



**RANCANG BANGUN PEMANTAUAN DAN PENJADWALAN ALAT
PEMBERI PAKAN IKAN OTOMATIS SECARA JARAK JAUH**



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

Oleh :

YOYOK SETIAWAN

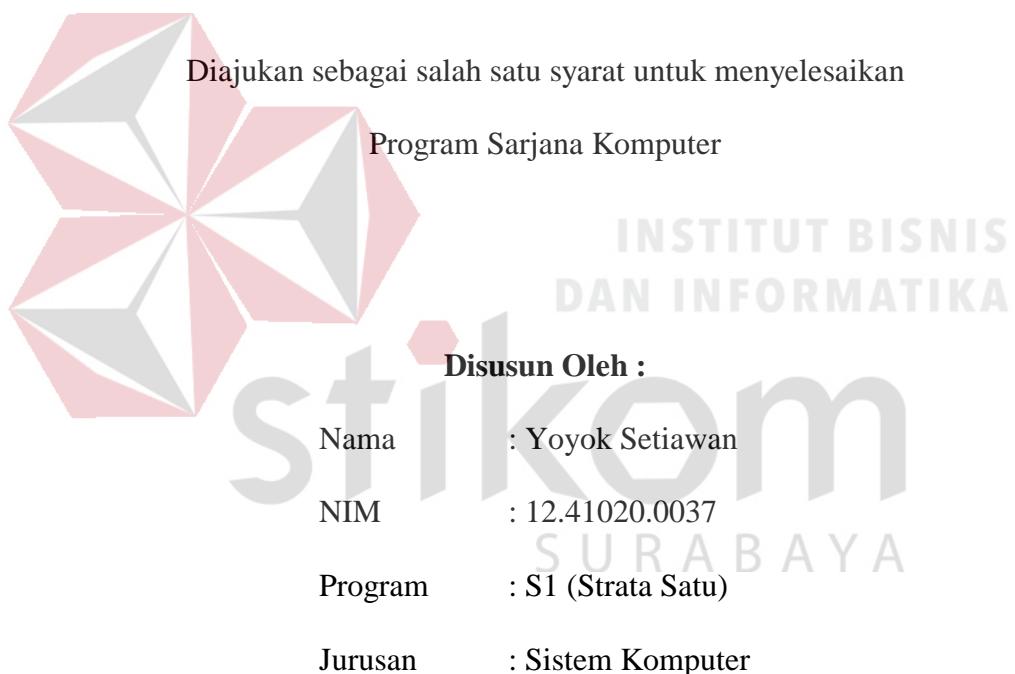
12.41020.0037

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA
2017**

RANCANG BANGUN PEMANTAUAN DAN PENJADWALAN ALAT

PEMBERI PAKAN IKAN OTOMATIS SECARA JARAK JAUH

TUGAS AKHIR



FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOMSURABAYA

2017

Janganlah berusaha melampaui kelebihan orang lain, tetapi berusahalah untuk melampaui kekurangan diri sendiri.



Puji Dan Syukur Bagi Allah SWT,
Penulis Dapat Menyelesaikan Tugas Akhir Ini Dengan Baik.



Tugas Akhir Ini Saya Persembahkan Kepada
Kedua Orang Tua, Istri Dan Anak Saya Tercinta
Serta Tidak Lupa Juga Semua Keluarga.
Terimakasih Kepada Dosen-Dosen Pembimbing
Serta Semua Rekan-Rekan Di Sistem Komputer Dan Dikampus Institut Bisnis
Dan Informatika Stikom Surabaya
Beserta Semua Orang – Orang Yang Turut Membantu Saya Selama Ini.

STIKOM
SURABAYA

TUGAS AKHIR
RANCANG BANGUN PEMANTAUAN DAN PENJADWALAN ALAT
PEMBERI PAKAN IKAN OTOMATIS SECARA JARAK JAUH

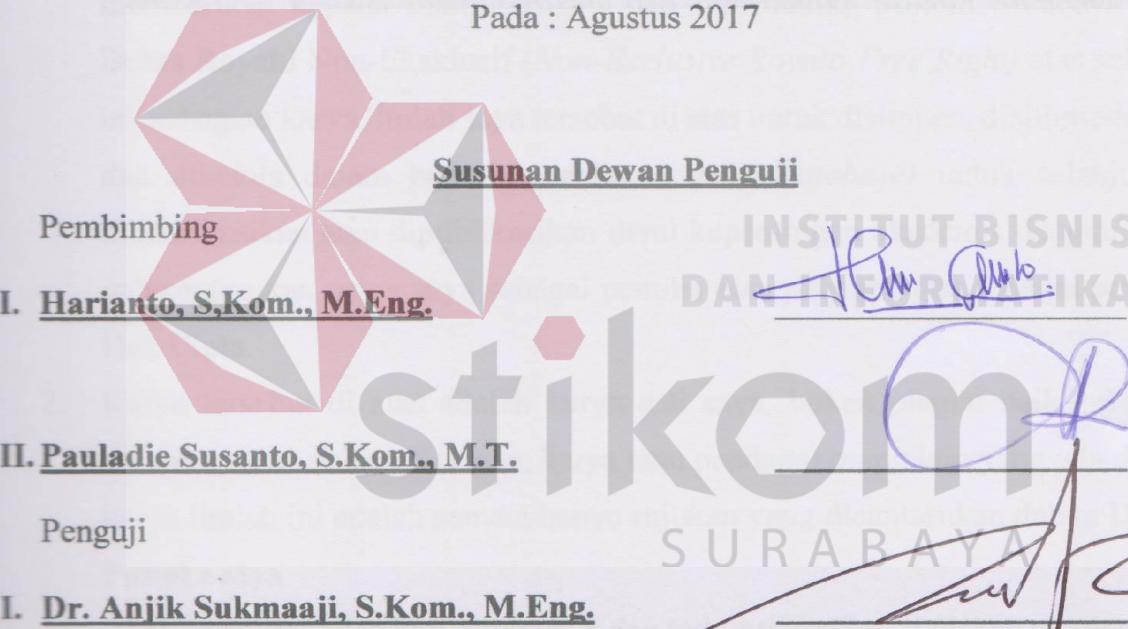
Dipersiapkan dan disusun oleh

Yoyok Setiawan

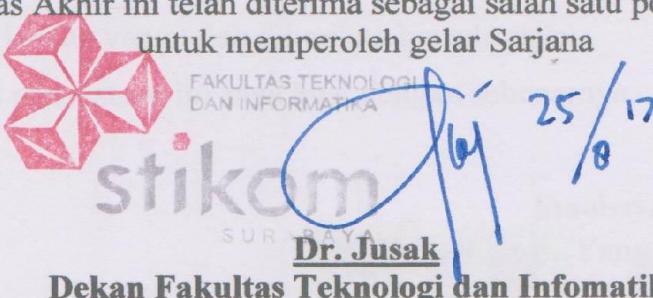
NIM : 12.41020.0037

Telah diperiksa, diuji dan disetujui oleh Dewan Pengaji

Pada : Agustus 2017



Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana



INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Yoyok Setiawan
NIM : 12.41020.0037
Program Studi : S1 Sistem Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **RANCANG BANGUN PEMANTAUAN DAN PENJADWALAN ALAT PEMBERI PAKAN IKAN OTOMATIS SECARA JARAK JAUH**

Menyatakan dengan sesungguhnya bahwa:

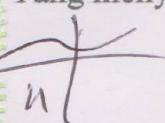
1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan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

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Agustus 2017

Yang menyatakan,




Yoyok Setiawan
NIM: 12.41020.0037

ABSTRAK

Saat ini, telah banyak organisasi maupun perorangan yang berusaha untuk memaksimalkan teknologi *Smartphone Android* dalam budidaya ikan. Salah satunya adalah Alat Pemberi Pakan Ikan Otomatis pada kolam ikan. Alat pemberi pakan ikan ini pada dasarnya sudah membantu pembudidaya dalam mengatasi masalah pemberian pakan yang dapat dilakukan secara otomatis oleh alat. Tetapi alat tersebut masih terkendala dalam hal pemantauan kerusakan, ketersediaan pakan dan penjadwalan pada alat, dimana harus dilakukan secara langsung pada alat.

Pada tugas akhir ini dirancang suatu sistem yang dapat memantau kerusakan dan ketersediaan pakan, serta dapat melakukan penjadwalan pada Alat Pemberi Pakan Ikan Otomatis secara jarak jauh. Perancangan menggunakan aplikasi *Smartphone Android* yang terintegrasi dengan *Arduino* dan *Module WiFi NodeMCU ESP8266* pada alat. Pengiriman data antara aplikasi *Smartphone Android* dengan alat dilakukan melalui *internet* dan *web server* sebagai perantara.

Berdasarkan pengujian yang dilakukan terdiri dari pemantauan kerusakan, pemantauan ketersediaan pakan, serta penjadwalan pada aplikasi *Android* maupun pada Alat Pemberi Pakan Ikan Otomatis. Rata-rata persentase tingkat keberhasilan $\pm 91.96\%$ dari beberapa pengujian yang dilakukan. Jaringan *internet* yang digunakan sangat berpengaruh terhadap pengiriman dan penerimaan data.

Keyword : *Module WiFi NodeMCU ESP8266, Alat Pemberi Pakan Ikan Otomatis, Android, Arduino.*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat yang telah diberikan - Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini. Penulisan Laporan ini adalah sebagai salah satu syarat Menempuh Tugas Akhir pada Program Studi S1 Sistem Komputer Institut Bisnis dan Informatika Stikom Surabaya.

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

1. **Orang Tua, Istri, Anak dan Saudara-saudara** saya tercinta yang telah memberikan dorongan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir maupun laporan ini.
2. Kepada **Harianto, S.Kom., M.Eng.** dan **Pauladie Susanto, S.Kom., M.T.** selaku Dosen Pembimbing. Terima kasih atas bimbingan yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik.
3. Kepada **Dr. Anjik Sukmaaji, S.Kom., M.Eng.** selaku Dosen Penguji dan Ketua Program Studi Sistem Komputer. Terima kasih atas masukan yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik serta terima kasih atas ijin yang diberikan untuk mengerjakan Tugas Akhir ini.
4. Semua staf Dosen yang telah mengajar dan memberikan ilmunya kepada penulis.

5. Seluruh saudara seperjuangan SK angkatan 2012, teman-teman SK dan alumni yang selalu memberikan semangat dan bantuannya.
6. Semua rekan-rekan yang ada di kampus Institut Bisnis Dan Informatika Stikom Surabaya dari semua jurusan.



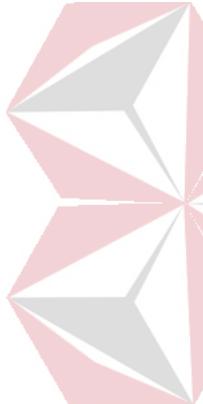
Surabaya, Agustus 2017

Penulis

DAFTAR ISI

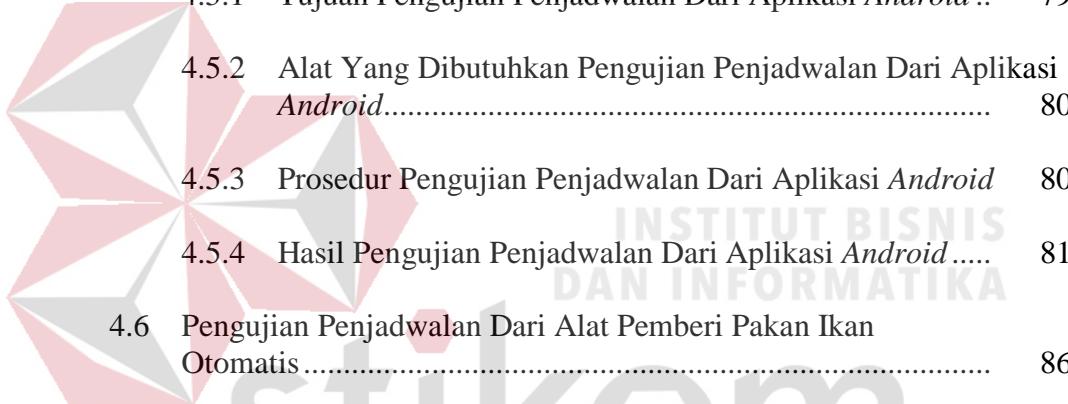
HALAMAN JUDUL.....	i
HALAMAN SYARAT	ii
MOTTO.....	iii
HALAMAN PERSEMBAHAN.....	iv
HALAMAN PENGESAHAN.....	v
HALAMAN PERNYATAAN	vi
ABSTRAK.....	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xx
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Pakan Ikan	6
2.2 <i>Arduino Mega 2560</i>	6
2.3 <i>Module WiFi NodeMCU ESP8266</i>	8

2.3.1	Konsep Dasar <i>NodeMCU</i>	8
2.3.2	Definisi <i>ESP8266</i>	10
2.4	<i>Internet</i>	11
2.5	<i>Modem</i>	13
2.6	<i>Web Server</i>	14
2.7	Motor DC	15
2.8	<i>Rotary</i>	16
2.9	Komunikasi <i>Serial</i>	18
2.10	<i>Arduino Software IDE</i>	19
2.11	<i>Android</i>	20
2.12	<i>APP Inventor</i>	20
BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM		22
3.1	Metode Penelitian	22
3.2	Model Perancangan	24
3.2.1	Perancangan Data Pada Sistem	26
3.2.2	Perancangan Proses Pemantauan Kerusakan	28
3.2.3	Perancangan Proses Pemantauan Ketersediaan Pakan ...	28
3.2.4	Perancangan Proses Penjadwalan Pemberian Pakan	29
3.2.5	Perancangan <i>Flowchart</i> Sistem Pada <i>Module WiFi NodeMCU ESP8266</i>	30
3.2.6	Perancangan <i>Flowchart</i> Sistem Pada <i>Arduino</i>	33
3.2.7	Perancangan <i>Flowchart</i> Sistem Pada Aplikasi <i>Smartphone Android</i>	36
3.3	Perancangan Perangkat Keras	40



3.3.1	Pemasangan <i>Module WiFi NodeMCU ESP8266</i> pada <i>Arduino Mega 2560</i>	40
3.3.2	Pemasangan Sensor <i>Rotary</i> Pada Motor DC dan <i>Arduino Mega 2560</i>	40
3.4	Perancangan Perangkat Lunak	41
3.4.1	Perancangan Desain Aplikasi <i>Smartphone Android</i>	41
3.4.1.1	<i>Form Input Set</i>	42
3.4.1.2	<i>Form Loading</i>	42
3.4.1.3	<i>Form Monitoring</i>	43
3.4.1.4	<i>Form Scheduling</i>	44
3.4.1.5	<i>Form Set Scheduling</i>	45
3.4.1.6	<i>CheckBox</i>	47
3.4.1.7	<i>Form Setting</i>	47
3.4.1.8	<i>Form Channel Setting</i>	49
3.4.1.9	<i>Form Takaran Setting</i>	49
3.4.1.10	<i>Taskbar</i>	50
3.4.1.11	<i>Tombol Back</i>	51
3.4.2	Perancangan <i>Block Ambil</i> dan <i>Update Data Web Server</i>	51
3.4.3	Perancangan Komunikasi <i>Serial</i> pada <i>Arduino</i> dan <i>Module WiFi NodeMCU ESP8266</i>	53
3.4.4	Perancangan Program Koneksi <i>Module WiFi NodeMCU ESP8266</i> Dengan <i>WiFi</i>	54
3.4.5	Perancangan Program <i>Module WiFi NodeMCU ESP8266</i> Untuk Ambil Data Dari <i>Web Server</i>	56
3.4.6	Perancangan Program <i>Module WiFi NodeMCU ESP8266</i> Untuk <i>Update Data Web Server</i>	57

