran S3 Pertama

- Pengukuran pada S4 :



Gambar 4.28 Pengukuran S4 Pertama

- Pengukuran pada S5 :



Gambar 4.29 Pengukuran S5 Pertama

- Pengukuran pada S6 :



Gambar 4.30 Pengukuran S6 Pertama

Setelah pengukuran pertama dilakukan pada pagi hari. Pengukuran kedua dilakukan pada sore hari dengan 6 titik pengukuran yang dilakukan. Pengukuran terbagi menjadi 3 bagian pada satu balur, yaitu bagian kiri, tengah dan kanan juga seperti pada gambar 4.24 diatas. Pengukuran kedua bertujuan untuk mengetahui perbedaan kondisi kelembaban tanah pada waktu dan suhu dari waktu – waktu sebelumnya.

Berikut hasil pengukuran kedua yang dilakukan ditunjukkan seperti pada tabel 4.2 berikut :

Tabel 4.2 Hasil Pengukuran *Soil Moisture Sensor 2*

No.	P. Sensor (%)			P. Alat Ukur (%)			Error (%)			Rata – rata Error (%)
S1	69	70	70	61	65	65	8	5	5	6
S2	69	71	70	61	69	70	8	2	0	3.33
S3	68	70	70	62	68	70	6	2	0	2.66
S4	66	68	68	61	69	66	5	1	2	2.66
S5	67	68	67	65	70	70	2	2	3	2.33
S6	62	64	66	60	63	61	2	1	5	2.66

Ket : Pengukuran dilakukan pada sore hari pada pukul 17.35 – 18.00 dengan kondisi cuaca berawan.

Berikut gambar hasil pengukuran 2 :

- Pengukuran pada S1



Gambar 4.31 Pengukuran S1 Kedua

- Pengukuran pada S2



Gambar 4.32 Pengukuran S2 Kedua

- Pengukuran pada S3



Gambar 4.33 Pengukuran S3 Kedua

- Pengukuran pada S4



Gambar 4.34 Pengukuran S4 Kedua

- Pengukuran pada S5



Gambar 4.35 Pengukuran S5 Kedua

- Pengukuran pada S6



Gambar 4.36 Pengukuran S6 Kedua

Berdasarkan hasil pengukuran yang telah didapatkan, dapat dianalisa bahwa persentase *error* dapat dihitung dengan rumus :

$$|Error (\%)| = P. Sensor (\%) - P. Alat Ukur(\%) \dots\dots\dots (4.1)$$

Dan dapat disimpulkan pembacaan memiliki rata – rata *error* dari seluruh pengukuran data yaitu sebesar 2.54 % terhadap alat ukur dan dapat berubah – ubah dipengaruhi beberapa faktor seperti kepadatan tanah dan kedalaman peletakan alat ukur.

4.6.3. Pembuatan Rangkaian *Temperature and Humidity Sensor (DHT 11)*

Pembuatan rangkaian berdasarkan dengan rancangan skematik yang telah dibuat. Berikut peletakan sensor DHT 11 ditunjukkan pada gambar 4.37 :



Gambar 4.37 Peletakan Sensor DHT11

Setelah instalasi sensor selesai, masuk ke tahap *coding* program agar sensor dapat bekerja.

```
#include "DHT.h"
#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

float h = dht.readHumidity();
// Read temperature as Celsius (the default)
float t = dht.readTemperature();
// Read temperature as Fahrenheit (isFahrenheit = true)
float f = dht.readTemperature(true);

if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
}

// Compute heat index in Fahrenheit (the default)
float hif = dht.computeHeatIndex(f, h);
// Compute heat index in Celsius (isFahreheit = false)
float hic = dht.computeHeatIndex(t, h, false);
```

Memanggil *lib* DHT11.
Mendefinisikan *pin* yang digunakan.
Mendefinisikan tipe DHT yang digunakan.
Mendeklarasikan *object* dari *lib* DHT11.

Memanggil fungsi pembacaan suhu dan kelembaban udara yang dideklarasikan ke dalam variabel dengan tipe data *float*.

Berikut Pembacaan *Temperature and Humidity Sensor* (DHT 11) ditunjukkan pada gambar 4.38 :



Gambar 4.38 Pembacaan *Temperature and Humidity Sensor* (DHT 11)

Untuk mengetahui tingkat keakuratan pembacaan sensor DHT 11 dilakukan beberapa pengambilan data yang dibandingkan dengan alat ukur suhu dan kelembaban udara HTC-2. Adapun hasil pengukuran ditunjukkan pada tabel berikut :

Tabel 4.3 Hasil Pengukuran *Temperature* (DHT 11)

No.	Waktu	P. Sensor (°C)			P. Alat Ukur (°C)			Error (°C)			Rata – rata Error (°C)
1	06.40	29	28	28	27.9	27.7	28.1	1.1	0.3	0.1	0.5
2	14.00	40	38	38	35.2	34.9	34.1	4.8	3.1	3.9	3.93
3	18.35	27	27	27	27.7	27.7	27.7	0.7	0.7	0.7	0.7

Ket : Pengukuran dilakukan pada kondisi cuaca cerah.

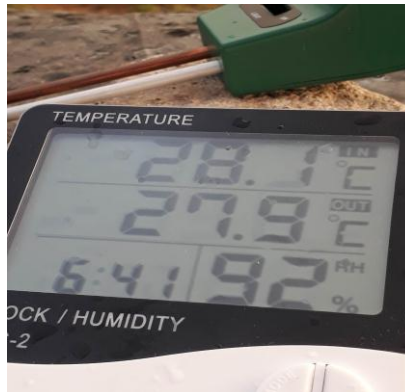
Tabel 4.4 Hasil Pengukuran *Humidity* (DHT 11)

No.	Waktu	P. Sensor (%RH)			P. Alat Ukur (%RH)			Error (%RH)			Rata – rata Error (%RH)
1	06.40	95	95	95	92	97	99	3	2	4	3
2	14.00	53	51	52	51	51	54	2	0	2	1.33
3	18.35	79	79	80	77	77	77	2	2	3	2.33

Ket : Pengukuran dilakukan pada kondisi cuaca cerah.

Berikut gambar hasil pengujian suhu dan kelembaban udara yang dilakukan pada 3 waktu dan kondisi yang berbeda :

- Pengujian 1 :



Gambar 4.39 Pengujian Sensor DHT 11 Pertama

- Pengujian 2 :



Gambar 4.40 Pengujian sensor DHT 11 Kedua

- Pengujian 3 :



Gambar 4.41 Pengujian sensor DHT 11 Ketiga

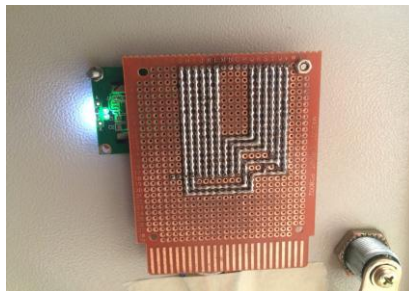
Berdasarkan hasil pengukuran yang didapat, dapat dianalisa bahwa pembacaan sensor suhu memiliki *error temperature* sebesar 1.71°C dan *error humidity* sebesar 2.22%RH terhadap alat ukur dan berikut merupakan perhitungan *error* yang telah dilakukan :

$$\text{Error total} = \frac{\text{Rata - rata error 1} + \text{Rata - rata error 2} + \text{Rata - rata error 3}}{3}$$

Dan dapat disimpulkan bahwa data pembacaan DHT11 dapat berubah sesuai dengan faktor cuaca, lingkungan sekitar, sensitivitas sensor itu sendiri.

4.6.4. Pembuatan Rangkaian LCD (*Liquid Crystal Display*)

Pembuatan rangkaian berdasarkan dengan rancangan skematik yang telah dibuat. Berikut rangkian LCD (*Liquid Crystal Display*) ditunjukkan pada gambar 4.42 :



Gambar 4.42 Rangkaian LCD (*Liquid Crystal Display*)

Setelah rangkaian selesai, masuk ke tahap *coding* program untuk memastikan apakah LCD dapat bekerja sesuai dengan yang diinginkan.

<pre>#include <LiquidCrystal.h> LiquidCrystal lcd(12, 11, 5, 4, 3, 2);</pre>	}	<p>Memanggil <i>lib</i> LCD dan mendefinisikan <i>object</i> dengan pin yang digunakan</p>
<pre>lcd.begin(16, 2); lcd.setCursor(0, 1); lcd.print("hello, world!");</pre>	}	<p>Memulai fungsi dari <i>lib</i> LCD dengan tipe 16x2, dan menset posisi awal karakter, serta menuliskan karakter yang akan ditampilkan</p>

Berikut hasil yang ditampilkan oleh LCD. Berikut Pengujian Tampilan LCD 16x2 ditunjukkan pada gambar 4.43 :



Gambar 4.43 Pengujian Tampilan LCD 16x2

Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa LCD dapat bekerja sesuai dengan yang diinginkan.

4.6.5. Pembuatan Rangkaian *Real Time Clock* DS3231

Pembuatan rangkaian berdasarkan dengan rancangan skematik yang telah dibuat. Berikut Perangkaian Real Time Clock DS3231 ditunjukkan pada gambar 4.38 :



Gambar 4.44 Perangkaian Real Time Clock DS3231

Setelah rangkaian selesai, masuk ke tahap *coding* program untuk memastikan apakah LCD dapat bekerja sesuai dengan yang diinginkan.

```
#include <DS3231.h>
DS3231 rtc(SDA, SCL);
Serial.print(rtc.getDOWStr());
Serial.print(tc.getDateStr());
Serial.println(rtc.getTimeStr());
```

} Memanggil *lib* DS3231 dan mendefinisikan *object lib*.