3.5 Pengaturan <i>Web Server</i>	58
BAB IV HASIL PENGUJIAN DAN PENGAMATAN	61
4.1 Pengujian <i>Arduino</i>	61
4.1.1 Tujuan Pengujian <i>Arduino</i>	61
4.1.2 Alat Yang Dibutuhkan Pengujian <i>Arduino</i>	62
4.1.3 Prosedur Pengujian <i>Arduino</i>	62
4.1.4 Hasil Pengujian <i>Arduino</i>	62
4.2 Pengujian <i>Module WiFi NodeMCU ESP8266</i>	63
4.2.1 Tujuan Pengujian <i>Module WiFi NodeMCU ESP8266</i>	64
4.2.2 Alat Yang Dibutuhkan Pengujian <i>Module WiFi NodeMCU ESP8266</i>	64
4.2.3 Prosedur Pengujian <i>Module WiFi NodeMCU ESP8266</i>	64
4.2.4 Hasil Pengujian <i>Module WiFi NodeMCU ESP8266</i>	65
4.3 Pengujian Pemantauan Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi <i>Android</i>	66
4.3.1 Tujuan Pengujian Pemantauan Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi <i>Android</i>	66
4.3.2 Alat Yang Dibutuhkan Pengujian Pemantauan Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi <i>Android</i>	67
4.3.3 Prosedur Pengujian Pemantauan Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi <i>Android</i> ...	67
4.3.4 Hasil Pengujian Pemantauan Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi <i>Android</i>	68
4.4 Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi <i>Android</i>	72
4.4.1 Tujuan Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi <i>Android</i> ...	73



4.4.2	Alat Yang Dibutuhkan Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi <i>Android</i>	73
4.4.3	Prosedur Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi <i>Smartphone Android</i>	73
4.4.4	Hasil Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi <i>Android</i> ...	75
4.5	Pengujian Penjadwalan Dari Aplikasi <i>Android</i>	79
4.5.1	Tujuan Pengujian Penjadwalan Dari Aplikasi <i>Android</i> ..	79
4.5.2	Alat Yang Dibutuhkan Pengujian Penjadwalan Dari Aplikasi <i>Android</i>	80
4.5.3	Prosedur Pengujian Penjadwalan Dari Aplikasi <i>Android</i>	80
4.5.4	Hasil Pengujian Penjadwalan Dari Aplikasi <i>Android</i>	81
4.6	Pengujian Penjadwalan Dari Alat Pemberi Pakan Ikan Otomatis	86
4.6.1	Tujuan Pengujian Penjadwalan Dari Alat Pemberi Pakan Ikan Otomatis	87
4.6.2	Alat Yang Dibutuhkan Pengujian Penjadwalan Dari Alat Pemberi Pakan Ikan Otomatis.....	87
4.6.3	Prosedur Pengujian Penjadwalan Dari Alat Pemberi Pakan Ikan Otomatis	88
4.6.4	Hasil Pengujian Penjadwalan Dari Alat Pemberi Pakan Ikan Otomatis	89
4.7	Pengujian Perubahan Takaran Pemberian Pakan Dari Aplikasi <i>Android</i> Kepada Alat Pemberi Pakan Ikan Otomatis	94
4.7.1	Tujuan Pengujian Perubahan Takaran Pemberian Pakan Dari Aplikasi <i>Android</i> Kepada Alat Pemberi Pakan Ikan Otomatis	95

4.7.2	Alat Yang Dibutuhkan Pengujian Perubahan Takaran Pemberian Pakan Dari Aplikasi <i>Android</i> Kepada Alat Pemberi Pakan Ikan Otomatis	95
4.7.3	Prosedur Pengujian Perubahan Takaran Pemberian Pakan Dari Aplikasi <i>Android</i> Kepada Alat Pemberi Pakan Ikan Otomatis	95
4.7.4	Hasil Pengujian Perubahan Takaran Pemberian Pakan Dari Aplikasi <i>Android</i> Kepada Alat Pemberi Pakan Ikan Otomatis	97
4.8	Analisis Hasil Pengujian	99
BAB V PENUTUP	102
5.1	Kesimpulan	102
5.2	Saran.....	103
DAFTAR PUSTAKA	104
BIODATA PENULIS	105
LAMPIRAN	106

DAFTAR GAMBAR

Gambar 2.1	<i>Arduino Mega 2560</i>	7
Gambar 2.2	<i>Module WiFi NodeMCU ESP8266</i>	8
Gambar 2.3	Gambaran Umum <i>Web Server</i>	15
Gambar 2.4	Motor DC	16
Gambar 2.5	Blok Sensor <i>Optocoupler</i>	17
Gambar 2.6	Konstruksi Sensor Putaran	17
Gambar 2.7	Tampilan <i>Arduino IDE</i>	19
Gambar 2.8	Tampilan <i>App Inventor</i>	21
Gambar 3.1	Perancangan Sistem	24
Gambar 3.2	<i>Flowchart Module WiFi NodeMCU ESP8266</i>	30
Gambar 3.3	<i>Flowchart Sub Algoritma Baca Serial</i>	31
Gambar 3.4	<i>Flowchart Sub Algoritma Ambil Data Server</i>	32
Gambar 3.5	<i>Flowchart Arduino</i>	34
Gambar 3.6	<i>Flowchart Sub algoritma Jadwal ON/OFF</i>	35
Gambar 3.7	<i>Flowchart Sub Algoritma Check Data Arduino</i>	35
Gambar 3.8	<i>Flowchart Aplikasi Smartphone Android</i>	36
Gambar 3.9	<i>Flowchart Sub Algoritma Scheduling</i>	37
Gambar 3.10	<i>Flowchart Sub Algoritma Monitoring</i>	38
Gambar 3.11	<i>Flowchart Sub Algoritma Takaran</i>	39
Gambar 3.12	Konfigurasi <i>Module WiFi ESP8266</i> Pada <i>Arduino Mega</i>	40
Gambar 3.13	Konfigurasi <i>Rotary</i> Pada Motor DC dan <i>Arduino Mega</i>	41

Gambar 3.14	Desain <i>Form Input Set</i>	42
Gambar 3.15	Desain <i>Form Loading</i>	43
Gambar 3.16	Desain <i>Form Monitoring</i>	44
Gambar 3.17	Desain <i>Form Scheduling</i> Saat Tidak Ada Jadwal	45
Gambar 3.18	Desain <i>Form Scheduling</i> Saat Ada Jadwal	45
Gambar 3.19	Desain <i>Form Set Scheduling</i> Saat Tidak Ada Jadwal	46
Gambar 3.20	Desain <i>Form Set Scheduling</i> Saat Ada Jadwal	46
Gambar 3.21	Tampilan Pengaturan atau Perubahan Jadwal	47
Gambar 3.22	<i>CheckBox</i> Pada <i>Form Set Scheduling</i>	48
Gambar 3.23	Desain Tombol Untuk <i>Form Setting</i>	48
Gambar 3.24	Desain <i>Form Setting</i>	48
Gambar 3.25	Desain <i>Form Channel Setting</i>	49
Gambar 3.26	Desain <i>Form Takaran Setting</i>	50
Gambar 3.27	Desain <i>Taskbar</i>	50
Gambar 3.28	Desain Tombol <i>Back</i>	51
Gambar 3.29	Inisialisasi <i>Component web</i>	52
Gambar 3.30	Pengaksesan dan Penyimpanan Data <i>Web Server</i>	53
Gambar 3.31	<i>Update Data Web Server</i>	53
Gambar 3.32	Kode Komunikasi <i>Serial Arduino</i> dan <i>Module ESP8266</i>	54
Gambar 3.33	Kode Koneksi <i>Module WiFi ESP8266</i> Dengan <i>WiFi</i>	55
Gambar 3.34	Kode Ambil Data <i>Web Server</i> Pada <i>Module WiFi ESP8266</i> ..	56
Gambar 3.35	Kode <i>Update Data Web Server</i> Pada <i>Module WiFi ESP8266</i> ..	57
Gambar 3.36	<i>Sign Up thingspeak.com</i>	58

Gambar 3.37	Berhasil Membuat <i>Account</i> dan Masuk Pada <i>Web Server</i>	59
Gambar 3.38	Halaman <i>Channel</i> Baru	59
Gambar 3.39	Tampilan Awal <i>Channel</i> Baru	60
Gambar 3.40	Tampilan <i>API Keys</i> dan <i>Channel ID</i>	60
Gambar 4.1	Tampilan Proses <i>Upload</i> Dari <i>Arduino IDE</i> Kepada <i>Arduino</i>	63
Gambar 4.2	Tampilan <i>Comment</i> Saat Berhasil Upload Kepada <i>Arduino</i> ...	63
Gambar 4.3	Tampilan Proses <i>Upload</i> dari <i>Arduino IDE</i> Kepada <i>Module WiFi NodeMCU ESP8266</i>	65
Gambar 4.4	Tampilan <i>Comment</i> Saat Berhasil <i>Upload</i> Kepada <i>Module WiFi NodeMCU ESP8266</i>	66
Gambar 4.5	Tampilan Menu <i>Monitoring</i> Kerusakan	68
Gambar 4.6	Tampilan <i>Notifikasi</i> Terjadi Kerusakan	69
Gambar 4.7	Tampilan <i>Input</i> Berat Pakan Pada <i>Display LCD</i>	74
Gambar 4.8	Tampilan <i>Input</i> Takaran Pada <i>Display LCD</i>	74
Gambar 4.9	Tampilan <i>Input</i> Jadwal Pada <i>Display LCD</i>	74
Gambar 4.10	Tampilan <i>Input Start</i> Pada <i>Display LCD</i>	75
Gambar 4.11	Tampilan Menu <i>Monitoring</i> Pakan	75
Gambar 4.12	Tampilan Menu <i>Monitoring</i> Pada Aplikasi “ <i>SMaS.apk</i> ”	76
Gambar 4.13	Tampilan Atur Jadwal	81
Gambar 4.14	Tampilan Simpan Dan <i>Update</i> Jadwal	81
Gambar 4.15	<i>Display LCD</i> Perubahan Jadwal Pada Alat Pemberi Pakan Ikan Otomatis	82
Gambar 4.16	<i>Display LCD</i> Data Jadwal Yang Diterima Oleh Alat Pemberi Pakan Ikan Otomatis.....	82
Gambar 4.17	Tampilan Menu <i>Scheduling</i>	88

Gambar 4.18	Tampilan <i>Input</i> Menu Pada <i>Display LCD</i>	89
Gambar 4.19	Tampilan <i>Input</i> Jadwal Pada <i>Display LCD</i>	89
Gambar 4.20	Tampilan <i>Start</i> Pada <i>Display LCD</i>	89
Gambar 4.21	Tampilan Menu <i>Scheduling</i> Pada Aplikasi “SMaS.apk”	90
Gambar 4.22	Tampilan Menu <i>Setting</i>	96
Gambar 4.23	Tampilan Menu Takaran <i>Setting</i>	96
Gambar 4.24	Tampilan Simpan Dan <i>Update</i> Data Takaran	97
Gambar 4.25	<i>Display LCD</i> Data Takaran Yang Diterima Oleh Alat Pemberi Pakan Ikan Otomatis	97



DAFTAR TABEL

Tabel 2.1	Spesifikasi <i>Arduino Mega 2560</i>	7
Tabel 2.2	Spesifikasi <i>NodeMCU</i>	9
Tabel 3.1	Penjelasan Data <i>Server</i>	28
Tabel 4.1	Parameter Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis..	69
Tabel 4.2	Hasil Pengujian Pemantauan Kerusakan Pada Penggerak Alat Pemberi Pakan Ikan Otomatis Dengan Aplikasi <i>Android</i>	70
Tabel 4.3	Hasil Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dengan Aplikasi <i>Android</i>	76
Tabel 4.4	Hasil Pengujian Penjadwal Dari Aplikasi <i>Android</i>	82
Tabel 4.5	Hasil Pengujian Penjadwal Dari Alat Pemberi Pakan Ikan Otomatis Kepada Aplikasi <i>Android</i>	90
Tabel 4.6	Hasil Pengujian Perubahan Takaran Pada Alat Pemberi Pakan Ikan Otomatis Dengan Aplikasi <i>Android</i>	98

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Salah satu faktor utama dalam pertumbuhan ikan adalah pakan. Hal itu dikarenakan pakan akan mempengaruhi penambahan bobot, panjang atau volume ikan. Pemberian pakan yang tidak efisien akan berpengaruh terhadap penumpukan sisa pakan dan ekskresi ikan, yang dapat menjadi penyebab penurunan kualitas kolam ikan, dan secara tidak langsung dapat mempengaruhi produktivitas kolam ikan. Pada saat ini sudah banyak terdapat alat pemberi pakan ikan otomatis, akan tetapi kebanyakan masih terkendala dalam pemantauan kerusakan dan pemantauan ketersediaan pakan pada alat tersebut, serta pengaturan jadwal yang harus dilakukan secara manual dan langsung pada alat tersebut.

Dengan demikian, pemantauan dan pengaturan jadwal pemberian pakan secara jarak jauh adalah hal yang penting untuk diupayakan. Melihat perkembangan teknologi saat ini yang semakin mempermudah aktivitas manusia, maka dalam tugas akhir ini akan dibuat suatu sistem yang dapat mempermudah aktivitas pemantauan kerusakan dan ketersediaan pakan, serta dapat mengatur jadwal pemberian pakan pada Alat Pemberi Pakan Ikan Otomatis yang dapat dilakukan oleh pembudidaya secara jarak jauh tanpa mendatangi alat secara langsung.

Perancangan sistem manajemen pakan kolam ikan ini sebelumnya sudah pernah dilakukan oleh (Fathurohim, 2015). Pada sistem yang sebelumnya menggunakan *raspberry pi* sebagai *web server* dan aplikasi *android* sebagai *user interface*. Dalam sistem yang sebelumnya dapat memanajemen pakan kolam ikan

secara jarak jauh, tetapi tidak dapat memantau kerusakan yang terjadi pada alat dan ketersediaan pakan pada alat. Sehingga jika terjadi kerusakan atau tidak tersedianya pakan pada alat, pembudidaya tidak dapat mengetahui informasi tersebut sebelum mendatangi dan memantau alat tersebut secara langsung.

Pada perancangan sistem yang selanjutnya ini, akan dibuat sebuah sistem yang dapat memantau kerusakan dan ketersediaan pakan pada alat, serta mengatur jadwal pemberian pakan ikan pada kolam ikan secara jarak jauh. Sehingga pembudidaya dapat memantau kerusakan yang terjadi dan ketersediaan pakan pada alat, serta mengatur jadwal pemberian pakan secara jarak jauh tanpa mendatangi alat secara langsung. Pada sistem yang selanjutnya ini, menggunakan aplikasi *smartphone android* sebagai *user interface* yang akan berkomunikasi dengan mikrokontroller *Arduino* dan *Module WiFi NodeMCU ESP8266* pada alat menggunakan *internet* dan *web server* sebagai perantara, serta *Modem WiFi* sebagai media agar alat dapat terhubung dengan internet.

1.2 Rumusan Masalah

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

1. Bagaimana mengetahui dan memantau kerusakan alat pemberi pakan ikan otomatis secara jarak jauh menggunakan aplikasi *smartphone android* melalui *internet*.

2. Bagaimana mengetahui dan memantau ketersediaan pakan pada alat pemberi pakan ikan otomatis secara jarak jauh menggunakan aplikasi *smartphone android* melalui *internet*.
3. Bagaimana mengubah jadwal pemberian pakan secara jarak jauh menggunakan aplikasi *smartphone android* melalui *internet*.

1.3 Batasan Masalah

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

1. Kontroler yang digunakan adalah mikrokontroller *Arduino Mega 2560*.
2. Pakan ikan yang digunakan adalah pakan ikan jenis *pellet*.
3. Pemantauan kerusakan alat secara jarak jauh hanya mencakup kerusakan yang terjadi pada bagian penggerak untuk mengeluarkan pakan dan pelempar pakan.
4. Pemantauan ketersediaan pakan secara jarak jauh hanya mencakup ketersediaan pakan yang ada pada tempat penampung pakan.
5. Kontrol atau pengaturan alat secara jarak jauh hanya mencakup pada pengaturan jadwal pemberian pakan.
6. Pada tugas akhir ini menggunakan aplikasi *smartphone android* yang berfungsi sebagai *user interface*.
7. *Smartphone android* dan Alat pemberi pakan ikan harus terhubung dengan *wifi* atau dapat mengakses *internet*.

1.4 Tujuan

Tujuan dari penelitian yang akan dilakukan adalah sebagai berikut :

1. Sistem dapat mengetahui dan memantau kerusakan alat pemberi pakan ikan otomatis secara jarak jauh menggunakan aplikasi *smartphone android* melalui *internet*.
2. Sistem dapat mengetahui dan memantau ketersediaan pakan pada alat pemberi pakan ikan otomatis secara jarak jauh menggunakan aplikasi *smartphone android* melalui *internet*.
3. Sistem dapat mengubah jadwal pemberian pakan secara jarak jauh menggunakan aplikasi *smartphone android* melalui *internet*.

1.5 Sistematika Penulisan

Tugas Akhir ini secara garis besar tersusun dari 5 (lima) bab, yaitu diuraikan sebagai berikut:

BAB I PENDAHULUAN

Pada Bab ini akan dibahas mengenai latar belakang masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada Bab ini akan dibahas teori penunjang dari permasalahan, yaitu membahas mengenai Mikrokontroller *Arduino*, *Module WiFi NodeMCU ESP8266*, *Internet*, *Modem WiFi*, *Arduino software IDE*, *Android*, *App Inventor*, dan serta perangkat *hardware* pendukung.

BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM

Pada Bab ini akan dibahas tentang *blog diagram* sistem serta metode yang dilakukan dalam mewujudkan pembuatan alat dan aplikasi *Android*. Pada bab ini juga akan dibahas tentang cara berkirim data antara alat dan aplikasi *Android* yang akan dibuat, serta membahas tentang metode yang digunakan pada Tugas Akhir ini.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil dari pengujian masing-masing komponen pendukung dalam pembuatan alat dan aplikasi *Android*, dimana nantinya hasil dari pengujian masing-masing komponen akan menentukan apakah aplikasi *Android* dan perangkat keras bekerja dengan baik. Selain itu data dari pengujian aplikasi *Android* dapat digunakan sebagai dasar pengambilan nilai-nilai data pada sistem keseluruhan. Kemudian akan dibahas dari hasil pengujian perancangan seluruh sistem yang nantinya dapat diperoleh hasil nilai-nilai kondisi yang tepat agar sistem dapat bekerja dengan baik sesuai dengan yang diharapkan.

BAB V KESIMPULAN DAN SARAN

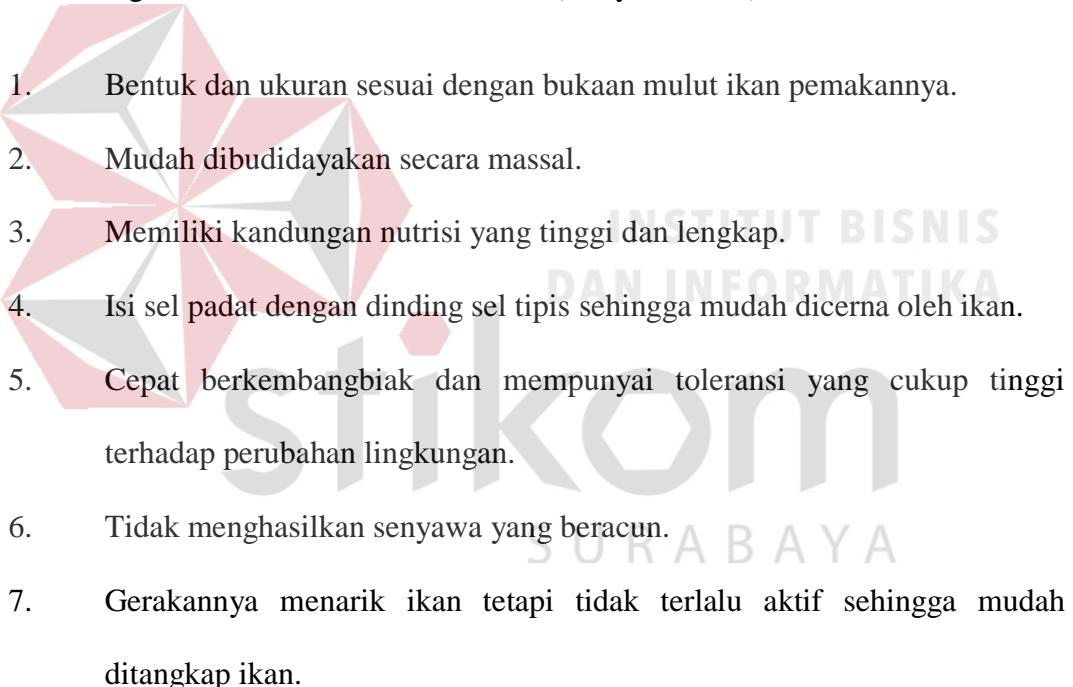
Berisi kesimpulan yang didapat dari hasil penelitian berdasarkan rumusan masalah serta saran untuk perkembangan penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Pakan Ikan

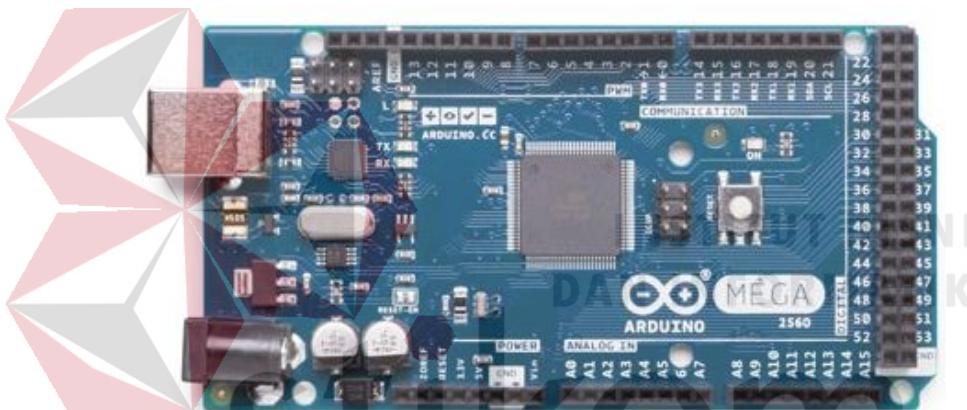
Untuk membantu pertumbuhan pakan alami dapat dilakukan dengan pemupukan. Akan tetapi tidak semua organisme renik dapat dimanfaatkan oleh ikan sebagai makanan alami. Berikut kriteria pakan alami yang dapat dimanfaatkan oleh ikan sebagai sumber makanan antara lain: (Cahyadi, 2005)

- 
1. Bentuk dan ukuran sesuai dengan bukaan mulut ikan pemakannya.
 2. Mudah dibudidayakan secara massal.
 3. Memiliki kandungan nutrisi yang tinggi dan lengkap.
 4. Isi sel padat dengan dinding sel tipis sehingga mudah dicerna oleh ikan.
 5. Cepat berkembangbiak dan mempunyai toleransi yang cukup tinggi terhadap perubahan lingkungan.
 6. Tidak menghasilkan senyawa yang beracun.
 7. Gerakannya menarik ikan tetapi tidak terlalu aktif sehingga mudah ditangkap ikan.

2.2 Arduino Mega 2560

Arduino Mega 2560 adalah piranti mikrokontroller menggunakan ATmega2560. Modul ini memiliki 54 digital *input* atau *output*. Dimana 14 pin digunakan untuk PWM output dan 16 pin digunakan sebagai *analog input*, 4 pin untuk UART, 16 MHz osilator kristal, koneksi USB, *power jack ICSP header*, dan

tombol *reset*. Modul ini memiliki segala yang dibutuhkan untuk memprogram mikrokontroller seperti kabel USB dan catu daya melalui adaptor atau baterai. Semua ini diberikan untuk mendukung pemakaian mikrokontroller Arduino, hanya terhubung ke computer dengan kabel USB atau listrik dengan adaptor dari AC ke DC atau baterai untuk pemakaian. Arduino Mega kompatibel dengan *shield* yang dirancang untuk Arduino Duemilanove, Decimila maupun UNO. (Arduino, 2017).



Gambar 2.1 *Arduino Mega 2560*

Tabel 2.1 Spesifikasi *Arduino Mega 2560*

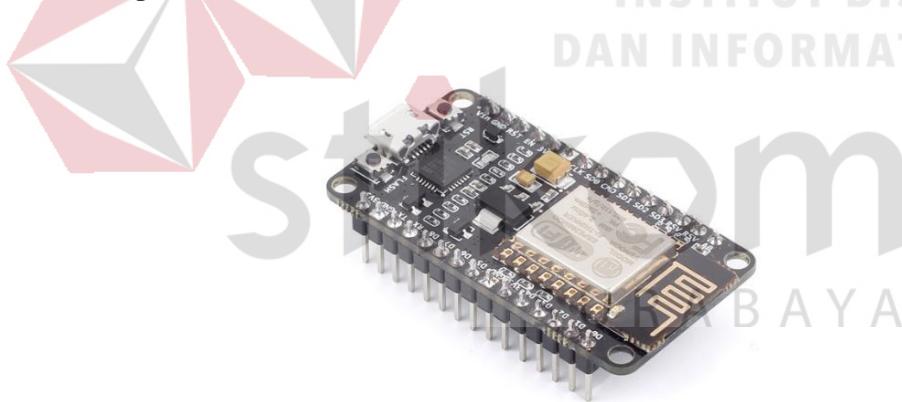
Mikrokontroller	ATmega 2560
<i>Operating Voltage</i>	5V
<i>Input Voltage (recommended)</i>	7 – 12 V
<i>Input Voltage (limits)</i>	6 – 20 V
<i>Digital I/O Pins</i>	54 (15 PWM output)
<i>Analog Input Pins</i>	16
<i>DC current for I/O pin</i>	40 Ma
<i>DC current for 3.3 V pin</i>	50 Ma
<i>Flash Memory</i>	256 KB (8 KB digunakan untuk <i>bootloader</i>)
<i>SRAM</i>	8 KB
<i>EEPROM</i>	4 KB
<i>Clock Speed</i>	16 MHz

2.3 *Module WiFi NodeMCU ESP8266*

2.3.1 Konsep Dasar *NodeMCU*

1. Definisi *NodeMCU*

NodeMCU adalah *Open-source firmware* dan pengembangan kit yang membantu untuk membuat prototipe produk IOT (*Internet Of Things*) dalam beberapa baris skrip. *Node Mcu* adalah sebuah platform open source IOT (*Internet Of Things*). *Node Mcu* menggunakan *Lua* sebagai bahasa *scripting*. Hal ini didasarkan pada proyek *Elua*, dan dibuat di atas *ESP8266* SDK 1.4. Menggunakan banyak proyek *open source*, seperti *lua-cjson*. Ini mencakup firmware yang berjalan pada *Wi-Fi SoC ESP8266*, dan perangkat keras yang didasarkan pada *ESP-12* modul.



Gambar 2.2 *Module WiFi NodeMCU ESP8266*

2. Sejarah *Node Mcu*

Node Mcu diciptakan tidak lama setelah *ESP8266* keluar. Pada 30 Desember 2013, sistem *Espressif* mulai produksi *ESP8266* tersebut. *ESP8266* adalah *SoC Wi-Fi* terintegrasi dengan inti *Tensilica Xtensa LX106*, banyak

digunakan dalam aplikasi IOT (*Internet Of Things*). *NodeMCU* dimulai pada 13 Oktober 2014, ketika Hong berkomitmen file pertama *nodeMCU-firmware* untuk *GitHub*. Dua bulan kemudian, proyek ini diperluas untuk mencakup sebuah *platform* terbuka-hardware ketika pengembang Huang R berkomitmen file *Gerber* dari papan *ESP8266*, bernama *devkit* 1.0. Kemudian di bulan itu, Tuan PM porting perpustakaan klien MQTT dari *Contiki* ke *platform* *ESP8266 SoC*, dan berkomitmen untuk proyek *NodeMCU*, maka *NodeMCU* mampu mendukung protokol MQTT IOT(*Internet Of Things*), menggunakan Lua untuk mengakses MQTT *broker*. Update penting lain dibuat pada 30 Januari 2015, ketika Devsaurus untuk proyek *Node MCU*. *Node Mcu* memungkinkan untuk dengan mudah mengarahkan LCD, Screen, OLED, bahkan VGA *display*.

3. Spesifikasi *Node Mcu*

Fitur yang disediakan oleh *Node Mcu* adalah bersifat *Open source*, Interaktif, telah diprogram, biaya yang tergolong rendah, sederhana, *Smart*, dan *WiFi* diaktifkan.

Tabel 2.2 Spesifikasi *NodeMCU*

Pengembang	<i>ESP8266 Open source Komunitas</i>
Jenis	<i>Single</i> -papan mikrokontroler
Sistem Operasi	XTOS
CPU	<i>ESP8266</i> dan (LX106)
Memori	20 kBytes
Penyimpanan	4 MBytes
Power	USB

4. Pengembangan Alat

Pengembangan alat berdasarkan *ESP8266*, mengintegrasikan GPIO, PWM, IIC, 1-Wire dan ADC semua dalam satu papan. Daya perkembangan dalam cara combining tercepat dengan *NodeMCU Firmware*. USB-TTL termasuk, plug, 10 GPIO, setiap GPIO bisa PWM, I2C, 1-kawat, dan FCC BERSERTIFIKAT WI-FI modul, antena PCB.