} Memanggil fungsi penunjukan jam dan tanggal dan ditampilkan di *serial monitor*

Berikut tampilan tanggal dan waktu RTC ditunjukkan pada gambar 4.45 :



Gambar 4.45 Tampilan Tanggal dan Waktu RTC

Adapun pengujian pencacahan waktu RTC ditunjukkan pada tabel berikut :

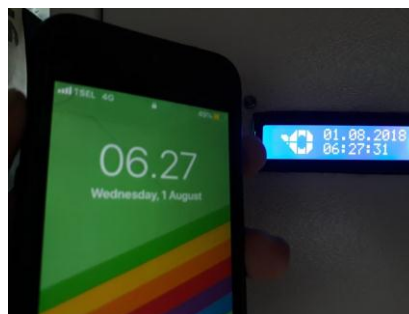
Tabel 4.5 Pengujian Pencacahan Waktu RTC

RTC	Jam Handphone	Error (minute)
06:27	06:27	0
08:11	08:11	0
09:15	09:15	0

Ket : Pengujian berdasarkan selisih menit.

Berikut gambar hasil pengujian yang dilakukan menggunakan perbandingan dengan waktu pada *handphone* :

- Pengujian 1 :



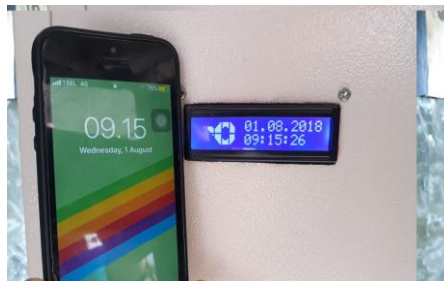
Gambar 4.46 Pengujian RTC Pertama

- Pengujian 2 :



Gambar 4.47 Pengujian RTC Kedua

- Pengujian 3 :

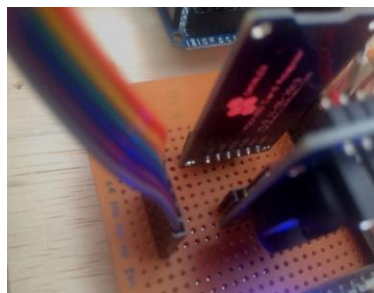


Gambar 4.48 Pengujian RTC Ketiga

Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa RTC dapat mencacah waktu sesuai dengan waktu yang diinginkan dan presisian yang cukup baik.

4.6.6. Pembuatan Rangkaian SD Card Module

Pembuatan rangkaian berdasarkan dengan rancangan skematik yang telah dibuat. Berikut rangkaian SD Card Module ditunjukan pada gambar 4.43 :



Gambar 4.49 Perangkaian SD Card Module

Setelah rangkaian selesai, masuk ke tahap *coding* program untuk memastikan apakah modul SD dapat bekerja sesuai dengan yang diinginkan.

```

#include <SPI.h>
#include <SD.h>
File myFile;
myFile = SD.open("test.txt", FILE_WRITE);
if (myFile) {
  Serial.print("Writing to test.txt...");
  myFile.println("Tes 123!");
  myFile.close();
  Serial.println("done.");
}

```

} Memanggil *lib* SD *card* dan *lib* komunikasi SPI.
 } Mendeklarasikan *object lib* SD dan membuka file SD test.txt.
 } Jika file SD sudah terbuka. Selanjutnya menuliskan karakter ke dalam SD. Dan menutup file kembali.

Adapun pengujian SD *Card Module* dilakukan guna mengetahui apakah SD *Card Module* dapat menyimpan data yang ingin direkam oleh *user*. Berikut hasil penyimpanan di SD *Card Module* pada gambar 4.44 :

```

Status pada tanggal : 17 . 7 . 2018
Jam : 19 : 13 : 39
-----
Kondisi sensor Soil Moisture:
S1: 41
S2: 43
S3: 55
S4: 60
S5: 47
S6: 58
Rata - rata sensor soil: 50
-----
Dengan Suhu : 29.00°C
Kelembaban: 83.00% RH
Status pada tanggal : 18 . 7 . 2018
Jam : 7 : 41 : 22
-----
Kondisi sensor soil Moisture:
S1: 70
S2: 64
S3: 66
S4: 65
S5: 65
S6: 68
Rata - rata sensor soil: 66
-----
Dengan Suhu : 34.00°C
Kelembaban: 79.00% RH

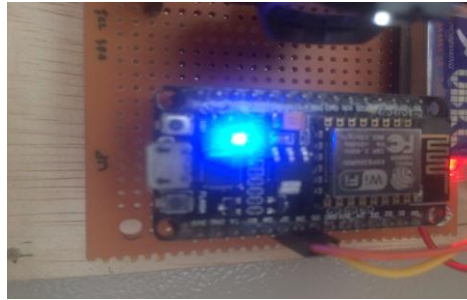
```

Gambar 4.50 Hasil Penyimpanan di SD *Card Module*

Dari hasil pengujian yang dilakukan SD *Card Module* dapat menyimpan data – data pembacaan hasil sensor kelembaban tanah, suhu, serta waktu dan tanggal sehingga dapat disimpulkan bahwa SD *card* module bekerja sesuai dengan fungsi yang diinginkan.

4.6.7. Pembuatan Rangkaian NodeMCU ESP8266 12E

Pembuatan rangkaian berdasarkan dengan rancangan skematik yang telah dibuat. Berikut rangkaian NodeMCU ESP8266 12E ditunjukkan pada gambar 4.51 :



Gambar 4.51 Perangkaian NodeMCU ESP8266 12E

Setelah rangkaian selesai, masuk ke tahap *coding* program untuk memastikan apakah NodeMCU dapat bekerja sesuai dengan yang diinginkan.

```
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
char ssid[] = "XXXXXX";
char password[] = "YYYYYY";
#define BOTtoken "XXXXXXXXX:XXXXXXXXXXXXXXXXXXXX"
UniversalTelegramBot bot(BOTtoken, client);
String text;

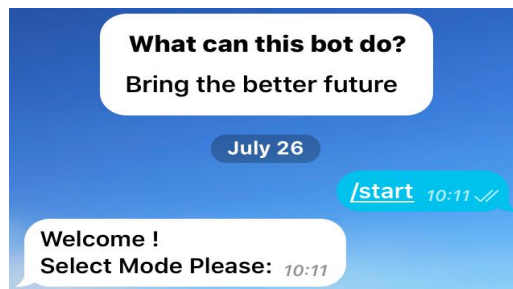
if (text == "/ledon") {digitalWrite(ledPin, HIGH);
    bot.sendMessage(chat_id, "Led is ON", "");
}
if (text == "/ledoff") {digitalWrite(ledPin, LOW);
    bot.sendMessage(chat_id, "Led is OFF", "");
}
if (text == "/start") {
    digitalWrite(ledPin, LOW);
    String welcome = "Welcome!";
    welcome += "Select Mode Please:";
    bot.sendMessage(chat_id, "welcome:", "");}
```

Memanggil *lib Telegram*, memasukan nama SSID dan Password serta token API Telegram.

Mendklarasikan *Object lib Telegram* dan

Fungsi percabangan penerima perintah dari user yang disimpan ke dalam variabel text

Adapun pengujian dilakukan untuk mengetahui apakah NodeMCU dapat terkoneksi dengan internet dan mengirim data pembacaan sensor serta menerima perintah dari *user*. Berikut pengujian *Chatting Telegram* NodeMCU pada gambar 4.46 :

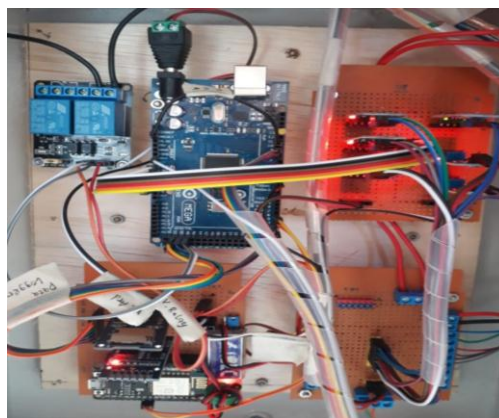


Gambar 4.52 Pengujian *Chatting Telegram* NodeMCU

Dari hasil percobaan yang dilakukan dapat disimpulkan bahwa NodeMCU dapat mengirim dan menerima perintah dari *user* menggunakan aplikasi *Telegram* yang disesuaikan dengan kebutuhan dari *user*.

4.6.8. Perakitan Rangkaian Keseluruhan

Setelah bagian – bagian komponen terpenuhi dan selesai dibuat, tahap selanjutnya masuk ke penggabungan seluruh komponen sehingga menjadi sebuah rangkaian proyek akhir “Sistem *Monitoring* Kebun Sawi *Via Telegram*”. Berikut rangkaian sistem *Monitoring* ditunjukkan pada gambar 4.47 :

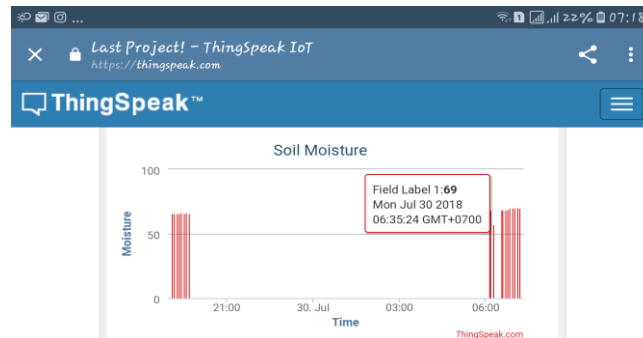


Gambar 4.53 Rangkaian Sistem *Monitoring*

Untuk mengetahui keberhasilan sistem *monitoring* yang dibuat, dilakukan beberapa pengujian yaitu :

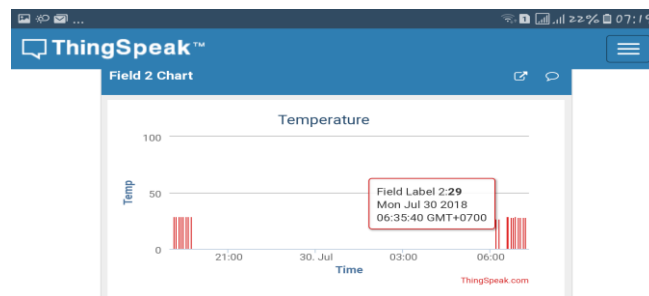
- Pengiriman data ke server *ThingSpeak* sebagai data grafik sensor – sensor

1. Pengiriman data kelembaban tanah ke server *ThingSpeak* :



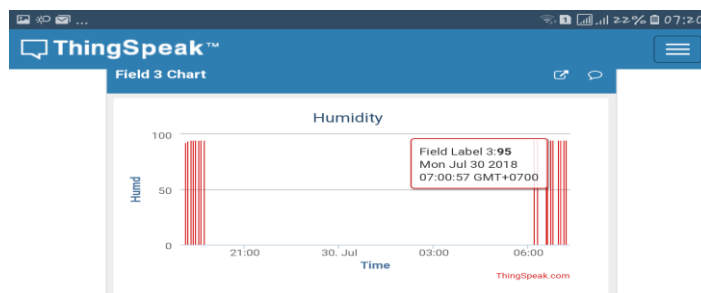
Gambar 4.54 Pengiriman Data Kelembaban Tanah Ke *ThingSpeak*

2. Hasil Pengiriman data suhu ke server *ThingSpeak* :



Gambar 4.55 Pengiriman Data Suhu Ke *ThingSpeak*

3. Hasil Pengiriman data kelembaban udara ke server *ThingSpeak* :



Gambar 4.56 Pengiriman Data Kelembaban Udara Ke *ThingSpeak*

- Pengujian waktu antara pengiriman dan penerimaan pesan dari *user* ke NodeMCU dan sebaliknya. Adapun pengujian yang dilakukan ditunjukkan pada tabel berikut :

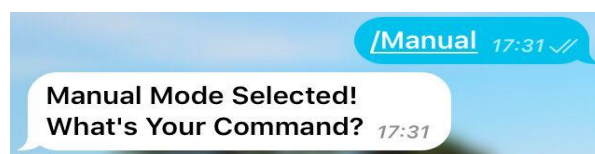
Tabel 4.6 Pengujian Waktu Pengiriman Pesan

Perintah	Waktu Pengiriman (jam)	Waktu Penerimaan (jam)	Jeda Waktu (menit)
<i>Manual</i>	17:31	17:31	0
<i>Pump Off</i>	18:35	18:36	1
<i>PumpOn</i>	18:32	18:33	1
<i>Status</i>	18:53	18:54	1
<i>Save Data</i>	18:05	18:07	2
<i>Auto</i>	17:34	17:34	0

Ket : Pengujian dilakukan pada waktu yang berbeda – beda dan menggunakan *Internet Service Provider (ISP)* yang sama dilokasi seputaran daerah Nangnung.

Berikut gambar hasil pengujian waktu kirim dan terima pesan *Telegram* :

1. Pengujian perintah “*Manual*” :



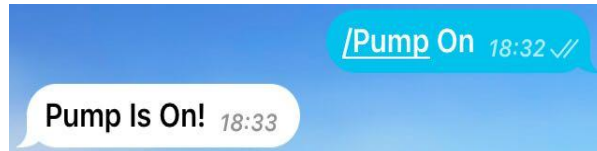
Gambar 4.57 Pengujian Perintah “*Manual*”

2. Pengujian perintah “*Pump Off*” :



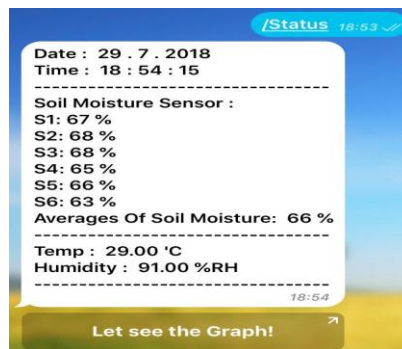
Gambar 4.58 Pengujian Perintah “*Pump Off*”

3. Pengujian perintah “*Pump On*” :



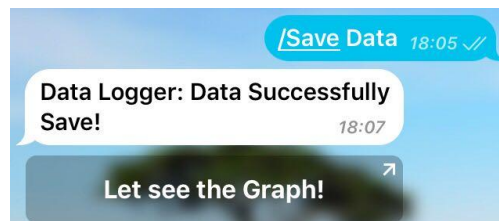
Gambar 4.59 Pengujian perintah “*Pump On*”

4. Pengujian perintah “*Status*” :



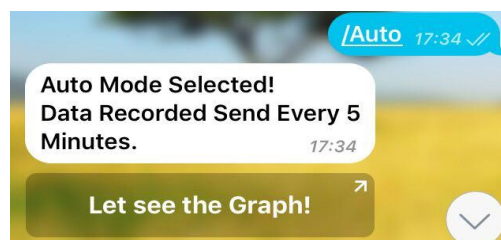
Gambar 4.60 Pengujian Perintah “*Status*” :

5. Pengujian perintah “*Save Data*” :



Gambar 4.61 Pengujian Perintah “*Save Data*”

6. Pengujian perintah “*Auto*” :



Gambar 4.62 Pengujian Perintah “*Auto*”

Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa NodeMCU dapat mengirimkan data ke server *Thinkspeak* sebagai display grafik data sensor dan waktu jeda antara pengiriman dan penerimaan pesan memiliki rata – rata jeda 0.833 menit berdasarkan perhitungan :

$$\text{Jeda rata – rata} = \frac{0 + 1 + 1 + 1 + 2 + 0}{6}$$

4.7. Pembuatan Rangkaian Catu Daya

Rangkaian catu daya memiliki peranan penting dalam rangkaian *monitoring* yang dibuat, rangkaian catu daya harus dapat menghasilkan keluaran yang sesuai dengan rangkaian *monitoring* yang dibuat. Berikut rangkaian catu daya ditunjukkan pada gambar 4.63 :



Gambar 4.63 Rangkaian Catu Daya

Adapun pengujian rangkaian catu daya ditunjukkan pada tabel berikut :

Tabel 4.7 Pengujian Rangkaian Catu Daya

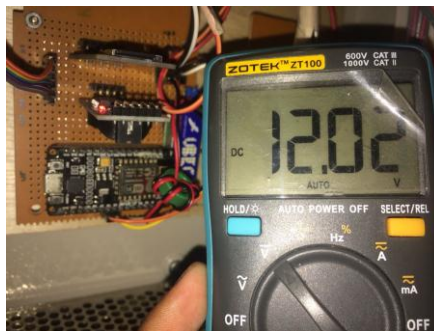
Teori (V)	Pengujian (V)	Error (V)
5	5.18	0.18
12	12.02	0.02

Ket : Pengujian menggunakan multimeter digital.

Berikut gambar hasil pengujian tegangan catu daya 5 V dan 12 V pada gambar 4.64 dan 4.65 :



Gambar 4.64 Hasil Pengujian Tegangan Catu Daya 5 V



Gambar 4.65 Hasil Pengujian Tegangan Catu Daya 12 V

Dari pengujian yang dilakukan dapat diketahui perhitungan *error* antara teori dan hasil pengujian yang telah dilakukan dengan rumus :

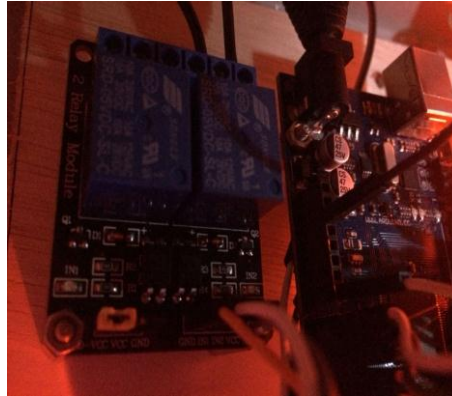
$$|Error| = \text{teori} - \text{pengujian} \dots\dots\dots (4.2)$$

Dan dapat disimpulkan bahwa rangkaian catu daya yang dibuat sesuai dengan rancangan yang diinginkan untuk kebutuhan komponen – komponen yang digunakan.

4.8. Pembuatan Rangkaian *Driver* Pompa

Rangkaian *driver* memiliki fungsi sebagai saklar elektrik yang dapat menghidupkan dan mematikan pompa penyiraman kebun yang dirangkai sesuai

dengan skema rangkaian yang telah dibuat. Berikut rangkaian *driver* pompa pada gambar 4.66 :



Gambar 4.66 Rangkaian *Driver* Pompa

Adapun pengujian rangkaian *driver* pompa ditunjukkan pada tabel berikut :

Tabel 4.8 Pengujian Rangkaian *Driver* Pompa

Nilai Digital	Keterangan
0	Relay Aktif
1	Relay Mati

Ket : Pengujian menggunakan keluaran digital dari Arduino Mega 2560.