2.3.2 Definisi *ESP8266*

ESP8266 adalah *wifi module* dengan *output serial TTL* yang dilengkapi dengan GPIO, *wifi module* ini dapat dipergunakan secara *standalone* maupun dengan mikrokontroler tambahan untuk kendalinya. Ada beberapa jenis *ESP8266* yang dapat ditemui dipasaran, namun yang paling mudah didapatkan di Indonesia adalah type *ESP-01*, *07*, dan *12* dengan fungsi yang sama perbedaannya terletak pada GPIO pin yang disediakan. Tegangan kerja *ESP-8266* adalah sebesar 3.3V, sehingga untuk penggunaan mikrokontroler tambahannya dapat menggunakan board arduino atau node mcu yang memiliki fasilitas tengangan sumber 3.3V, akan tetapi akan lebih baik jika membuat secara terpisah level shifter untuk komunikasi dan sumber tegangan untuk wifi module ini. Karena, wifi module ini dilengkapi dengan Mikrokontroler dan GPIO sehingga banyak yang mengembangkan *firmware* untuk dapat menggunakan module ini tanpa perangkat mikrokontroler tambahan. Firmware yang digunakan agar wifi module ini dapat bekerja *standalone* adalah *Node Mcu*. Dengan menggunakan *node MCU* kita dapat membuat kode

untuk *wifi module* ini dalam bentuk *LUA* sehingga *GPIO* yang tedapat pada *wifi module* ini dapat dipergunakan sesuai dengan keinginan kita. (Novianty, 2016).

2.4 *Internet*

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

Ada beberapa jenis internet, antara lain (Nurahman, 2012) :

1. *Local Area Network (LAN)*

Merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan workstation dalam kantor suatu

perusahaan atau pabrik-pabrik untuk memakai bersama sumberdaya (misalnya printer) dan saling bertukar informasi.

2. *Metropolitan Area Network (MAN)*

Pada dasarnya merupakan versi LAN yang berukuran lebih besar dan biasanya menggunakan teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang letaknya berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi (swasta) atau umum. MAN mampu menunjang data dan suara, bahkan dapat berhubungan dengan jaringan televisi kabel.

3. *Wide Area Network (WAN)*

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

4. *Internet*

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

5. Jaringan Tanpa Kabel

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

2.5 **Modem**

Modem berasal dari singkatan *Modulator Demodulator*. *Modulator* merupakan bagian yang mengubah sinyal informasi kedalam sinyal pembawa (*carrier*) dan siap untuk dikirimkan, sedangkan *Demodulator* adalah bagian yang memisahkan sinyal informasi (yang berisi data atau pesan) dari sinyal pembawa yang diterima sehingga informasi tersebut dapat diterima dengan baik. *Modem* merupakan penggabungan kedua-duanya, artinya *modem* adalah alat komunikasi dua arah. Setiap perangkat komunikasi jarak jauh dua arah umumnya menggunakan bagian yang disebut “MODEM”, seperti VSAT, *Microwave radio*, dan lain sebagainya, namun umumnya istilah modem lebih dikenal sebagai perangkat keras yang sering digunakan untuk komunikasi pada komputer. Ada beberapa jenis *modem*, antara lain (Nurahman, 2012) :

1. *Modem analog* adalah modem yang berfungsi mengubah sinyal *analog* menjadi sinyal *digital*.
2. *Modem ADSL* adalah teknologi modem yang memungkinkan pengguna bisa melakukan *browsing internet* secara bersamaan dengan aktifitas lain seperti telepon dalam satu konektor kabel telepon.
3. *Modem kabel* adalah perangkat keras yang digunakan untuk menyambungkan PC dengan sambungan TV kabel.
4. *Modem CDMA* adalah *modem* yang jaringannya melalui frekuensi pada jaringan yang berbasis CDMA, biasanya modemnya berupa ponsel berbasis CDMA.
5. *Modem GSM* adalah modem yang jaringannya melalui frekuensi pada jaringan berbasis GSM. Modem ini tidak seperti modem CDMA yang berupa ponsel, modem ini memiliki bentuk sendiri yang bekerja pada frekuensi GSM.
6. *Modem satelit* adalah modem yang transfer datanya menggunakan komunikasi satelit sebagai relay.
7. *Modem 3G* adalah modem yang jaringannya melalui frekuensi jaringan berbasis 3G digunakan pada perkembangan teknologi telepon nirkabel (*wirelees*).

2.6 *Web Server*

Menurut Muhammad dan Emi, 2015, *web server* merupakan *software* yang memberikan layanan data, berfungsi menerima permintaan HTTP atau HTTPS dari

client yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML. Gambaran umum dari *web server* dapat dilihat pada gambar 3.

Antara *web server*, *browser* dan *user* adalah suatu proses yang *tridimensional*, artinya proses yang dimulai dari permintaan *web-client* (aplikasi *browser*), diterima *web server*, diproses dan dikembalikan hasil prosesnya oleh *web server* ke *web client* lagi yang dikerjakan secara transparan. Secara garis besarnya *web server* hanya memproses semua masukan yang diperolehnya dari *web client* (Pratama, 2017).

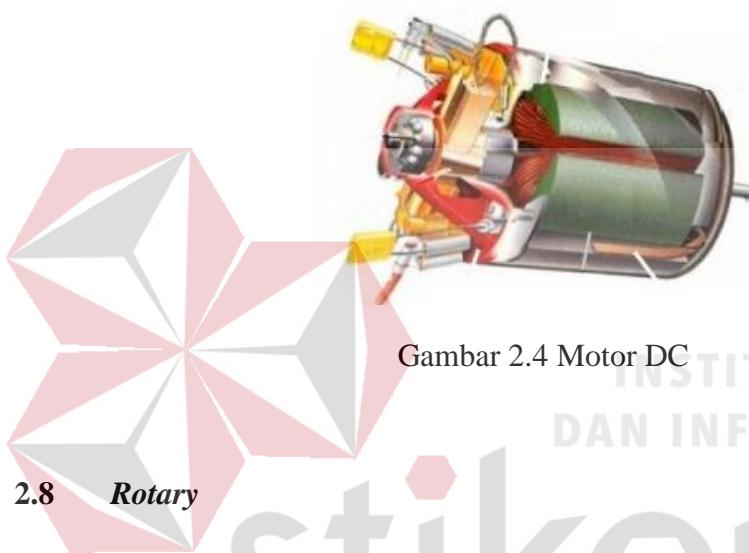


Gambar 2.3 Gambaran Umum *Web Server*

2.7 Motor DC

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor dc disebut *stator* (bagian yang tidak berputar) dan kumparan jangkar disebut *rotor* (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik *phasa* tegangan dari gelombang yang mempunyai nilai

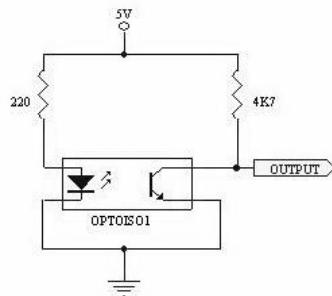
positif dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. Pada gambar 6 dibawah ini dapat dilihat gambaran dari Motor DC. (Saragih, 2016).



Gambar 2.4 Motor DC

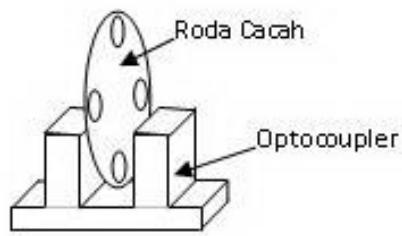
Sensor putaran atau kecepatan dapat dibuat dengan sebuah *optocoupler* tipe “U” dan sebuah roda cacah. Sensor putaran atau kecepatan ini dapat digunakan untuk membaca putaran suatu object yang berputar seperti roda kendaraan, putaran motor listrik dan lain nya. Sensor putaran atau kecepatan ini dibuat dengan *optocoupler* tipe “U” yang ditengahnya diletakan sebuah roda cacah. *Optocoupler* merupakan komponen *optoisolator* yang memiliki karakteristik penerima (*phototransistor*) akan mengalami perubahan logika bila terjadi perubahan intensitas cahaya yang dipancarkan oleh pemancar (LED infra merah) untuk penerima. Kecerahan led berbanding lurus dengan arus diodanya. Karena arus kolektor sebanding dengan tingkat kecerahan dari led maka dapat dikatakan bahwa arus dioda mengendalikan arus kolektor seperti transistor pada umumnya. Biasanya arus

yang diperbolehkan mengalir pada infra merah adalah berkisar pada 15 hingga 25 miliamper. Foto transistor merupakan jenis transistor yang peka terhadap cahaya infra merah. Pada gambar 4 menunjukkan rangkaian blok sensor *optocoupler*.



Gambar 2.5 Blok Sensor *Optocoupler*

Roda cacah yang diletakan ditengah *optocoupler* tersebut berfungsi untuk mempengaruhi intensitas cahaya yang diberikan oleh LED pada *optocoupler* ke *photo* transistor yang akan memberikan perubahan level logika sesuai dengan putaran roda cacah. Kecepatan perubahan logika *photo* transistor akan sebanding dengan kecepatan putaran roda cacah. Konstruksi sensor putaran dapat dilihat pada gambar 5 berikut.



Gambar 2.6 Konstruksi Sensor Putaran

Rotary encoder, atau disebut juga *Shaft encoder*, merupakan perangkat elektromekanikal yang digunakan untuk mengkonversi posisi anguler (sudut) dari

shaft (lubang) atau roda ke dalam kode digital, menjadikannya semacam *transducer*.

Perangkat ini biasanya digunakan dalam bidang robotika, perangkat masukan komputer (seperti *optomekanikal mouse* dan *trackball*), serta digunakan dalam kendali putaran radar, dll.

Terdapat dua tipe utama *rotary encoder*, yaitu tipe absolut dan tipe relatif. *Rotary encoder*, atau disebut juga *Shaft encoder*, merupakan perangkat elektromekanikal yang digunakan untuk mengkonversi posisi anguler (sudut) dari *shaft* (lubang) atau roda ke dalam kode digital, menjadikannya semacam *transducer*. Perangkat ini biasanya digunakan dalam bidang robotika, perangkat masukan komputer (seperti *optomekanikal mouse* dan *trackball*), serta digunakan dalam kendali putaran radar, dll. Terdapat dua tipe utama *rotary encoder*, yaitu tipe *absolut* dan tipe *relatif* (Sensor Putaran, 2012).

2.9 Komunikasi Serial

Komunikasi serial adalah komunikasi yang pengiriman datanya per-bit secara berurutan dan bergantian. Komunikasi ini mempunyai suatu kelebihan yaitu hanya membutuhkan satu jalur dan kabel yang sedikit dibandingkan dengan komunikasi paralel. Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per-bit sehingga lebih lambat dibandingkan komunikasi paralel, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data di mana hanya satu bit data yang dikirimkan pada suatu waktu tertentu (Shneiderman, 2000).

Komunikasi serial ada dua macam, *asynchronous* serial dan *synchronous* serial. *Synchronous* serial adalah komunikasi dimana hanya ada satu pihak (pengirim atau penerima) yang menghasilkan *clock* dan mengirimkan *clock* tersebut bersama-sama dengan data. Contoh penggunaan *synchronous* serial terdapat pada transmisi data *keyboard*. *Asynchronous* serial adalah komunikasi dimana kedua pihak (pengirim dan penerima) masing-masing menghasilkan *clock* namun hanya data yang ditransmisikan, tanpa *clock*. Agar data yang dikirim sama dengan data yang diterima, maka kedua frekuensi *clock* harus sama dan harus terdapat sinkronisasi. Setelah adanya sinkronisasi, pengirim akan mengirimkan datanya sesuai dengan frekuensi *clock* pengirim dan penerima akan membaca data sesuai dengan frekuensi *clock* penerima. Contoh penggunaan *asynchronous* serial adalah pada *universal asynchronous receiver transmitter* (UART) yang digunakan pada serial port (COM) komputer.

2.10 Arduino Software IDE



Gambar 2.7 Tampilan *Arduino IDE*

Platform dari *arduino* disusun pada sebuah *software* yang diberi nama *Arduino IDE*. *Software* inilah yang paling utama, membantu menjembatani antara bahasa mesin yang begitu rumit sehingga menjadi bahasa dan *logic* yang lebih mudah dimengerti manusia (Ecadio, 2015).

2.11 *Android*

Android pada mulanya berasal dari perusahaan bernama *Android, Inc.* Didirikan di Palo Alto, California, pada Oktober tahun 2003, Pada tahun 2005 Google membeli/mengakusisi perusahaan yang waktu itu bernama *Android* sebuah perusahaan yang berkecimpung di dunia *mobile*. Setelah diambil alih oleh Google, pada November 2007 mereka mengumumkan bahwa sedang mengembangkan ponsel Google dengan OS (*Operating System*) *mobile* terbaru yang diberi nama *Android*. Menurut Teguh Arifanto (2011:1), *Android* merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis linux.

Terdapat beberapa daftar urutan pengembangan sistem operasi *Android*, oleh Google dan Open Handset Alliance (OHA) : *Android, Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice cream Sandwich, Jelly Bean, Kitkat, Lollipop, Marshmallow* (Aristo, 2016).

2.12 *App Inventor*

App Inventor for Android adalah aplikasi yang pada dasarnya disediakan oleh Google, dan sekarang di-*maintanance* oleh Massachusetts Institute of Technology (MIT). *App Inventor* memungkinkan semua orang untuk membuat

software aplikasi untuk sistem operasi Android. Pengguna dapat menggunakan tampilan grafis GUI dan fitur *drag and drop* visual objek untuk membuat sebuah aplikasi dapat berjalan pada sistem operasi Android. Tampilan dari *App Inventor* dapat dilihat pada gambar 2.8.

App inventor menggunakan *Kawa Language Framework* dan *Kawa's dialect* yang dikembangkan oleh Per Bothner. Kedua aplikasi tersebut didistribusikan sebagai bagian GNU *Operating System* oleh *Free Software Foundation*. Kedua aplikasi tersebut dijadikan sebagai *compiler* dan menerjemahkan *Visual Block Programming* untuk diimplementasikan pada *platform Android* (Sulton, 2016).



Gambar 2.8 Tampilan *App Inventor*

BAB III

METODE PENELITIAN DAN PERENCANAAN SISTEM

3.1 Metode Penelitian

Dalam perancangan sistem ini, awal mula program dibuat pada personal komputer, setelah itu program di *compile* untuk dijadikan sebuah aplikasi *smartphone android*. Aplikasi yang telah dibuat dikirimkan dan di *install* pada *Smartphone Android*. Pada aplikasi *smartphone Android* yang telah di *install* terdapat beberapa fungsi, yaitu untuk memantau kerusakan dan ketersediaan pakan ikan pada Alat Pemberi Pakan Ikan Otomatis, serta mengatur jadwal dan takaran pemberian pakan ikan pada Alat Pemberi Pakan Ikan Otomatis.