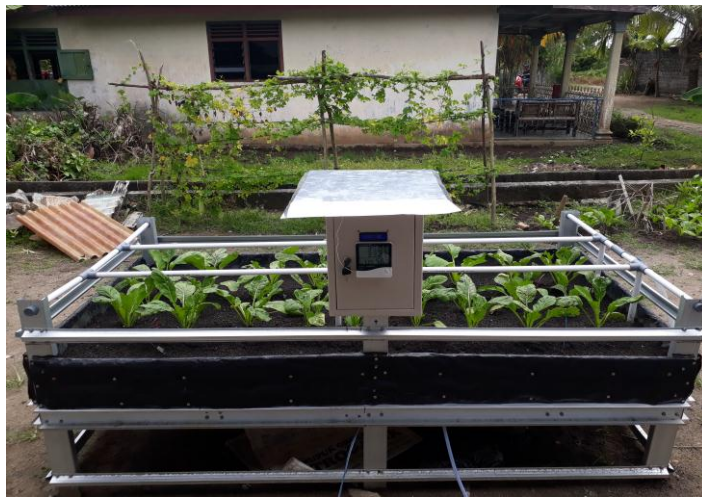
Dari pengujian yang dilakukan dapat disimpulkan rangkaian *driver* memiliki sifat *active low* atau relay akan aktif apabila diberikan logika “0” dari mikrokontroler. Adapun rangkaian yang dibuat sesuai dengan kebutuhan yang diinginkan.

4.9. Pembuatan Konstruksi Kebun

Konstruksi kebun memiliki peranan penting dalam mendukung sistem *monitoring* yang dibuat, Setelah dilakukan perancangan konstruksi kebun, selanjutnya masuk ke tahap pembuatan konstruksi kebun yang dibagi kedalam beberapa bagian :

4.9.1. Pembuatan Rangka Kebun

Setelah didapatkan rancangan kebun yang akan dibuat, selanjutnya masuk ke tahap pembuatan rangka kebun yang disesuaikan dengan rancangan. Pada Proyek Akhir ini rangka kebun dibuat menggunakan material baja ringan, dan papan cor, serta *polyback* sebagai wadah media tanam yang digunakan. Berikut hasil pembuatan rangka kebun ditunjukkan pada gambar 4.67 :



Gambar 4.67 Hasil Pembuatan Rangka Kebun

Berikut pengujian lebar kebun ditunjukkan pada gambar 4.68 :



Gambar 4.68 Pengujian Lebar Kebun

Berikut Pengujian Panjang Kebun ditunjukkan pada gambar 4.69 :



Gambar 4.69 Pengujian Panjang Kebun

4.9.2. Pembuatan Pemipaan Penyiraman Kebun

Sistem penyiraman merupakan bagian yang sangat penting dalam pembuatan kebun, maka dari itu pembuatan pemipaan penyiraman kebun, harus sesuai dengan hasil rancangan yang dibuat. Pada Proyek Akhir ini sistem penyiraman menggunakan pompa air 12 VDC yang sesuai dengan rangkaian catu daya yang dibuat. Berikut pemipaan penyiraman kebun ditunjukkan pada gambar 4.70 :



Gambar 4.70 Pemipaan Penyiraman Kebun

Adapun pengujian dilakukan dengan menghidupkan pompa dan melakukan penyiraman terhadap tanaman yang ditunjukkan. Berikut pengujian sistem penyiraman kebun ditunjukkan pada gambar 4.71 :



Gambar 4.71 Pengujian Sistem Penyiraman Kebun

Dari hasil pengujian yang dilakukan dapat diketahui bahwa sistem penyiraman berfungsi cukup baik dan sesuai dengan keinginan dan kebutuhan yang diperlukan oleh kebun.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisa terhadap fungsi alat pada Proyek Akhir dengan judul “Sistem *Monitoring* Kebun Sawi *Via Telegram*” ini maka dapat ditarik kesimpulan sebagai berikut :

1. Sistem *monitoring* kebun sawi *Via Telegram* ini dapat berkomunikasi secara dua arah dan menghilangkan batasan jarak dan waktu dalam memantau kebun sawi yang dimiliki melalui aplikasi *Telegram*. Dan waktu jeda antara pengiriman dan penerimaan pesan memiliki rata – rata jeda 0.833 menit, dan tergantung konektivitas internet saat itu.
2. Pembacaan *soil moisture sensor* (YL-69) memiliki rata – rata *error* dari seluruh pengukuran data yaitu sebesar 2.54 % terhadap alat ukur dan dapat berubah – ubah dipengaruhi beberapa faktor seperti kepadatan tanah dan kedalaman peletakan alat ukur.
3. Pembacaan sensor suhu DHT11 memiliki *error temperature* sebesar 1.71°C dan *error humidity* sebesar 2.22 %RH terhadap alat ukur.

5.2 Saran

Apabila alat ini akan dikembangkan lebih lanjut, fungsi yang perlu diperbaiki dan ditambahkan antara lain :

1. Menambahkan kamera di area kebun, karena menggunakan kamera petani sawi bisa mendapatkan visualisasi kebunnya secara langsung dimanapun dan kapanpun.
2. Menambahkan katub – katub yang dapat dikontrol secara elektrik pada sistem penyiraman sehingga penyiraman dapat dikontrol hanya pada titik – titik pada kelembaban tanah yang kurang.

DAFTAR PUSTAKA

- [1]. Mahrus Ali, waka Kogoya, Yeni Ika Pratiwi. 2017. TEKNIK BUDIDAYA TANAMAN SAWI HIJAU (*Brassica Juncea L*). Universitas Merdeka Surabaya.
- [2]. Istarofah, Zuchrotus Salamah. 2017. PERTUMBUHAN TANAMAN SAWI HIJAU (*Brassica juncea L*) DENGAN PEMBERIAN KOMPOS BERBAHAN DASAR DAUN PAITAN (*Thitonia diversifolia*). Universitas Ahmad Dahlan Yogyakarta. Vol. 03 No. 1, Mei 2017 : 39 – 46
- [3]. Dede Erawan, Wa Ode Yani, Andi Bahrun. 2013. PERTUMBUHAN DAN HASIL TANAMAN SAWI (*Brassica juncea L*) PADA BERBAGAI PUPUK UREA. Universitas Haluoleo Kendari
- [4]. Telegram FAQ [Online], diakses pada tanggal 10 Juli 2018, Available : <https://telegram.org/faq#q-what-is-telegram-what-do-i-do-here>.
- [5]. ARDUINO MEGA 2560 REV3 [Online], diakses pada tanggal 19 April 2018, Available : <https://store.arduino.cc/usa/arduino-mega-2560-rev3>.
- [6]. User Manual for ESP-12E DevKit based on ESP8266 [Online], diakses pada tanggal 20 April 2018, Available : <https://legacy.gitbook.com/book/smartarduino/user-manual-for-esp-12e-devkit/details>.
- [7]. Lcd 16x2 module datasheet [Online], diakses pada tanggal 20 April 2018, Available : <https://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>.
- [8]. Soil Moisture Sensor YL-69 Hygrometer w/ HC-38 Module Board LM393 [Online], diakses pada tanggal 20 April 2018, Available : <http://qqtrading.com.my/soil-moisture-sensor-yl-69-hygrometer-w-hc-38-module-board-lm393>.
- [9]. Digital-output relative humidity & temperature sensor/module – DHT11, [Online], diakses pada tanggal 20 April 2018, Available : http://image.dfrobot.com/image/data/DFR0067/DFR0067_DS_10_en.pdf.

- [10]. Tutorial Using DS1307 and DS3231 Real-time Clock Modules with Arduino [Online], diakses pada tanggal 20 April 2018, *Available* : <http://tronixstuff.com/2014/12/01/tutorial-using-ds1307-and-ds3231-real-time-clock-modules-with-arduino/>.
- [11]. Extremely Accurate I2C-Integrated [Online], diakses pada tanggal 20 April 2018, *Available* : <https://web.wpi.edu/Pubs/E-project/Available/E-project-010908-124414/unrestricted/DS3231-DS3231S.pdf>.
- [12]. Micro SD Card Micro SDHC mini TF Card Adapter Reader Module for Arduino [Online], diakses pada tanggal 20 April 2018, *Available* : <http://datalogger.pbworks.com/w/file/attach/89507207/Datalogger%20-%20SD%20Memory%20Reader%20Datasheet.pdf>.
- [13]. 2 CHANNEL 5V 10A RELAY MODULE [Online], diakses pada tanggal 20 April 2018, *Available* : http://www.fecegypt.com/uploads/dataSheet/1480848003_2_channel_5v_10a_relay_module.pdf.
- [14]. 3 Amp Swicth-Mode UBEC [Online], diakses pada tanggal 1 Agustus 2018, *Available* : <http://www.hobbywing.com/products/enpdf/UBEC3A2-6S.pdf>.
- [15]. Pompa Air DC 12 Volt Penguat Tekanan Air High Pressure Mini [Online], diakses pada tanggal 1 Agustus 2018, *Available* : https://www.tokopedia.com/tukura/pompa-air-dc-12-volt-penguat-tekanan-air-high-pressure-mini?ref=whead_pa1_po4&src=wish&type=1.
- [16]. API Reference [Online], diakses pada tanggal 2 Agustus 2018, *Available* : <https://www.mathworks.com/help/thingspeak/channels-and-charts-api.html>.
- [17]. Send Data to ThinkSpeak with Arduino [Online], diakses pada tanggal 2 Agustus 2018, *Available* : <https://community.thingspeak.com/tutorials/arduino/send-data-to-thingspeak-with-arduino/>.

LAMPIRAN

LAMPIRAN 1

DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Afif
Tempat dan tanggal lahir : Pangkalpinang, 11 April 1998
Alamat rumah : Kp Ulu Muntok Bangka Barat
Telp : -
Hp : 085788343657
Email : afif.polman.babel@gmail.com
Jenis kelamin : Laki - laki
Agama : Islam



2. Riwayat Pendidikan

SD Muhammadiyah	Muntok	2003 – 2009
SMP N1	Muntok	2009 – 2012
SMK N1	Muntok	2012 – 2015

3. Pendidikan Non Formal

-

Sungailiat, 1 Agustus 2018

Afif

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Ferdiansyah
Tempat dan tanggal lahir : Pangkalpinang, 30 November 1996
Alamat rumah : Jl. Kutilang 1 Bukit Baru,
Pangkalpinang Bangka
Telp : -
Hp : 081995328131
Email : Ferdipolman@gmail.com
Jenis kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

SD N10	Pangkalpinang	2003 – 2009
SMP N3	Pangkalpinang	2009 – 2012
SMA N3	Pangkalpinang	2012 – 2015

3. Pendidikan Non Formal

-

Sungailiat, 1 Agustus 2018

Ferdiansyah

LAMPIRAN 2

**Program Arduino Mega 2560: Sistem *Monitoring* Kebun Sawi Via
*Telegram***

```
#include <LiquidCrystal.h>

#include <DS3231.h>

#include <DHT.h>

#include <SPI.h>

#include <SD.h>

#define DHTPIN 2 // what digital pin we're connected to

#define DHTTYPE DHT11 // DHT 11

#define analog_max 1023

#define pump 48 // Pump's Pin

#define pump1 49 // Pump's Pin

#define ws_min_pagi 6

#define ws_max_pagi 8

#define ws_min_sore 15

#define ws_max_sore 16

#define kelembaban_ideal 60

DS3231 rtc(SDA, SCL);

LiquidCrystal lcd(3, 4, 5, 6, 8, 9);

DHT dht(DHTPIN, DHTTYPE);

File myFile;
```

```
////////// VARIABEL //////////
```

```
float h, temp, f, hif, hic;
```

```
unsigned int a, b, d, v, w, x, y, z, hh, mm, ss, tw, dd, mth, yy;
```

```
unsigned int S1, S2, S3, S4, S5, S6, SR;
```

```
byte quit=0, logger=0, mode=0, pompa=0, sd=0, treq=0, ws=0;
```

```
bool notif_on, notif_off;
```

```
char c;
```

```
String bulan;
```

```
////////////////////////////////////
```

```
void soil(){
```

```
    S1 = (0.12) * (analog_max - analogRead(A0));
```

```
    if(S1>=100)S1=100;
```

```
    S2 = (0.12) * (analog_max - analogRead(A1));
```

```
    if(S2>=100)S2=100;
```

```
    S3 = (0.12) * (analog_max - analogRead(A2));
```

```
    if(S3>=100)S3=100;
```

```
    S4 = (0.12) * (analog_max - analogRead(A3));
```

```
    if(S4>=100)S4=100;
```

```
    S5 = (0.12) * (analog_max - analogRead(A4));
```

```
    if(S5>=100)S5=100;
```

```
    S6 = (0.12) * (analog_max - analogRead(A5));
```

```
    if(S6>=100)S6=100;
```



```

SR = (S1+S2+S3+S4+S5+S6)/6;}

////////////////////////////////////

void suhu(){

    // Reading temperature or humidity takes about 250 milliseconds!

    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

    h = dht.readHumidity();

    // Read temperature as Celsius (the default)

    temp = dht.readTemperature();

    // Read temperature as Fahrenheit (isFahrenheit = true)

    f = dht.readTemperature(true);

    // Check if any reads failed and exit early (to try again).

    if (isnan(h) || isnan(temp) || isnan(f)) {

        Serial.println("Failed to read from DHT sensor!");

        return;

    }

    // Compute heat index in Fahrenheit (the default)

    hif = dht.computeHeatIndex(f, h);

    // Compute heat index in Celsius (isFahreheit = false)

    hic = dht.computeHeatIndex(temp, h, false);

}

////////////////////////////////////

```

```

void waktu(){

  logo();

  lcd.setCursor(6,0);

  lcd.print(rtc.getDateStr());

  lcd.setCursor(6,1);

  lcd.print(rtc.getTimeStr());

}

/////////////////////////////////////////////////////////////////

void data_logger(){

  if(sd == 1){

    Serial1.flush();

    // open the file. note that only one file can be open at a time,

    // so you have to close this one before opening another.

    myFile = SD.open("test.txt", FILE_WRITE);

    // if the file opened okay, write to it:

    if (myFile && logger==1) {

      lcd.clear();

      Serial.println("Proses menyimpan data ke file test.txt...");

      lcd.setCursor(0,0);

      lcd.print("Saving Data");

```

```
lcd.setCursor(0,1);

lcd.print("To SD Card");

delay(3000);

Serial1.print('Q');

delay(3000);

logger = 0;

delay(500);

myFile.print("Date : "); // tulis disini!!!!

myFile.print(dd);

myFile.print(" . ");

myFile.print(mth);

myFile.print(" . ");

myFile.println(yy);

myFile.print("Time : ");

myFile.print(hh);

myFile.print(" : ");

myFile.print(mm);

myFile.print(" : ");

myFile.println(ss);

myFile.println("-----");

myFile.println("Soil Moisture Sensor :");
```

```
myFile.print("S1: ");
myFile.println(S1);
myFile.print("S2: ");
myFile.println(S2);
myFile.print("S3: ");
myFile.println(S3);
myFile.print("S4: ");
myFile.println(S4);
myFile.print("S5: ");
myFile.println(S5);
myFile.print("S6: ");
myFile.println(S6);
myFile.print("Averages Of Soil Moisture: ");
myFile.println(SR);
myFile.println("-----");
myFile.print("Temp : ");
myFile.print(temp);
myFile.println("C");
myFile.print("Humidity : ");
myFile.print(h);
myFile.println("% RH");
```

```
// close the file:

myFile.close();

lcd.clear();

Serial.println("Process Complete!");

lcd.setCursor(0,0);

lcd.print("Process Complete!.");

}

else if (myFile && logger==0){

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Data Hasnt Taken");

    delay(3000);

    Serial.println("Data hasn't taken! ");

    Serial1.print('P'); // Apabila data di memory masih kosong

    delay(3000);

}

else {

    // if the file didn't open, print an error:

    lcd.clear();

    Serial.println("error opening test.txt");

    lcd.setCursor(0,0);

    lcd.print("Data Error!");
```

```
    delay(1000);

}

sd = 0;

lcd.clear();

}

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void serial(){

    Serial1.flush();

    quit = 0;

    d = 0;

    awal:

    if(Serial1.available()){

        awal1:

        Time t;

        c = Serial1.read();

        Serial.println(c);

        lcd.clear();

        lcd.setCursor(0,0);

        lcd.print("Progress Running");

        lcd.setCursor(2,1);

        lcd.print("Loading");
```

```
soil();

suhu();

t = rtc.getTime();

for(a=9;a<=13;a++){

    lcd.setCursor(a,1);

    lcd.print(".");

    delay(70);

}

if(c == 'A')

{

    d = 1;

    Serial1.print(S1);

    digitalWrite(13, HIGH);

    delay(50);

    digitalWrite(13, LOW);

    Serial1.print('A');

    Serial.println("Done!");

}

else if(c == 'B')

{

    d = 2;

    Serial1.print(S2);
```

```
digitalWrite(13, HIGH);  
  
delay(50);  
  
digitalWrite(13, LOW);  
  
Serial1.print('B');  
  
Serial.println("Done!");  
  
}  
  
else if(c == 'C')  
  
{  
  
d = 3;  
  
Serial1.print(S3);  
  
digitalWrite(13, HIGH);  
  
delay(50);  
  
digitalWrite(13, LOW);  
  
Serial1.print('C');  
  
Serial.println("Done!");  
  
}  
  
else if(c == 'D')  
  
{  
  
d = 4;  
  
Serial1.print(S4);  
  
digitalWrite(13, HIGH);  
  
delay(50);
```



```
digitalWrite(13, LOW);  
  
Serial1.print('D');  
  
Serial.println("Done!");  
  
}  
  
else if(c == 'E')  
  
{  
  
d = 5;  
  
Serial1.print(S5);  
  
digitalWrite(13, HIGH);  
  
delay(50);  
  
digitalWrite(13, LOW);  
  
Serial1.print('E');  
  
Serial.println("Done!");  
  
}  
  
else if(c == 'F')  
  
{  
  
d = 6;  
  
Serial1.print(S6);  
  
digitalWrite(13, HIGH);  
  
delay(50);  
  
digitalWrite(13, LOW);  
  
Serial1.print('F');
```

```
    Serial.println("Done!");  
}  
else if(c == 'G')  
{  
    d = 7;  
    Serial1.print(hh = rtc.getTime().hour);  
    digitalWrite(13, HIGH);  
    delay(50);  
    digitalWrite(13, LOW);  
    Serial1.print('G');  
    Serial.println("Done!");  
}  
else if(c == 'H')  
{  
    d = 8;  
    Serial1.print(mm = rtc.getTime().min);  
    digitalWrite(13, HIGH);  
    delay(50);  
    digitalWrite(13, LOW);  
    Serial1.print('H');  
    Serial.println("Done!");  
}
```

```
else if(c == 'I')
{
    d = 9;

    Serial1.print(ss = rtc.getTime().sec);

    digitalWrite(13, HIGH);

    delay(50);

    digitalWrite(13, LOW);

    Serial1.print('I');
}

else if(c == 'J')
{
    d = 10;

    Serial1.print(dd = t.date);

    Serial.println(dd);

    digitalWrite(13, HIGH);

    delay(50);

    digitalWrite(13, LOW);

    Serial1.print('J');
}

else if(c == 'K')
{
    d = 11;
```

```
bulan = String(rtc.getMonthStr());

if(bulan == "January"){mth = 1;}

if(bulan == "February"){mth = 2;}

if(bulan == "Maret"){mth = 3;}

if(bulan == "April"){mth = 4;}

if(bulan == "May"){mth = 5;}

if(bulan == "June"){mth = 6;}

if(bulan == "July"){mth = 7;}

if(bulan == "August"){mth = 8;}

if(bulan == "September"){mth = 9;}

if(bulan == "October"){mth = 10;}

if(bulan == "November"){mth = 11;}

if(bulan == "December"){mth = 12;}

Serial1.print(mth);

Serial.println(bulan);

Serial.println(mth);

digitalWrite(13, HIGH);

delay(50);

digitalWrite(13, LOW);

Serial1.print('K');

}
```

```
else if(c == 'L')
{
    d = 12;
    Serial1.print(yy = t.year);
    Serial.println(yy);
    digitalWrite(13, HIGH);
    delay(50);
    digitalWrite(13, LOW);
    Serial1.print('L');
}
else if(c == 'M') // SUHU
{
    d = 13;
    Serial1.print(temp);
    Serial.println(temp);
    digitalWrite(13, HIGH);
    delay(50);
    digitalWrite(13, LOW);
    Serial1.print('M');
}
else if(c == 'N') // HUMIDITY
{
```

```
d = 14;

logger = 1;

Serial1.print(h);

Serial.println(h);

digitalWrite(13, HIGH);

delay(50);

digitalWrite(13, LOW);

// Serial1.print('N');

}

else if(c == 'O') // DATA LOGGER

{

quit = 1;

sd = 1;

Serial1.flush();

digitalWrite(13, HIGH);

delay(500);

digitalWrite(13, LOW);

}

else if(c == 'V') //RESET MEGA FROM ESP

{

pinMode(10, OUTPUT);

digitalWrite(10, LOW);
```

```
    delay(5);

    digitalWrite(10, HIGH);

    pinMode(10, INPUT);
}

else if(c == '1') // AUTO MODE
{
    quit = 1;

    digitalWrite(13, HIGH);

    lcd.clear();

    lcd.setCursor(3,0);

    lcd.print("Auto Mode!");

    mode = 0;

    pumpon = 0;

    pumpoff = 0;

    delay(500);

    digitalWrite(13, LOW);
}

else if(c == '2') // MANUAL MODE
{
    quit = 1;

    mode = 1;

    digitalWrite(13, HIGH);
```

```
lcd.clear();

lcd.setCursor(2,0);

lcd.print("Manual Mode!");

delay(500);

digitalWrite(13, LOW);

}

else if(c == '3') // Pump On

{

    quit = 1;

    digitalWrite(13, HIGH);

    pompa = 1;

    mode = 1;

    delay(500);

    digitalWrite(13, LOW);

}

else if(c == '4') // Pump Off

{

    quit = 1;

    digitalWrite(13, HIGH);

    pompa = 0;

    mode = 1;

    delay(500);
```



```

    digitalWrite(13, LOW);

}

lcd.clear();

Serial.println(d);

delay(100);

Serial.println("quit: " + String(quit));

if(d == 14 || !Serial1.available()){goto awal;}

else if(quit == 1 || !Serial1.available()){goto awal;}

goto awal;

}

else if(!Serial1){;}

}

////////////////////////////////////

void siram(){

soil();

serial();

hh = rtc.getTime().hour;

mm = rtc.getTime().min;

ss = rtc.getTime().sec;

tw = mm%5;

if(hh >= ws_min_pagi && hh <= ws_max_pagi ){ws=1;}

else if(hh >= ws_min_sore && hh <= ws_max_sore){ws=2;}