Langkah pertama untuk menggunakan aplikasi *smartphone Android* ini, *user* harus memasukkan *API Key* dan *Channel ID* tertentu. *API Key* dan *Channel ID* tersebut berfungsi sebagai kunci untuk dapat mengakses *web server*. *Web server* digunakan sebagai perantara pengiriman data dari alat menuju aplikasi *smartphone Android* dan sebaliknya. Pada aplikasi *smartphone Android* ini terdapat dua menu utama, yaitu menu *monitoring*/pemantauan dan menu *scheduling*/penjadwalan. Menu *scheduling* berfungsi untuk merubah/mengatur jadwal pada Alat Pemberi Pakan Ikan Otomatis. Menu *monitoring* berfungsi untuk melakukan pemantauan kerusakan maupun ketersediaan pakan pada Alat Pemberi Pakan Ikan Otomatis. Pada menu pemantauan terdapat satu buah tombol tambahan yang berfungsi untuk *request / cek* kerusakan pada Alat Pemberi Pakan Ikan Otomatis. Sedangkan untuk perubahan takaran dapat dilakukan pada menu *setting*.

Untuk perubahan jadwal atau takaran dari sistem dapat dibagi menjadi dua, yaitu perubahan jadwal atau takaran dari aplikasi *Android* dan perubahan jadwal atau takaran dari alat. Perubahan jadwal atau takaran dari Alat Pemberi Pakan Ikan Otomatis, *user* akan menginputkan jadwal atau takaran pada alat dan dikirimkan pada *web server*, kemudian aplikasi akan mengakses dan menyimpan data takaran atau penjadwalan dari *web server*. Sedangkan untuk perubahan takaran atau jadwal dari aplikasi *smartphone Android*, data yang telah diinputkan oleh *user* pada aplikasi *smartphone Android* akan dikirimkan oleh aplikasi *smartphone Android* kepada *web server*, kemudian alat akan mengakses dan menyimpan data takaran atau penjadwalan dari *web server* tersebut.

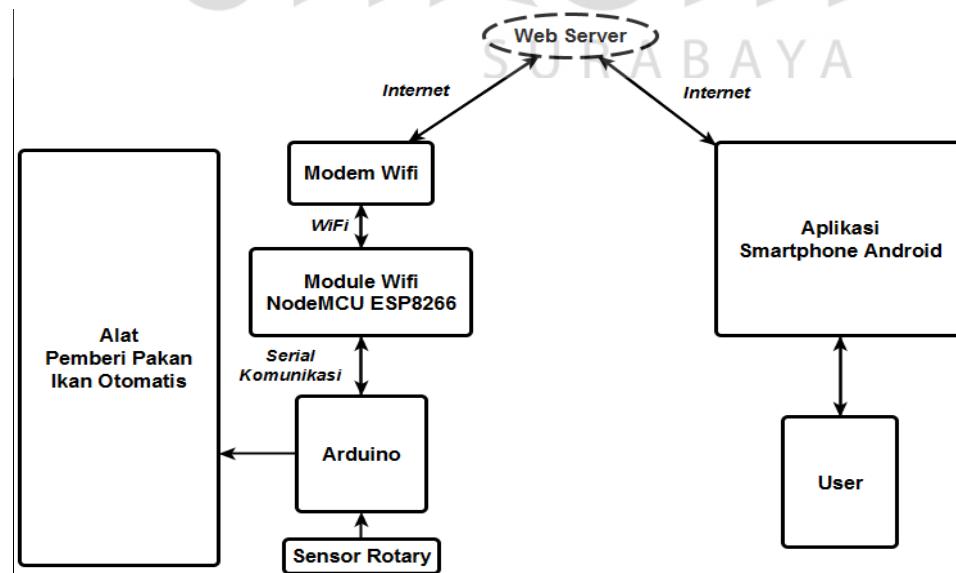
Untuk menu *monitoring* alat akan mengirimkan data *monitoring* kepada *web server*, kemudian aplikasi akan mengakses dan menyimpan data dari *web server* tersebut. Untuk tombol *Request/cek* kerusakan, aplikasi akan mengirimkan data *Request* kepada *web server*, kemudian alat akan mengakses dan menyimpan data dari *web server* tersebut. Pengiriman dan pengambilan data antara aplikasi *smartphone Android* dan Alat Pemberi Pakan Ikan Otomatis dilakukan melalui *internet*. Maka dari itu, dari segi perangkat *smartphone Android* maupun perangkat pada alat wajib terkoneksi dengan *internet*, serta dapat mengakses *internet*.

Pada Alat Pemberi Pakan Ikan Otomatis terdapat *module WiFi ESP8266* yang akan terhubung dengan *modem WiFi*. *Modul WiFi ESP8266* berfungsi sebagai pengirim dan pembaca data dari *web server* melalui *internet*. Untuk komunikasi antar perangkat pada Alat Pemberi Pakan Ikan Otomatis, yaitu *modul WiFi ESP8266* dan *Arduino* sebagai kontroller alat menggunakan komunikasi *serial*

(UART). *Modem WiFi* pada alat berfungsi sebagai perangkat tambahan agar *arduino* dan *module esp8266* dapat terhubung dengan *wifi* untuk mendapatkan *IP Address* agar dapat mengakses *internet* dan *web server*. *Modem WiFi* yang digunakan pada alat adalah jenis *MiFi (Mobile WiFi)* atau *modem wifi* yang sejenis. *MiFi* adalah sebuah perangkat *wireless router* yang berperan sebagai *WiFi Hotspot*. Penggunaan *MiFi* yang simpel dan praktis bertujuan supaya alat dapat ditempatkan dimanapun asalkan masih terdapat jaringan GSM maupun CDMA di area tersebut, serta jika nantinya terdapat penambahan alat pada satu lokasi yang sama atau masih dalam jangkauan *MiFi* tersebut maka tidak perlu adanya penambahan *Modem WiFi* atau *MiFi* lain lagi, jadi cukup menggunakan satu buah *Modem WiFi* atau *MiFi* saja

3.2 Model Perancangan

Pada rancangan penggerjaan Tugas Akhir ini penggambaran perancangan sistem seperti pada gambar 3.1 berikut :



Gambar 3.1 Perancangan Sistem

Perancangan sistem pada gambar 3.1 dapat dilihat bahwa dari segi aplikasi *smartphone Android* maupun dari segi Alat Pemberi Pakan Ikan Otomatis, keduanya menggunakan media *internet* sebagai sarana untuk mengakses data pada *web server*. *Arduino* pada segi alat berfungsi sebagai kontroller dari Alat Pemberi Pakan Ikan Otomatis yang berkomunikasi dengan *module WiFi NodeMCU ESP8266* menggunakan komunikasi *serial* untuk saling berkirim data. Sedangkan untuk *module WiFi ESP8266* berfungsi sebagai *module tambahan* agar *Arduino* dapat terhubung dengan *internet* supaya dapat mengakses data pada *web server*. *Module WiFi ESP8266* membutuhkan koneksi dengan *WiFi* agar dapat mengakses *internet*, disinilah fungsi dari *modem WiFi* pada Alat Pemberi Pakan Ikan Otomatis.

Proses pengiriman dan penerimaan data pada sistem ini dapat dikatakan terpisah. Untuk pengiriman data dari Alat Pemberi Pakan Ikan Otomatis maupun aplikasi *android* akan mengirimkan data *input* maupun data sensor kepada *web server*. Untuk penerimaan data pada Alat Pemberi Pakan Ikan Otomatis maupun aplikasi *android*, keduanya akan mengakses, menyimpan dan memproses data yang didapatkan dari *web server*. Proses pengiriman data dari Alat Pemberi Pakan Ikan Otomatis berawal dari *Arduino Mega 2560* yang mengirimkan data *input* maupun sensor kepada *module WiFi ESP8266*, selanjutnya data diproses dan dikirimkan kepada *web server* oleh *module WiFi ESP8266*. Untuk proses pengiriman data dari aplikasi *Android*, data yang telah didapatkan dari *input user* akan dikirimkan oleh aplikasi *android* kepada *web server*.

Sedangkan proses penerimaan data dari Alat Pemberi Pakan Ikan Otomatis berawal dari *module WiFi ESP8266* mengakses, menyimpan dan memproses data

dari *web server*, kemudian dikirimkan kepada *Arduino Mega 2560*, selanjutnya data diproses untuk dijadikan informasi pada Alat Pemberi Pakan Ikan Otomatis. Untuk proses penerimaan data pada aplikasi *Android*, aplikasi *Android* akan mengakses, menyimpan dan memproses data dari *web server*, kemudian data akan dijadikan informasi untuk *user*.

3.2.1 Perancangan Data Pada Sistem

Pada tugas akhir ini secara garis besar terdapat dua data, yaitu data *monitoring*/pemantauan dan data *setting*/pengaturan. Untuk data *monitoring* dibagi menjadi dua, yaitu data *monitoring* kerusakan dan data *monitoring* ketersediaan pakan pada Alat Pemberi Pakan Ikan Otomatis. Sedangkan untuk data *setting*/pengaturan dibagi menjadi dua, yaitu *scheduling*/penjadwalan dan takaran pemberian pakan.

Data *monitoring* kerusakan dibagi menjadi dua data utama, yaitu data Motor1 dan Motor2. Data tersebut didapatkan dari sensor *rotary* yang terdapat pada Motor DC1 sebagai penggerak untuk mengeluarkan pakan dari penampung, dan Motor DC2 yang berfungsi sebagai kipas untuk melemparkan pakan ikan pada kolam. Untuk data *monitoring* ketersediaan pakan dibagi menjadi dua data utama, yaitu data berat pakan dan status pakan. Data tersebut didapatkan dari hasil selisih dan persentase dari *input* berat pakan yang dilakukan oleh *user* pada alat dengan data pakan yang telah dikeluarkan oleh alat. Sedangkan untuk data *scheduling*/penjadwalan dibagi menjadi empat data, yaitu data jadwal1, jadwal2, jadwal3 dan jadwal4.

Dari beberapa data tersebut, semua data yang diperlukan akan dijadikan satu data yang bertipe *string*, kemudian satu data tersebut yang akan diterima maupun dikirimkan oleh Alat Pemberi Pakan Ikan Otomatis maupun oleh aplikasi *Android* kepada *web server*. Dapat dilihat pada format data *server* terdapat terdapat beberapa karakter seperti “ _ “, “ . “ dan “ : “. Karakter-karakter tersebut berfungsi sebagai pemisah setiap data untuk memudahkan pemrosesan data pada Alat Pemberi Pakan Ikan Otomatis maupun pada aplikasi *Android*. Penjelasan dan fungsi dari setiap data dan karakter dapat dilihat pada tabel 3.1 dan format data seperti dibawah ini.

OFF_14: 56.0:0.0:0.0:0_0.0.1000.1000.40.N

Tabel 3.1 Penjelasan Data *Server*

No	Data	Keterangan
1	-	Pemisah antara data <i>ON</i> / <i>OFF</i> , data <i>scheduling</i> /penjadwalan dan data <i>monitoring</i> /pemantauan
2	.	Pemisah setiap satu data (<i>ON</i> / <i>OFF</i> , Jadwal1, Jadwal2, Jadwal3, Jadwal4, Motor DC1, Motor DC2, Berat Pakan Awal, Berat Pakan Sekarang, Takaran dan <i>Request</i>)
3	:	Pemisah antara data Jam dan data Menit pada setiap data <i>scheduling</i> /penjadwalan
4	<i>ON</i> / <i>OFF</i>	<i>ON</i> / <i>OFF</i> menandakan waktunya pemberian pakan atau tidak sesuai penjadwalan
5	Jadwal	Penjadwalan (Jadwal1, Jadwal2, Jadwal3 dan Jadwal4)
6	Motor	Motor DC1 dan Motor DC2 (1 menandakan Motor bergerak dan 0 menandakan Motor tidak bergerak)
7	Berat Pakan Awal	Berat pakan awal yang berada pada penampung pakan atau berat pakan <i>input</i> dari <i>user</i> pada alat
8	Berat Pakan Sekarang	Berat pakan sekarang yang berada pada penampung pakan atau berat pakan yang sudah dihitung
9	Takaran	Takaran pemberian pakan setiap satu kali penjadwalan
10	<i>Request</i>	Adanya <i>request</i> /permintaan untuk memeriksa kerusakan atau tidak (N menandakan adanya <i>request</i> dan M menandakan tidak ada <i>request</i>)

3.2.2 Perancangan Proses Pemantauan Kerusakan

Pada tugas akhir ini pemantauan kerusakan pada Alat Pemberi Pakan Ikan Otomatis terdapat dua proses, yaitu proses data pada Alat Pemberi Pakan Ikan Otomatis dan proses data pada aplikasi *Android*. Pemantauan kerusakan pada tugas akhir ini hanya mencakup kerusakan yang terjadi pada Motor DC1 sebagai penggerak untuk mengeluarkan pakan ikan dari tempat penampung pakan dan Motor DC2 penggerak kipas untuk pelempar pakan ikan pada kolam ikan. Data kerusakan pada Motor DC1 dan Motor DC2 didapatkan dari sensor *rotary* yang terpasang pada setiap Motor.

Proses pemantauan kerusakan pada alat berawal dari pengambilan data dari sensor *rotary* pada alat, data tersebut akan diproses oleh *Arduino Mega 2560*, kemudian akan dikirimkan kepada *module WiFi NodeMCU ESP8266* dan selanjutnya akan dikirimkan oleh *module WiFi NodeMCU ESP8266* kepada *web server*. Untuk proses data pada aplikasi *Android*, aplikasi *Android* akan mengakses, menyimpan dan memproses data dari *web server*. Selanjutnya data tersebut akan dijadikan informasi Motor DC1 dan Motor DC2 bergerak atau tidak serta informasi apakah terjadi kerusakan pada alat atau tidak kepada *user*. Apabila terindikasi terjadi kerusakan pada alat, *user* akan mendapatkan notifikasi atau pemberitahuan pada aplikasi *Android* bahwa terjadi kerusakan pada Motor.

3.2.3 Perancangan Proses Pemantauan Ketersediaan Pakan

Pada tugas akhir ini pemantauan ketersediaan pakan pada tempat penampung pakan pada Alat Pemberi Pakan Ikan Otomatis terdapat dua proses,

yaitu proses data pada Alat Pemberi Pakan Ikan Otomatis dan proses data pada aplikasi *Android*. Pemantauan ketersediaan pakan pada tugas akhir ini hanya mencakup Berat Pakan dan Status Pakan pada alat. Data didapatkan dari proses penghitungan selisih antara data Berat Pakan yang telah dimasukkan oleh *user* pada alat dengan Berat Pakan yang telah dikeluarkan oleh alat.