```

```

else{ws=0;}

if(ws == 1 && SR < kelembaban_ideal && mode == 0){

  if(pompa == 0)Serial1.println('S');

  pompa = 1;

  serial();

  digitalWrite(pump, LOW);

  Serial.println("Pump On Auto 1!");

  if (SR > kelembaban_mati_pompa || hh > ws_max_pagi){digitalWrite(pump,
HIGH);mode=0;pompa=0;delay(100);}

  if(tw==0&&treq==0&&mode==0){Serial1.print('R');Serial.println("treq!");
treq++;delay(1000);} // interval auto mode

  else if (tw == 1){treq = 0;}

  delay(10);

}

else if(ws == 2 && SR < kelembaban_ideal && mode == 0){

  if(pompa == 0)Serial1.println('S');

  pompa = 1;

  serial();

  digitalWrite(pump, LOW);

  Serial.println("Pump On Auto 2!");

  if (SR > kelembaban_mati_pompa || hh > ws_max_sore){digitalWrite(pump,
HIGH);mode=0;pompa=0;delay(100);}

  if(tw==0&&treq==0&&mode==0){

```

```
Serial1.print('R');Serial.println("treq!");treq++;delay(1000);} // interval auto
mode

else if (tw == 1){treq = 0;}

delay(10);

}

else if(pompa == 0 && mode == 0 && SR > kelembaban_ideal){ // OFF AUTO

if(pompa == 1)Serial1.println("T");

pompa = 0;

serial();

digitalWrite(pump, HIGH);

Serial.println("Pump Off Auto!");

if(tw==0&&treq==0&&mode==0){

Serial1.print('R');Serial.println("treq!");treq ++;delay(1000);} // interval auto
mode

if (tw == 1){treq = 0;}

delay(10);

}

else if(pompa == 1 && mode == 1) { // ON MANUAL

digitalWrite(pump, HIGH);

Serial.println("Pump On Man!");

serial();

delay(10);

}
```

```

else if(pompa == 0 && mode == 1) { // OFF MANUAL

    digitalWrite(pump, HIGH);

    Serial.println("Pump Off Man!");

    serial();

    delay(10);

}

else if(tw==0&&treq==0&&mode==0){

Serial1.print('R');Serial.println("treq!");treq++;delay(1000);} // interval auto
mode

else if (tw == 1){treq = 0;}

}

////////////////////////////////////

void setup() {

    // set up the LCD's number of columns and rows:

    lcd.begin(16, 2);

    rtc.begin();

    dht.begin();

    Serial.begin(9600);

    Serial1.begin(9600);

    pinMode(13, OUTPUT);

    pinMode(pump, OUTPUT);

    pinMode(pump1, OUTPUT);

```

```
digitalWrite(pump, HIGH);  
digitalWrite(pump1, HIGH);  
digitalWrite(10, HIGH);  
////////////////////////////////////
```

awal:

```
delay(1000);  
if (!SD.begin(53) && logger==0) {  
    lcd.clear();  
    Serial.println("Inisialisasi gagal!");  
    lcd.setCursor(0,0);  
    lcd.print("Initialize");  
    lcd.setCursor(0,1);  
    lcd.print("Failed!");  
    Serial1.print('O');  
    delay(1000);  
    goto awal;  
}  
else {  
    lcd.clear();  
    Serial.println("Inisialisasi berhasil!");
```

```

    lcd.setCursor(0,0);

    lcd.print("Initialize");

    lcd.setCursor(0,1);

    lcd.print("Done!");

    delay(1000);

}

// rtc.setDOW(MONDAY); // Set Day-of-Week to THURSDAY

// rtc.setTime(18, 57, 0); // Set the time to 12:00:00 (24hr format)

// rtc.setDate(12, 7, 2018); // Set the date to January 1st, 2014

}

void loop() {

    lcd.clear();

    siram();

    serial();

    data_logger();

    for(v=0;v<=500;v++){ waktu();siram();serial();data_logger();delay(1);}

    for(w=0;w<=7;w++){ soil;tampil_soil();siram();serial();data_logger();delay(1);}

    for(x=0;x<=7;x++){ soil;tampil_soil2();siram();serial();data_logger();delay(1);}

    for(y=0;y<=7;y++){ soil;tampil_soil3();siram();serial();data_logger();delay(1);}

    for(z=0;z<=10;z++){ suhu();tampil_suhu();siram();serial();data_logger();delay();}

}

}

```

LAMPIRAN 3

**Program NodeMCU ESP8266 12E: Sistem *Monitoring* Kebun
Sawi *Via Telegram***

```

#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <SoftwareSerial.h>
#include "ThingSpeak.h"
SoftwareSerial esp(14,12); //rx,tx || d5,d6

// Initialize Wifi connection to the router
char ssid[] = "Afif.tpy"; // your network SSID (name)
char password[] = "Since1998"; // your network key

// replace with your channel's thingspeak API key
unsigned long myChannelNumber = 412985;
const char * myWriteAPIKey = "66TKCZMVMWZ70P5";

// Initialize Telegram BOT
#define BOTtoken
"528031075:AAE3dkixOM5AwTcVYiQYi3pwZgwWysRhDUc" // your Bot
WiFiClient client1;
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

int Bot_mtbs = 10; //mean time between scan messages
long Bot_lasttime; //last time messages' scan has been done

int S1, S2, S3, S4, S5, S6, SR, hh, mm, ss, dd, mth, yy, treq=0;
byte i=0, pesan=0, rst=0, pumpon=0, pumpoff=0, kondisi=0, user=0;
float h, temp;
char c;