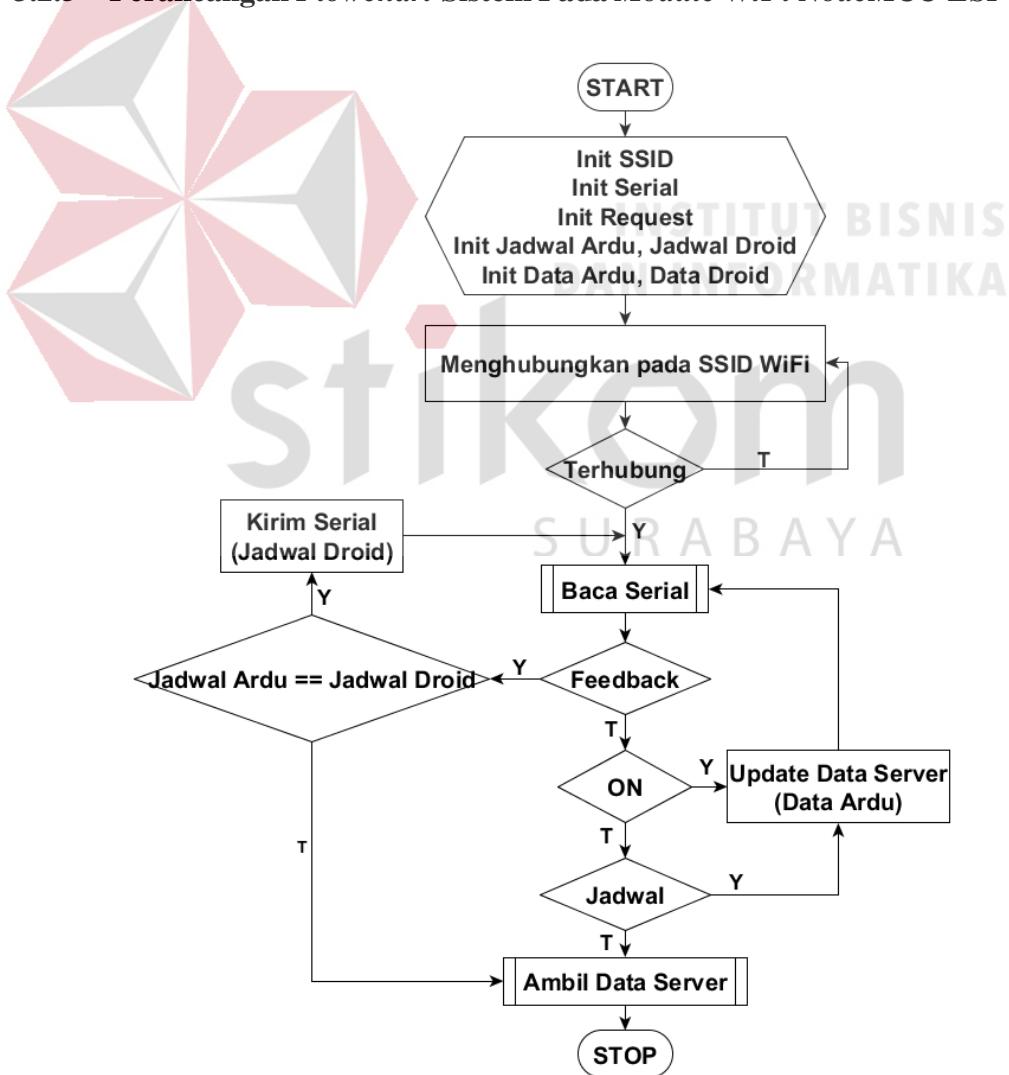
Proses pemantauan kerusakan pada alat berawal dari pengambilan data dari *input user* pada alat, data tersebut akan diproses oleh *Arduino Mega 2560*, kemudian akan dikirimkan kepada *module WiFi NodeMCU ESP8266* dan selanjutnya akan dikirimkan oleh *module WiFi NodeMCU ESP8266* kepada *web server*. Untuk proses data pada aplikasi *Android*, aplikasi *Android* akan mengakses, menyimpan dan memproses data dari *web server*. Selanjutnya data tersebut akan dijadikan informasi ketersediaan pakan pada alat kepada *user*, khususnya Berat Pakan dan Persentase Pakan yang tersisa pada tempat penampung pakan pada alat. Apabila terindikasi bahwa Status atau Persentase Pakan pada alat $< 10\%$, *user* akan mendapatkan notifikasi atau pemberitahuan pada aplikasi *Android* bahwa pakan ikan tersisa $X\%$.

3.2.4 Perancangan Proses Penjadwalan Pemberian Pakan

Pada tugas akhir ini penjadwalan pemberian pakan pada Alat Pemberi Pakan Ikan Otomatis dari aplikasi *Android* terdapat dua proses, yaitu proses data pada aplikasi *Android* dan proses data pada Alat Pemberi Pakan Ikan Otomatis. Data penjadwalan pemberian pakan didapatkan dari *input* yang dilakukan oleh *user*.

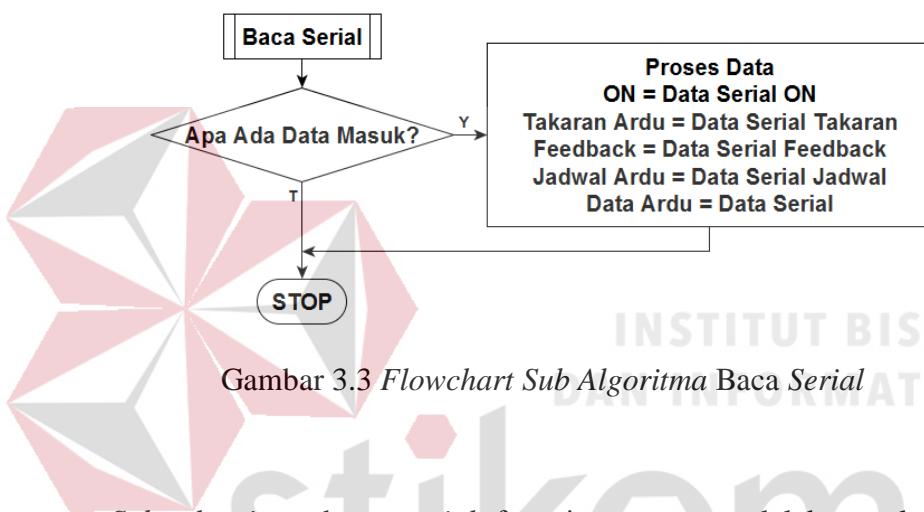
Proses penjadwalan pemberian pakan pada alat berawal dari *input* yang dilakukan oleh *user* pada aplikasi *Android*, kemudian data akan diproses dan dikirimkan kepada *web server*. Untuk proses data pada alat, *module WiFi NodeMCU ESP8266* akan mengakses, menyimpan dan memproses data dari *web server*. Selanjutnya data tersebut akan dikirimkan kepada *Arduino Mega 2560* untuk dijadikan data penjadwalan pada Alat Pemberi Pakan Ikan Otomatis.

3.2.5 Perancangan *Flowchart* Sistem Pada *Module WiFi NodeMCU ESP8266*



Gambar 3.2 *Flowchart* *Module WiFi NodeMCU ESP8266*

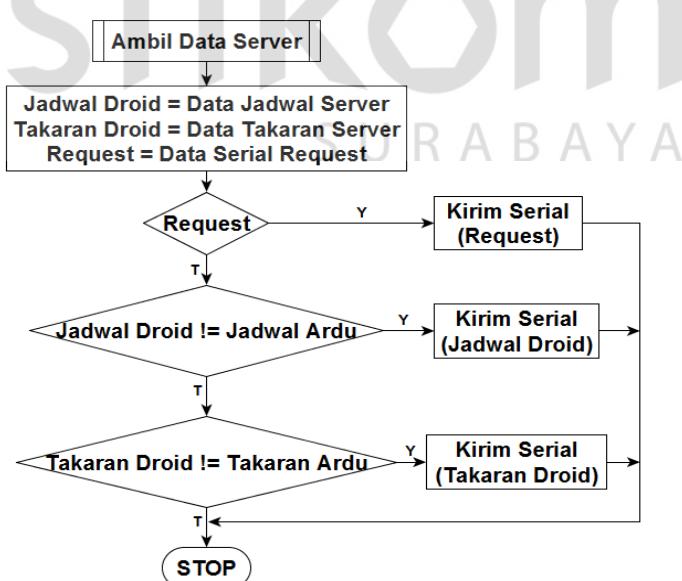
Pada gambar 3.2 adalah alur cara kerja dari *module WiFi ESP8266* pada alat. Proses dimulai dengan inisialisasi, selanjutnya proses akan menghubungkan *module WiFi ESP8266* dengan SSID dari *modem WiFi* yang telah ditentukan. Jika *module WiFi ESP8266* berhasil terhubung dengan *modem WiFi*, proses akan melanjutkan pemanggilan *sub algoritma baca serial*. *Flowchart* dari *sub algoritma baca serial* dapat dilihat pada gambar 3.3.



Sub algoritma baca serial fungsi utamanya adalah untuk menerima, menyimpan dan memproses data *serial* yang diterima dari *Arduino*. Jika ada data yang masuk maka data akan diproses untuk disimpan, dipilah dan diseleksi menjadi beberapa data, yaitu data *ON*, *Feedback*, *Jadwal Ardu* dan *Data Ardu*. Setelah selesai, proses kembali kepada proses utama pada *module WiFi ESP8266*. Proses selanjutnya setelah proses baca *serial* adalah melakukan pencocokan apakah data yang diterima oleh *serial* adalah data *Feedback* atau bukan. Secara keseluruhan, *Feedback* ini betujuan untuk memeriksa data yang telah dikirimkan oleh *module WiFi ESP8266* kepada *Arduino* sudah benar sama atau tidak, jika data belum sama maka pengiriman dilakukan sampai data sama.

Jika data yang diterima oleh serial bukan data *Feedback*, maka akan diperiksa apakah data yang diterima adalah data ON atau bukan. Jika data ON, maka *module WiFi ESP8266* akan mengirimkan atau *update* Data Ardu yang diterima oleh *serial* pada *web server*. Ketika alat ON, maka proses akan berulang dari *sub algoritma baca serial* hingga data yang diterima bukan ON, yaitu OFF. Proses ini bertujuan untuk mengirimkan data kepada *web server* secara berulang selama waktunya pemberian pakan pada alat.

Jika data yang diterima bukan ON, maka akan diperiksa apakah data yang diterima adalah data Jadwal atau bukan. Jika benar, maka jadwal yang diterima akan di kirimkan dan di *update* pada *web server*. Proses ini bertujuan apabila terjadi perubahan jadwal pada alat. Tetapi jika bukan data Jadwal, maka proses akan memanggil *sub algoritma Ambil Data Server*.



Gambar 3.4 Flowchart Sub Algoritma Ambil Data Server

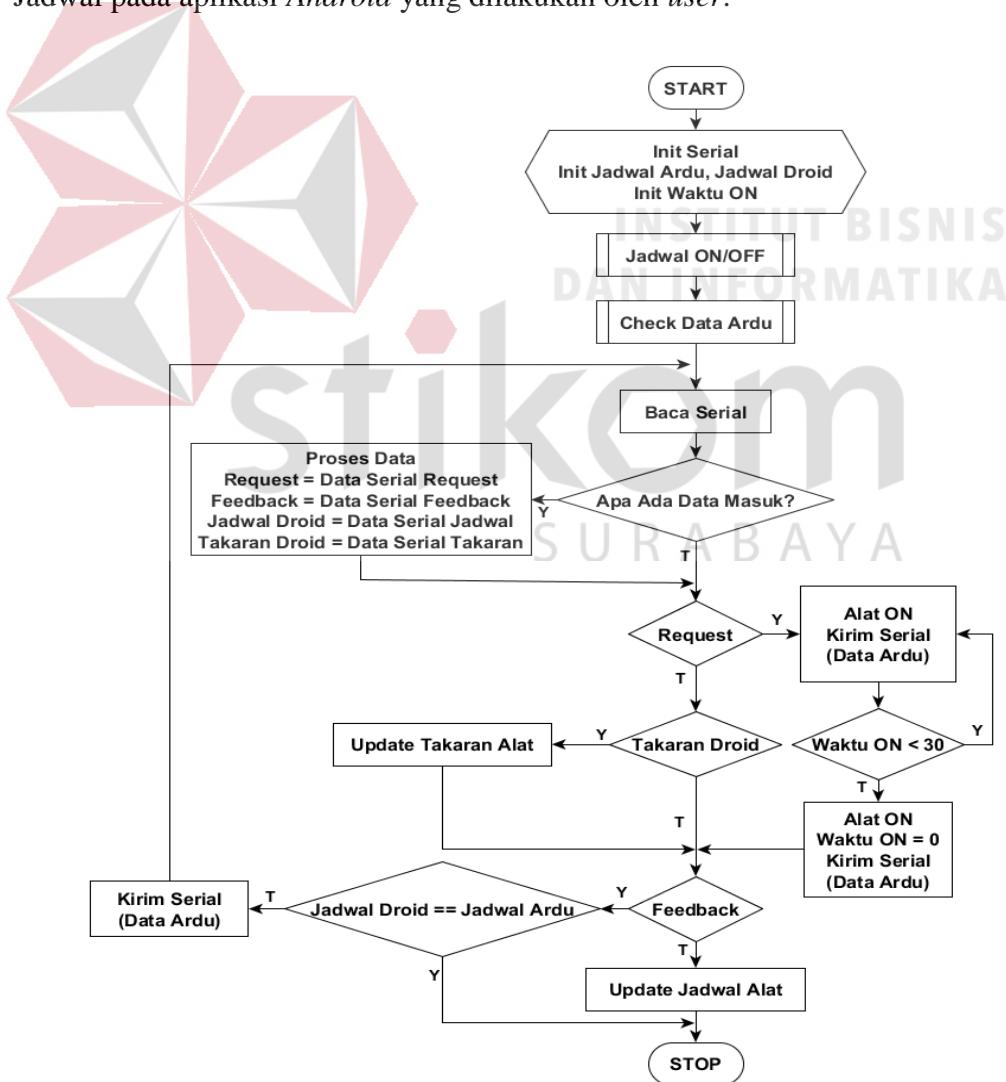
Flowchart sub algoritma Ambil Data Server dapat dilihat pada gambar 3.4 diatas. Pada proses ini *module WiFi ESP8266* akan mengakses, meyimpan dan memproses data dari *web server*. Selanjutnya akan dilakukan pemeriksaan, apakah data yang didapatkan adalah data *request*, data Jadwal atau data Takaran. Kemudian data akan dikirimkan kepada *Arduino* sesuai data yang didapatkan dari *web server*. Secara keseluruhan, proses ini bertujuan untuk memeriksa apakah terdapat perubahan data pada *web server* yang dilakukan oleh aplikasi *Android* atau tidak.

3.2.6 Perancangan *Flowchart* Sistem Pada *Arduino*

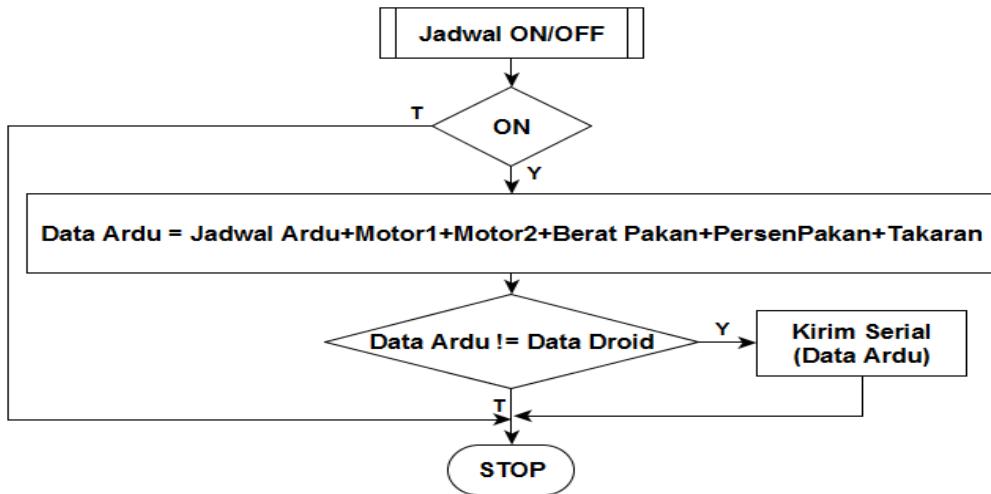
Flowchart Sistem Pada *Arduino* adalah alur cara kerja *Arduino* pada alat. Awal mula proses adalah inisialisasi, proses memanggil *sub algoritma* Jadwal ON/OFF, setelah itu proses memanggil *sub algoritma* *Check Data Ardu*. Setelah proses selesai dilaksanakan, selanjutnya adalah proses Baca *Serial*. Pada proses ini jika tidak ada data yang diterima maka proses berlanjut pada proses selanjutnya. Tetapi jika ada data yang diterima, maka data akan diproses untuk dipilah dan diseleksi menjadi beberapa data. Kemudian data yang didapatkan akan diperiksa apakah data yang diterima adalah data *Request*, *Takaran*, *Feedback* atau *Jadwal*.

Apabila data yang diterima adalah *Request*, maka *Arduino* akan menjalankan Motor penggerak pada alat atau ON selama waktu yang telah ditentukan, serta *Arduino* akan mengirimkan Data Ardu kepada *module WiFi ESP8266*. Proses *Request* ini berfungsi untuk memeriksa kerusakan pada alat. Apabila data yang diterima adalah *Takaran*, maka *Arduino* akan memperbarui atau *update* data *Takaran* pemberian pakan pada alat. Apabila data yang diterima adalah

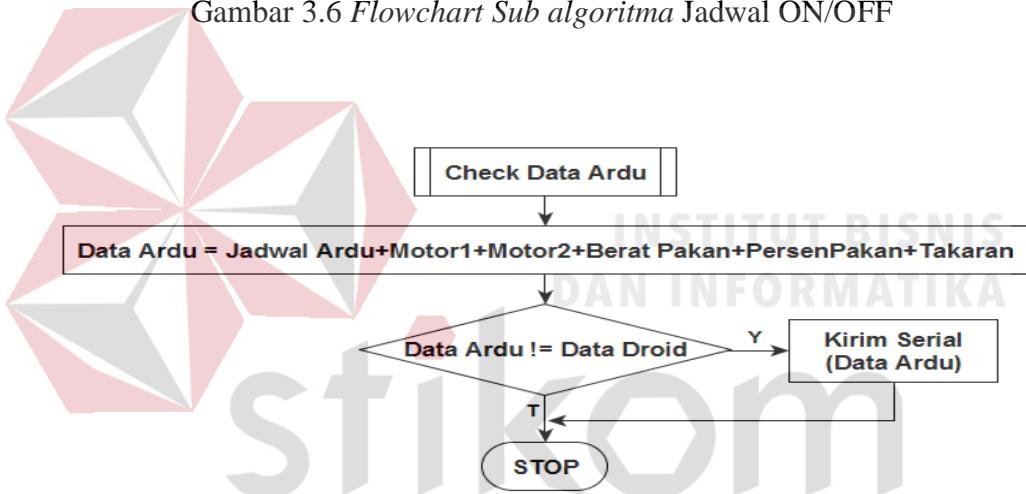
Feedback, maka *Arduino* akan memeriksa apakah data yang dikirimkan oleh *Arduino* kepada *module ESP8266* sudah sama dengan data yang diterima oleh *module ESP9266* atau tidak. Jika data belum sama, maka data akan dikirimkan ulang kepada *module ESP9266*. Proses *Feedback* ini berfungsi untuk memeriksa keakuratan data antara *Arduino* dengan *module ESP9266*. Apabila data yang diterima adalah Jadwal, maka *Arduino* akan memperbarui atau *update* data Jadwal pemberian pakan pada alat. Proses ini berfungsi apabila terdapat perubahan data Jadwal pada aplikasi *Android* yang dilakukan oleh *user*.



Gambar 3.5 Flowchart *Arduino*



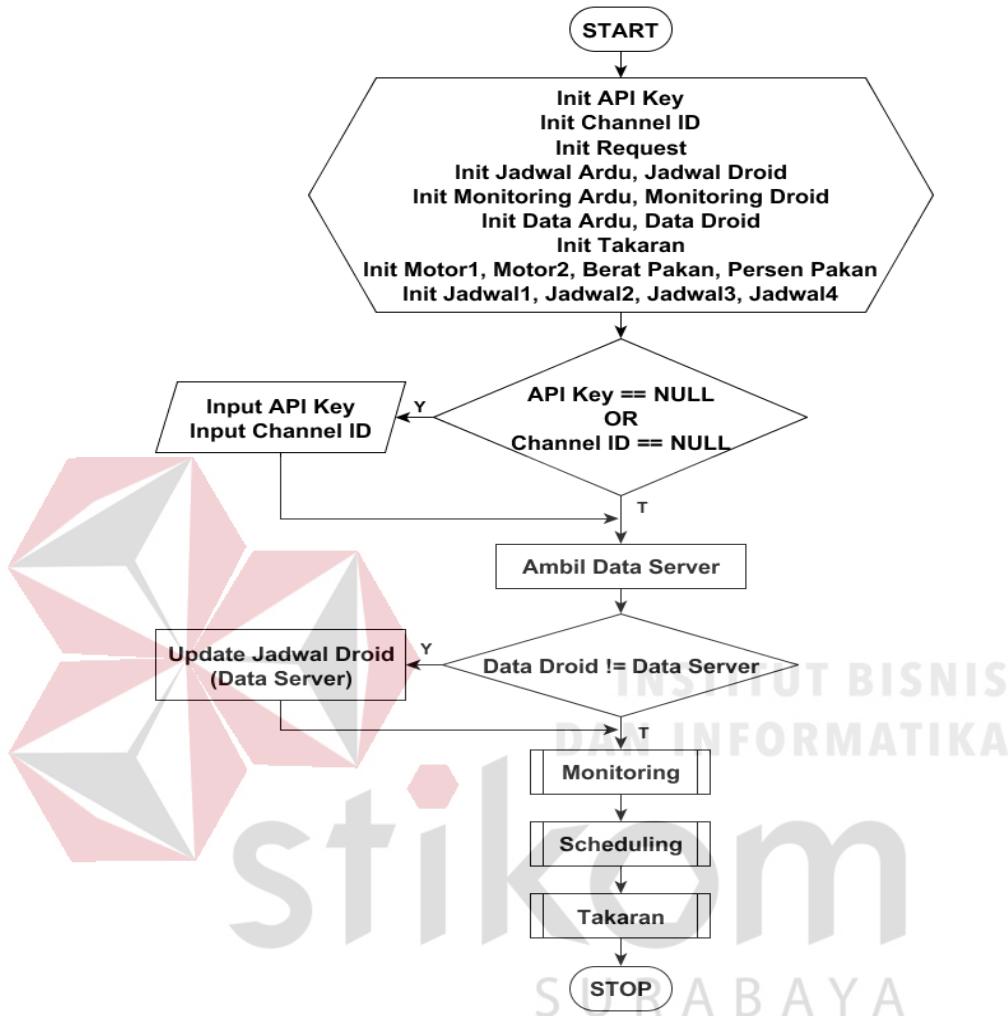
Gambar 3.6 Flowchart Sub algoritma Jadwal ON/OFF



Gambar 3.7 Flowchart Sub Algoritma Check Data Arduino

Pada *sub algoritma* jadwal ON/OFF bertugas pada saat alat ON. *Arduino* akan mengirimkan data yang dibutuhkan kepada *module ESP8266*. *Flowchart* dari *sub algoritma* Jadwal ON/OFF dapat dilihat pada gambar 3.6. Sedangkan untuk *sub algoritma* *check data Arduino* bertugas saat ada perubahan data pada alat, seperti perubahan jadwal, berat pakan maupun kerusakan, selanjutnya data akan dikirimkan kepada *module ESP8266*. *Flowchart* dari *sub algoritma* baca serial dan *sub algoritma* *Check Data Arduino* dapat dilihat pada gambar 3.7.

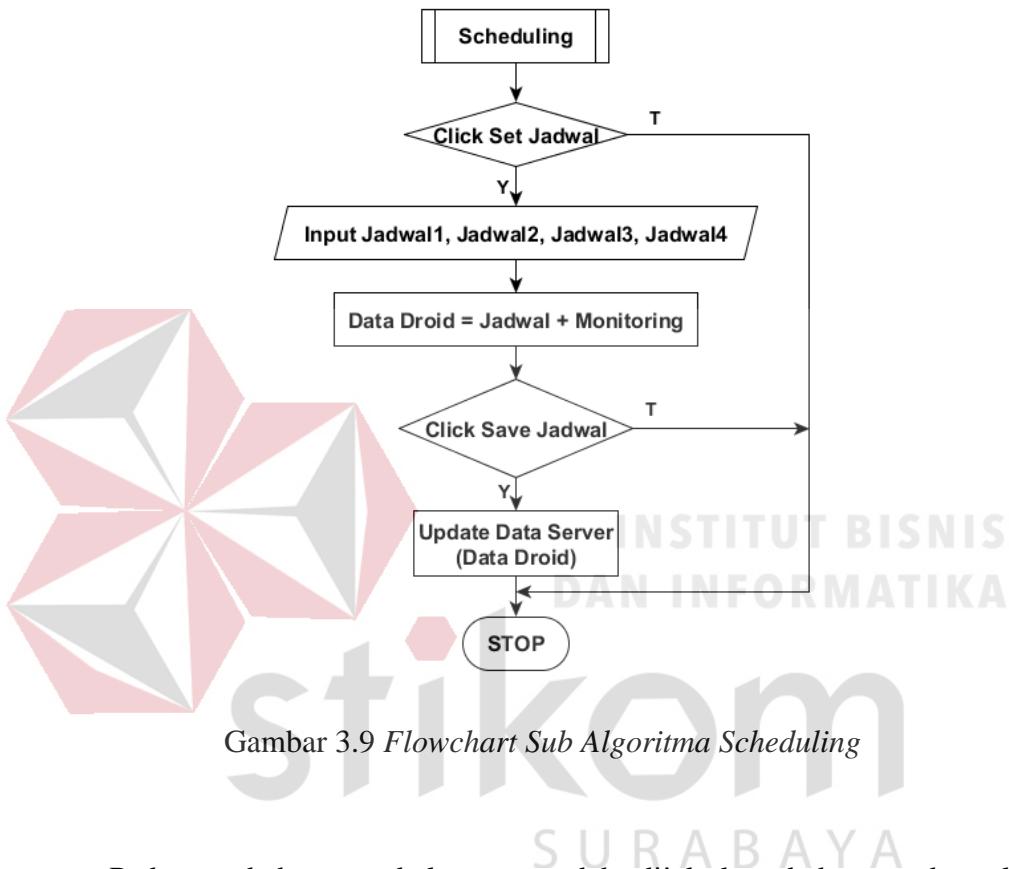
3.2.7 Perancangan Flowchart Sistem Pada Aplikasi *Smartphone Android*



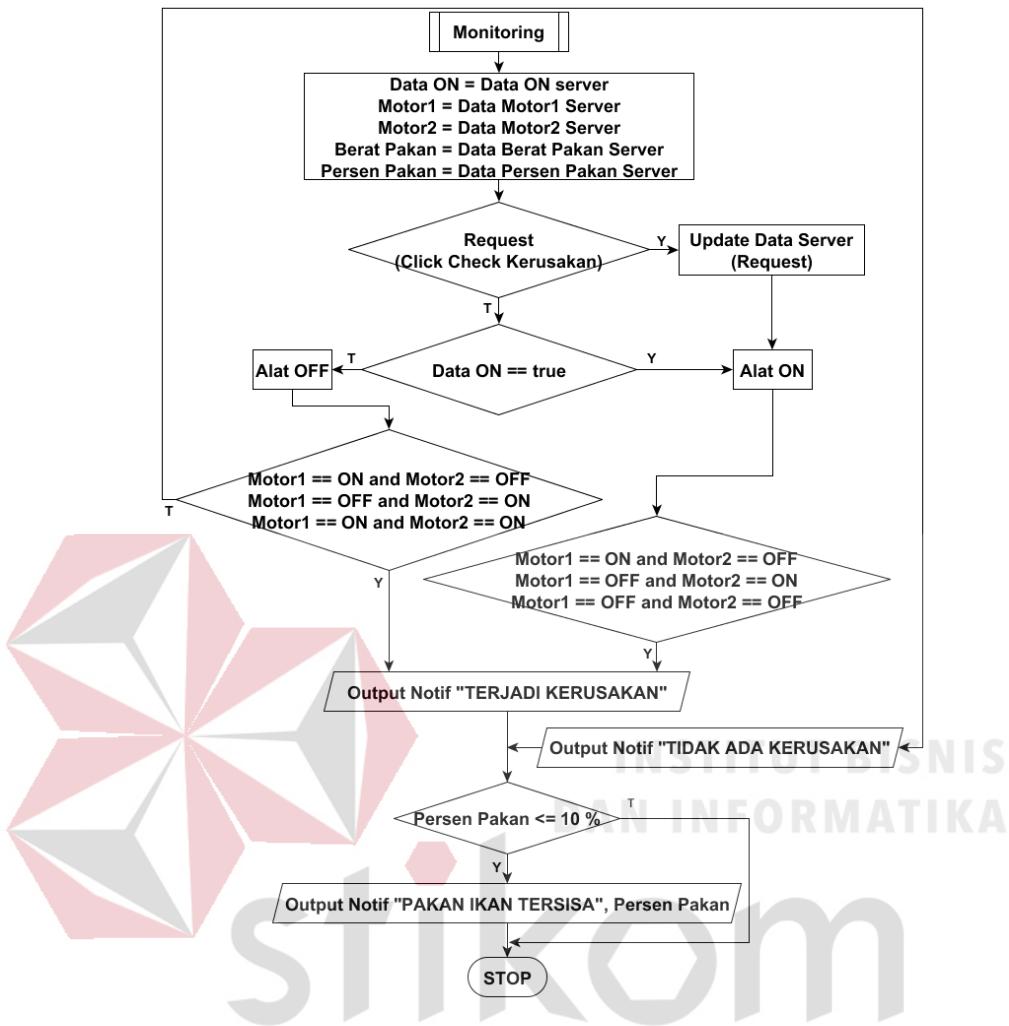
Gambar 3.8 *Flowchart Aplikasi Smartphone Android*

Pada gambar 3.8 adalah alur cara kerja dari aplikasi *Android*. Awal mula proses melakukan inisialisasi. Setelah itu memeriksa apakah *API Key* atau *Channel ID* kosong. Jika benar, maka *user* harus menginputkan *API Key* dan *Channel ID* yang telah ditentukan. Setelah itu masuk pada proses *Ambil Data Server*, proses ini berfungsi untuk mengakses dan mengambil data alat dari *server*. Data *server* akan diproses menjadi beberapa data utama. Jika data yang didapatkan dari *web server*

berbeda dengan data pada *Android*, maka data pada *Android* akan di *update* sesuai dengan data yang didapatkan dari *server*. Selanjutnya proses akan memanggil *sub algoritma Monitoring*, *sub algoritma Scheduling* dan *sub algoritma Takaran*.