```



```
void setup(){
  Serial.begin(9600);
  esp.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  Serial.print("Connecting Wifi: ");
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    digitalWrite(LED_BUILTIN, LOW);
    delay(500);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(500);
  }

  Serial.println("");
  Serial.println("WiFi connected to:");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  esp.println('V');
  delay(2000);
  ThingSpeak.begin(client1);
  digitalWrite(LED_BUILTIN,HIGH);
}
```

```

void thinkspeak(){
  int respon = ThingSpeak.writeField(myChannelNumber, 1, SR,
  myWriteAPIKey);
  Serial.print("Respon : ");
  Serial.println(respon);
  Serial.println("\n");
  delay(10000);
  int respon1 = ThingSpeak.writeField(myChannelNumber, 2, temp,
  myWriteAPIKey);
  Serial.print("Respon1 : ");
  Serial.println(respon1);
  Serial.println("\n");
  delay(10000);
  int respon2 = ThingSpeak.writeField(myChannelNumber, 3, h, myWriteAPIKey);
  Serial.print("Respon2 : ");
  Serial.println(respon2);
  Serial.println("\n");
}

```

```

void handleNewMessages(int numNewMessages){
  Serial.println("handleNewMessages");
  Serial.println(numNewMessages);
  digitalWrite(LED_BUILTIN,HIGH);

  String chat_id = String(bot.messages[i].chat_id);
  String text = bot.messages[i].text;
  String from_name = bot.messages[i].from_name;
  if (from_name == "") from_name = "Guest";
  if (from_name == "AFIF" || from_name == "Admin" ) user = 1;
  else if(from_name != "AFIF" && pesan == 0){
    pesan = 1;

```

```

user = 0;
numNewMessages = 0;
digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai
bot.sendChatAction(chat_id,"typing");
String welcome = "Unknown User!\n";
bot.sendMessage(chat_id, welcome);
Serial.println("Message sent!");
digitalWrite(LED_BUILTIN, HIGH); // Inisial Proses Selesai
}

////////////////////////////////////// START ////////////////////////////////////////

if (text == "/start" && pesan == 0 && user == 1){
  bot.sendChatAction(chat_id,"typing");
  pesan = 1;
  numNewMessages = 0;
  kondisi = 0;
  digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai
  String welcome = "Welcome !\n";
  welcome += "Select Mode Please:";
  String keyboardJson = "[[\"/Manual\"], [\"/Auto\"]]";
  bot.sendMessageWithReplyKeyboard(chat_id, welcome, "", keyboardJson,
  true);
  Serial.println("Message sent!");
  digitalWrite(LED_BUILTIN, HIGH); // Inisial Proses Selesai
}

else if (text == "/Manual" && pesan == 0 && user == 1){
  bot.sendChatAction(chat_id,"typing");
  pesan = 1;
  numNewMessages = 0;
  digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai

```

```

kondisi = 2; // Manual Mode
delay(100);
esp.println('2');
delay(200);
String welcome = "Manual Mode Selected!\n";
welcome += "What's Your Command?";
String keyboardJson = "[\"/Pump On\", \"/Pump Off\"], [\"/Status\", \"/Save
Data\"], [\"/Auto\"]";
bot.sendMessageWithReplyKeyboard(chat_id, welcome, "", keyboardJson,
true);
Serial.println("Message sent!");
digitalWrite(LED_BUILTIN, HIGH); // Inisial Proses Selesai
}

else if (text == "/Auto" && pesan == 0 && user == 1){
bot.sendChatAction(chat_id,"typing");
pesan = 1;
numNewMessages = 0;
digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai
kondisi = 1; // Auto Mode
delay(100);
esp.println('1');
delay(200);
String welcome = "Auto Mode Selected!\n";
welcome += "Data Recorded Send Every 10 Minutes to :\n";
welcome += "https://thingspeak.com/channels/412985";
String keyboardJson = "[\"/Status\"], [\"/Manual\"]";
bot.sendMessageWithReplyKeyboard(chat_id, welcome, "", keyboardJson,
true); Serial.println("Message sent!");
digitalWrite(LED_BUILTIN, HIGH); // Inisial Proses Selesai
}

```

```
////////////////////////////////// AUTO MODE ////////////////////////////////////
```

```
else if (kondisi != 2 && treq == 1){ // Masuk Apabila Telah Mode Auto
    esp.flush();
    bot.sendChatAction(chat_id,"typing");
    pesan = 1;
    numNewMessages = 0;
    digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai
    delay(1000);
    esp.println('A');
    delay(1000);
    S1 = esp.parseInt();
    c = esp.read();
    Serial.println(c);
    if(c == 'A')
    {
        esp.println('B');
        delay(1000);
        S2 = esp.parseInt();
    }
    c = esp.read();
    Serial.println(c);
    if(c == 'B')
    {
        esp.println('C');
        delay(1000);
        S3 = esp.parseInt();
    }
    c = esp.read();
    Serial.println(c);
```

```
if(c == 'C')
{
    esp.println('D');
    delay(1000);
    S4 = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'D')
{
    esp.println('E');
    delay(1000);
    S5 = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'E')
{
    esp.println('F');
    delay(1000);
    S6 = esp.parseInt();
}
// Jam //
c = esp.read();
Serial.println(c);
if(c == 'F')
{
    esp.println('G');
    delay(1000);
    hh = esp.parseInt();
}
```

```
c = esp.read();
Serial.println(c);
if(c == 'G')
{
    esp.println('H');
    delay(1000);
    mm = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'H')
{
    esp.println('I');
    delay(1000);
    ss = esp.parseInt();
}
// Tanggal //
c = esp.read();
Serial.println(c);
if(c == 'I')
{
    esp.println('J');
    delay(1000);
    dd = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'J')
{
    esp.println('K');
    delay(1000);
```

```
    mth = esp.parseInt();
  }
  c = esp.read();
  Serial.println(c);
  if(c == 'K')
  {
    esp.println('L');
    delay(1000);
    yy = esp.parseInt();
  }
  // DHT 11 //
  c = esp.read();
  Serial.println(c);
  if(c == 'L') // SUHU
  {
    esp.println('M');
    delay(1000);
    temp = esp.parseFloat();
  }
  c = esp.read();
  Serial.println(c);
  if(c == 'M')
  {
    esp.println('N');
    delay(1000);
    h = esp.parseFloat();
    delay(100);
    esp.println('O');
  }
  delay(3000);
```



```

SR = (S1+S2+S3+S4+S5+S6)/6;
treq = 0;
thinkspeak();
Serial.println("Data sensor sent!");
digitalWrite(LED_BUILTIN, HIGH); // Inisial Proses Selesai
}

else if (text == "/Status" && kondisi == 1 && pesan == 0 && user == 1){
  esp.flush();
  bot.sendChatAction(chat_id,"typing");
  pesan = 1;
  numNewMessages = 0;
  digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai
  delay(1000);
  esp.println('A');
  delay(1000);
  S1 = esp.parseInt();
  c = esp.read();
  Serial.println(c);
  if(c == 'A')
  {
    esp.println('B');
    delay(1000);
    S2 = esp.parseInt();
  }
  c = esp.read();
  Serial.println(c);
  if(c == 'B')
  {
    esp.println('C');
    delay(1000);
  }
}

```

```
S3 = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'C')
{
    esp.println('D');
    delay(1000);
    S4 = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'D')
{
    esp.println('E');
    delay(1000);
    S5 = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'E')
{
    esp.println('F');
    delay(1000);
    S6 = esp.parseInt();
}
// Jam //
c = esp.read();
Serial.println(c);
if(c == 'F')
{
```

```
    esp.println('G');
    delay(1000);
    hh = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'G')
{
    esp.println('H');
    delay(1000);
    mm = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'H')
{
    esp.println('I');
    delay(1000);
    ss = esp.parseInt();
}
// Tanggal //
c = esp.read();
Serial.println(c);
if(c == 'I')
{
    esp.println('J');
    delay(1000);
    dd = esp.parseInt();
}
c = esp.read();
Serial.println(c);
```

```
if(c == 'J')
{
    esp.println('K');
    delay(1000);
    mth = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'K')
{
    esp.println('L');
    delay(1000);
    yy = esp.parseInt();
}
// DHT 11 //
c = esp.read();
Serial.println(c);
if(c == 'L') // SUHU
{
    esp.println('M');
    delay(1000);
    temp = esp.parseFloat();
}
c = esp.read();
Serial.println(c);
if(c == 'M')
{
    esp.println('N');
    delay(1000);
    h = esp.parseFloat();
    delay(100);
```

```
}  
delay(3000);  
SR = (S1+S2+S3+S4+S5+S6)/6;
```

```
String welcome = "Date : \t";  
welcome += dd;  
welcome += " . ";  
welcome += mth;  
welcome += " . ";  
welcome += yy;
```

```
welcome += "\nTime : \t";  
welcome += hh;  
welcome += " : ";  
welcome += mm;  
welcome += " : ";  
welcome += ss;
```

```
welcome += "\n-----\n";  
welcome += "Soil Moisture Sensor :\n";  
welcome += "S1: ";  
welcome += S1;  
welcome += " %";  
welcome += "\nS2: ";  
welcome += S2;  
welcome += " %";  
welcome += "\nS3: ";  
welcome += S3;  
welcome += " %";  
welcome += "\nS4: ";  
welcome += S4;
```

```

welcome += " %";
welcome += "\nS5: ";
welcome += S5;
welcome += " %";
welcome += "\nS6: ";
welcome += S6;
welcome += " %";
welcome += "\nAverages Of Soil Moisture: \t";
welcome += SR;
welcome += " %";
welcome += "\n-----\n";

welcome += "Temp : \t";
welcome += temp;
welcome += " 'C";
welcome += "\nHumidity : \t";
welcome += h;
welcome += " %RH";
welcome += "\n-----\n";
delay(100);
String keyboardJson = "[[{\ \"text\" : \"Let see the Graph!\", \"url\" :
  \"/https://thingspeak.com/channels/412985\" } ]]";
bot.sendMessageWithInlineKeyboard(chat_id, welcome, "", keyboardJson);
Serial.println("Data sensor sent!");
digitalWrite(LED_BUILTIN, HIGH); // Inisial Proses Selesai
}

////////////////////// MANUAL MODE ////////////////////////////////////////
else if (text == "/Pump On" && kondisi == 2 && pesan == 0 && user == 1){
  bot.sendChatAction(chat_id, "typing");
  pesan = 1;

```

```

numNewMessages = 0;
digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai
esp.println('3');
delay(500);
String welcome = "Pump Is On!";
String keyboardJson = "[\"/Pump On\", \"/Pump Off\"], [\"/Status\", \"/Save
Data\"], [\"/Auto\"]";
bot.sendMessageWithReplyKeyboard(chat_id, welcome, "", keyboardJson,
true);
Serial.println("Message sent!");
digitalWrite(LED_BUILTIN, HIGH); // Inisial Proses Selesai
}

```

```

else if (text == "/Pump Off" && kondisi == 2 && pesan == 0 && user == 1){
bot.sendChatAction(chat_id,"typing");
pesan = 1;
numNewMessages = 0;
digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai
esp.println('4');
delay(500);
String welcome = "Pump Is Off!";
String keyboardJson = "[\"/Pump On\", \"/Pump Off\"], [\"/Status\", \"/Save
Data\"], [\"/Auto\"]";
bot.sendMessageWithReplyKeyboard(chat_id, welcome, "", keyboardJson,
true);
Serial.println("Message sent!");
digitalWrite(LED_BUILTIN, HIGH); // Inisial Proses Selesai
}

```

```

else if (text == "/Save Data" && kondisi == 2 && pesan == 0 && user == 1){
// Manual

```

```

esp.flush();
pesan = 1;
numNewMessages = 0;
digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai
bot.sendChatAction(chat_id,"typing");
delay(1000);
esp.println('O');
delay(7000);

if(esp.available()){
  c = esp.read();
  delay(300);
}

Serial.println("C: "+String(c));
String welcome = "Data Logger: ";
if(c == 'O'){welcome += "Data Unsuccessfully Save!\n";}
else if(c == 'P'){welcome += "Data Hasn't Taken!\n";}
else if(c == 'Q') {welcome += "Data Successfully Save!\n";thinkSpeak();}
String keyboardJson = "[[{\ \"text\" : \"Let see the Graph!\", \"url\" :
  \"/>

```



```
digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai
delay(1000);
esp.println('A');
delay(1000);
S1 = esp.parseInt();
c = esp.read();
Serial.println(c);
if(c == 'A')
{
    esp.println('B');
    delay(1000);
    S2 = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'B')
{
    esp.println('C');
    delay(1000);
    S3 = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'C')
{
    esp.println('D');
    delay(1000);
    S4 = esp.parseInt();
}
c = esp.read();
Serial.println(c);
```

```
if(c == 'D')
{
    esp.println('E');
    delay(1000);
    S5 = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'E')
{
    esp.println('F');
    delay(1000);
    S6 = esp.parseInt();
}
// Jam //
c = esp.read();
Serial.println(c);
if(c == 'F')
{
    esp.println('G');
    delay(1000);
    hh = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'G')
{
    esp.println('H');
    delay(1000);
    mm = esp.parseInt();
}
```

```
c = esp.read();
Serial.println(c);
if(c == 'H')
{
    esp.println('I');
    delay(1000);
    ss = esp.parseInt();
}
// Tanggal //
c = esp.read();
Serial.println(c);
if(c == 'I')
{
    esp.println('J');
    delay(1000);
    dd = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'J')
{
    esp.println('K');
    delay(1000);
    mth = esp.parseInt();
}
c = esp.read();
Serial.println(c);
if(c == 'K')
{
    esp.println('L');
    delay(1000);
```

```

    yy = esp.parseInt();
}
// DHT 11 //
c = esp.read();
Serial.println(c);
if(c == 'L') // SUHU
{
    esp.println('M');
    delay(1000);
    temp = esp.parseFloat();
}
c = esp.read();
Serial.println(c);
if(c == 'M')
{
    esp.println('N');
    delay(1000);
    h = esp.parseFloat();
    delay(100);
}
delay(3000);
SR = (S1+S2+S3+S4+S5+S6)/6;

```

```

String welcome = "Date : \t";
welcome += dd;
welcome += " . ";
welcome += mth;
welcome += " . ";
welcome += yy;

welcome += "\nTime : \t";

```

```
welcome += hh;  
welcome += " : ";  
welcome += mm;  
welcome += " : ";  
welcome += ss;
```

```
welcome += "\n-----\n";  
welcome += "Soil Moisture Sensor :\n";  
welcome += "S1: ";  
welcome += S1;  
welcome += " %";  
welcome += "\nS2: ";  
welcome += S2;  
welcome += " %";  
welcome += "\nS3: ";  
welcome += S3;  
welcome += " %";  
welcome += "\nS4: ";  
welcome += S4;  
welcome += " %";  
welcome += "\nS5: ";  
welcome += S5;  
welcome += " %";  
welcome += "\nS6: ";  
welcome += S6;  
welcome += " %";  
welcome += "\nAverages Of Soil Moisture : \t";  
welcome += SR;  
welcome += " %";  
welcome += "\n-----\n";
```

```

welcome += "Temp : \t";
welcome += temp;
welcome += " 'C";
welcome += "\nHumidity : \t";
welcome += h;
welcome += " %RH";
welcome += "\n-----\n";

String keyboardJson = "[\"/Pump On\", \"/Pump Off\"], [\"/Status\", \"/Save
Data\"], [\"/Auto\"]";
bot.sendMessageWithReplyKeyboard(chat_id, welcome, "", keyboardJson,
true);
Serial.println("Data sensor sent!");
digitalWrite(LED_BUILTIN, HIGH); // Inisial Proses Selesai
}
else if(pesan == 0 && user == 1){
    pesan = 1;
    numNewMessages = 0;
    digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai
    bot.sendChatAction(chat_id,"typing");
    if(kondisi == 0){
        String welcome = "Please Select Mode First.\n";
        welcome += "Auto Mode Selected As Default Mode.";
        String keyboardJson = "[\"/Auto\"], [\"/Manual\"]";
        bot.sendMessageWithReplyKeyboard(chat_id, welcome, "", keyboardJson,
true);
    }
    else {
        String welcome = "Unknown Message!\n";
        bot.sendMessage(chat_id, welcome);
    }
}

```

```

Serial.println("Message sent!");
digitalWrite(LED_BUILTIN, HIGH); // Inisial Proses Selesai
}
else if(kondisi == 1 && pumpon == 1 || pumpoff == 1 && user == 1){
    pesan = 1;
    numNewMessages = 0;
    digitalWrite(LED_BUILTIN, LOW); // Proses Dimulai
    bot.sendChatAction(chat_id,"typing");
    String welcome = "Condition: ";
    if(pumpon == 1){ welcome = "Pump is On Automatically!\n";}
    else if(pumpoff == 1){ welcome = "Pump is Off Automatically!\n";}
    pumpon = 0;
    pumpoff = 0;
    String keyboardJson = "[[\"/Status\"], [\"/Manual\"]]";
    bot.sendMessageWithReplyKeyboard(chat_id, welcome, "", keyboardJson,
    true);
    Serial.println("Message sent!");
    digitalWrite(LED_BUILTIN, HIGH); // Inisial Proses Selesai
}
}

void loop() {
    if(esp.available() > 0){
        c = esp.read();
        if(c == 'R' && kondisi != 2)
        {
            treq = 1;
            Serial.println("treq:"+String(treq));
            delay(100);
        }
    }
}

```

```

else if(c == 'S' && kondisi != 2)
{
    pumpon = 1;
    Serial.println("pumpon:"+String(pumpon));
    delay(100);
}
else if(c == 'T' && kondisi != 2)
{
    pumpoff = 1;
    Serial.println("pumpoff:"+String(pumpoff));
    delay(100);
}
}

int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
handleNewMessages(numNewMessages);
if(numNewMessages>=1&&pesan>=1){ numNewMessages=0;pesan=0;delay(
100);}
Serial.println("pesan: " + String(pesan));
Serial.println("kondisi: " + String(kondisi));
Serial.println("treq: " + String(treq));
Serial.println("numNewMessages: " + String(numNewMessages));
Bot_lasttime = millis();
}

```


LAMPIRAN 4

Spesifikasi Alat Ukur Suhu dan Kelembaban Udara :

Hygrometer HTC-2

Spesifikasi :

1. *Temperature range: -50C ~ 70C (-58F ~ 158F)*
2. *Humidity range: 10% RH ~ 95% RH*
3. *Accuracy: 1C (or 2F), 5% RH*
4. *Display comfort level as comfortable, wet or dry*
5. *Recording MAX/MIN temperature and humidity value automatically*
6. *Dimension: 110 x100 x 21mm*
7. *Power: 1 x AAA Battery (termasuk)*
8. *Dual Temperature:*
 - *INDOOR* dengan sensor di unit
 - *OUTDOOR* dengan sensor di kabel (kabel bisa juga dicabut)

LAMPIRAN 5

Spesifikasi Alat Ukur Kelembaban Tanah:

Tester Soil Water 3 in 1

Spesifikasi :

1. *3 functions in one compact meter - measure the moisture and pH content of the soil and the environmental light*
2. *Scientifically accurate*
3. *No need for batteries*
4. *Easy-to-read scale indicator*
5. *Works with house and garden plants*
6. *Dimension: 5 x 8 x 3.5cm [2 x 3.2 x 1.5"]*
7. *Probe length: 21cm [8"]*