Pada pembahasan sebelumnya sudah dijelaskan bahwa pada aplikasi *Android* terdapat 2 menu utama, yaitu menu *monitoring* dan *scheduling*. Untuk menu *scheduling*, *Flowchart* dapat dilihat pada gambar 3.9. *Sub algoritma scheduling* akan bekerja pada saat *user* memilih menu *scheduling*. Saat tombol *Set Jadwal* pada menu *scheduling* ditekan, maka *user* dapat memasukkan penjadwalan sesuai yang diinginkan. Selanjutnya ketika *user* melakukan penekanan tombol *save*, maka data Jadwalan yang telah dimasukkan oleh *user* akan diproses akan di-*update* atau dikirimkan kepada *server*.

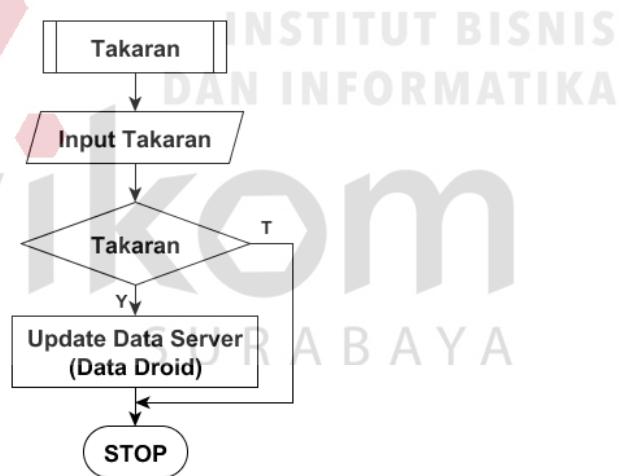


Gambar 3.10 Flowchart Sub Algoritma Monitoring

Untuk menu *monitoring*, *Flowchart* dapat dilihat pada gambar 3.10. Pada menu *monitoring*, alur proses dapat dijabarkan mulai dari proses data yang disertai dengan *update* tampilan data *monitoring*, seperti Motor1, Motor2, Berat Pakan dan Persen Pakan. Selanjutnya dilakukan pemeriksaan apakah ada penekanan tombol *request* atau tidak, jika ada maka aplikasi *Android* akan mengirimkan perintah untuk *Check Kerusakan* kepada *Server*. Apabila tidak ada penekanan pada tombol *request* atau *check kerusakan*, proses akan berlanjut untuk membandingkan apakah

Data ON sama dengan *true* atau tidak. Jika benar maka alat pada saat ini sedang ON atau saatnya pemberian pakan sesuai jadwal, tetapi jika salah maka alat OFF.

Setelah diidentifikasi alat ON atau OFF, dilakukan pemeriksaan apakah Motor1 atau Motor2 berjalan atau tidak, saat motor berjalan dengan benar maka tampilan display pada aplikasi *Android* adalah “TIDAK TERJADI KERUSAKAN”, jika sebaliknya maka display pada aplikasi *Android* adalah “TERJADI KERUSAKAN”, sekaligus pada aplikasi *Android* akan muncul *notifikasi* bahwa “TERJADI KERUSAKAN PADA MOTOR”. Kemudian proses akan memeriksa apakah persen pakan < 10 % atau tidak, jika benar maka *output* “PAKAN IKAN TERSISA KURANG DARI 10 %”.



Gambar 3.11 *Flowchart Sub Algoritma Takaran*

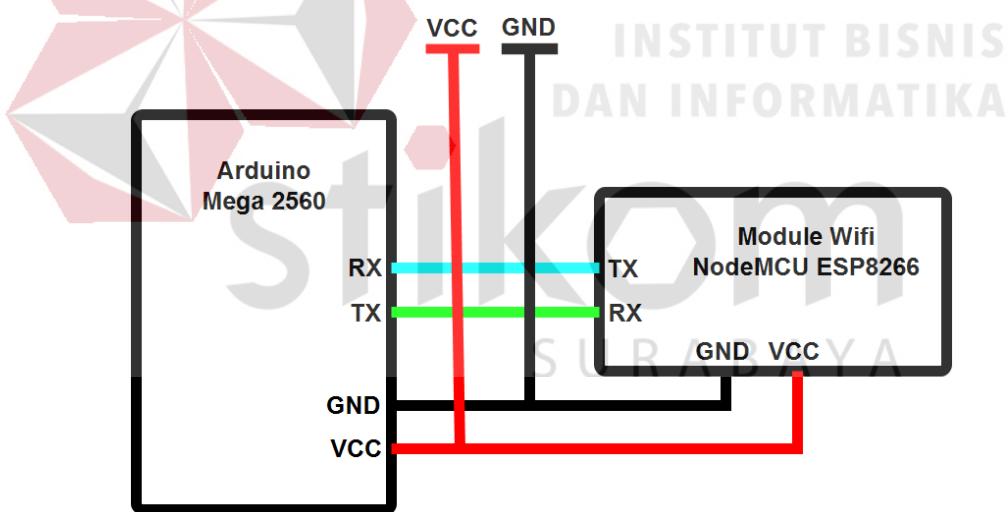
Sub algoritma Takaran akan bekerja pada saat *user* memilih menu *Setting* Takaran. Saat *user* menginputkan nilai takaran pada aplikasi *Android*, maka data Takaran yang telah dimasukkan oleh *user* akan di-update atau dikirimkan kepada *server*. *Flowchart* dari *sub algoritma* *Scheduling* dapat dilihat pada gambar 3.11.

3.3 Perancangan Perangkat Keras

Setelah model perancangan sistem telah selesai dilakukan, selanjutnya penulis akan melakukan perancangan perangkat keras sesuai model perancangan sistem yang telah dibuat seperti perancangan alat keras yang saling terhubung dan terintegrasi.

3.3.1 Pemasangan *Module WiFi ESP8266* pada *Arduino Mega 2560*

Gambar 3.12 dibawah ini merupakan konfigurasi Pin *Module WiFi ESP8266* pada *Arduino Mega*, keduanya saling terhubung melalui PIN TX dan RX masing-masing supaya dapat saling mengirim data secara *serial*.

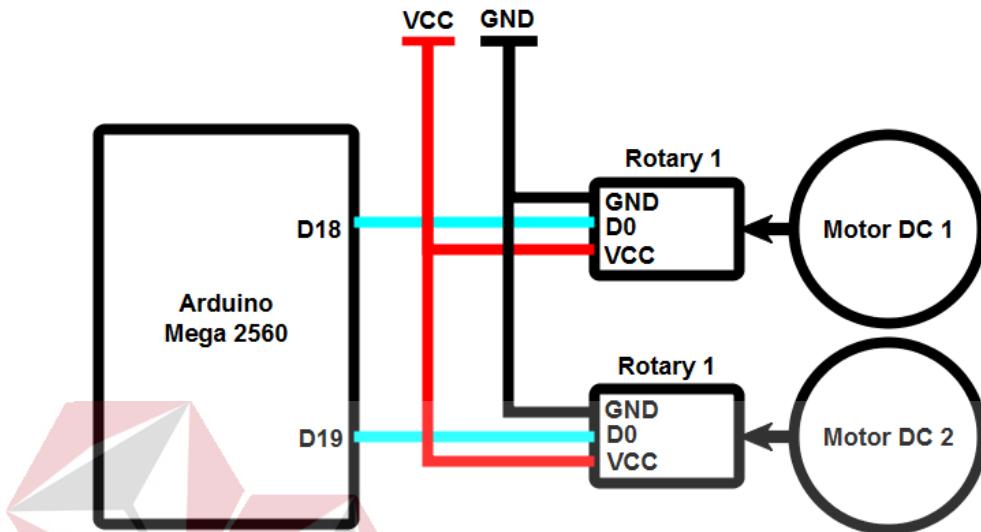


Gambar 3.12 Konfigurasi *Module WiFi ESP8266* Pada *Arduino Mega*

3.3.2 Pemasangan Sensor *Rotary* Pada Motor DC dan *Arduino Mega 2560*

Pada gambar 3.13 adalah konfigurasi sensor *rotary* pada motor DC 1 dan Motor DC 2, serta *output* dari sensor yang dihubungkan dengan Pin Digital *Arduino*

Mega 2560. Sensor *rotary* pada Motor DC berfungsi sebagai sensor indikasi putaran Motor DC 1 maupun Motor DC 2.



Gambar 3.13 Konfigurasi *Rotary* Pada Motor DC dan *Arduino Mega*

3.4 Perancangan Perangkat Lunak

Perancangan perangkat lunak sangatlah penting untuk menentukan alur jalannya program sesuai perancangan perangkat keras yang diimplementasikan pada Tugas Akhir ini. Perancangan perangkat lunak yang akan dilakukan yaitu perancangan desain tampilan aplikasi maupun pemrograman aplikasi *smartphone* *Android* dengan menggunakan aplikasi *App Inventor*, dan prancangan program pada *Module ESP8266* maupun pada *Arduino* menggunakan aplikasi *Arduino IDE*. Selanjutnya adalah penjelasan masing-masing perancangan perangkat lunak.

3.4.1 Perancangan Desain Aplikasi *Smartphone Android*

Perancangan perangkat lunak pada tugas akhir ini diawali dengan pembuatan desain aplikasi *Smartphone Android* yang akan digunakan untuk

monitoring dan *scheduling* alat pemberi pakan ikan otomatis. Pembuatan aplikasi *Android* ini menggunakan aplikasi berbasis *web* yaitu *App Inventor* yang diharuskan menggunakan koneksi *internet* untuk mengaksesnya. Desain aplikasi ini terdiri dari beberapa *Form* yang memiliki desain dan fungsi masing-masing.

3.4.1.1 *Form Input Set*

Saat pertama kali memulai aplikasi, tampilan desain yang dibuat yaitu *Form Input Set*, berfungsi untuk menginputkan *API Key* maupun *Channel ID*. *Form* ini hanya akan aktif jika *API Key* maupun *Channel ID* masih kosong. Tampilan dari *Form* ini dapat dilihat pada gambar 3.14 dibawah ini.



Gambar 3.14 Desain *Form Input Set*

3.4.1.2 *Form Loading*

Pada gambar 3.15 adalah desain dari *Form Loading*. *Form Loading* ini berfungsi sebagai *delay* pada saat aplikasi sedang melakukan inisialisasi,

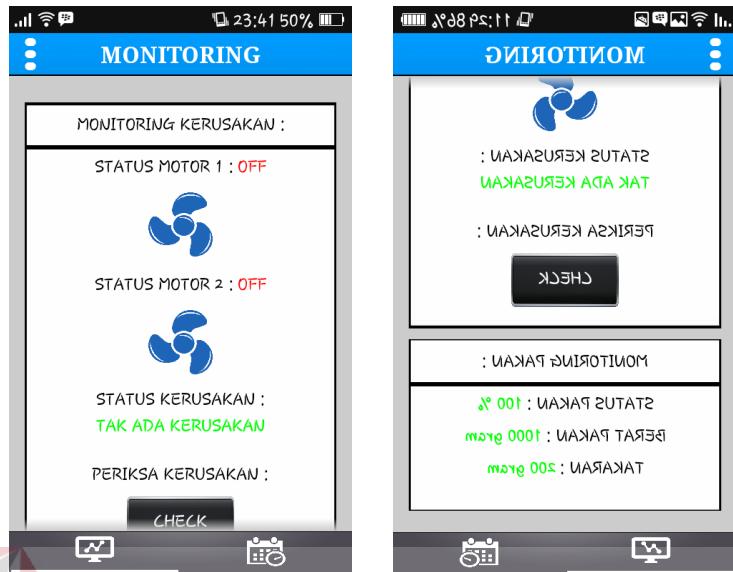
pengambilan data dari *database* aplikasi, pengaksesan dan pengambilan data dari *server* serta pemrosesan data hingga menjadi informasi ygng siap untuk *user*.



Gambar 3.15 Desain Form Loading

3.4.1.3 Form Monitoring

Selanjutnya setelah proses pada *Form Loading* selesai, maka aplikasi akan masuk pada salah satu menu utama, yaitu menu *monitoring*. Pada *Form Monitoring* ini dibagi menjadi 2 bagian utama, yaitu *Monitoring Kerusakan* dan *Monitoring Pakan*. Pada bagian *Monitoring Pakan* memiliki informasi Berat Pakan, Persen Pakan dan Takaran. Sedangkan pada bagian *Monitoring Kerusakan* memiliki informasi Status Motor 1, Status Motor 2, Status Kerusakan dan satu tombol Periksa Kerusakan.



Gambar 3.16 Desain *Form Monitoring*

Tombol Periksa Kerusakan yang ada pada menu inilah yang dimaksud pada pembahasan sebelum-sebelumnya sebagai tombol *request*, dimana berfungsi untuk memeriksa kerusakan secara manual pada alat. Tampilan dari *Form* ini dapat dilihat pada gambar 3.16.

3.4.1.4 *Form Scheduling*

Form Scheduling ini memiliki *display* informasi Jadwal Pemberian Pakan dan satu buah tombol Atur Jadwal. Pada *display* informasi berfungsi untuk menampilkan jadwal yang telah diinputkan oleh *user* pada alat maupun aplikasi. Sedangkan tombol Atur Jadwal berfungsi untuk mengatur, merubah, menambah maupun menghapus jadwal. Ketika ada penekanan tombol Atur Jadwal maka tampilan akan menuju *Form Set Scheduling*. Pada gambar 3.17 dan 3.18 dapat dilihat tampilan *Form Scheduling* pada saat ada jadwal dan tidak ada jadwal.



Gambar 3.17 Desain Form Scheduling



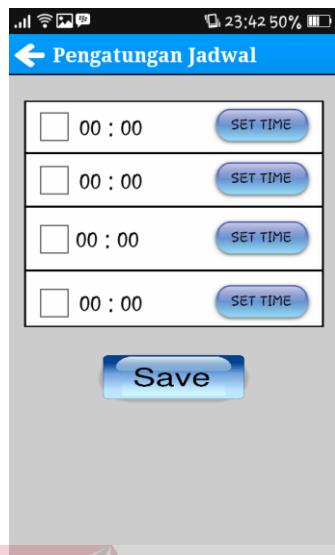
Gambar 3.18 Desain Form Scheduling

Saat Tidak Ada Jadwal

Saat Ada Jadwal

3.4.1.5 Form Set Scheduling

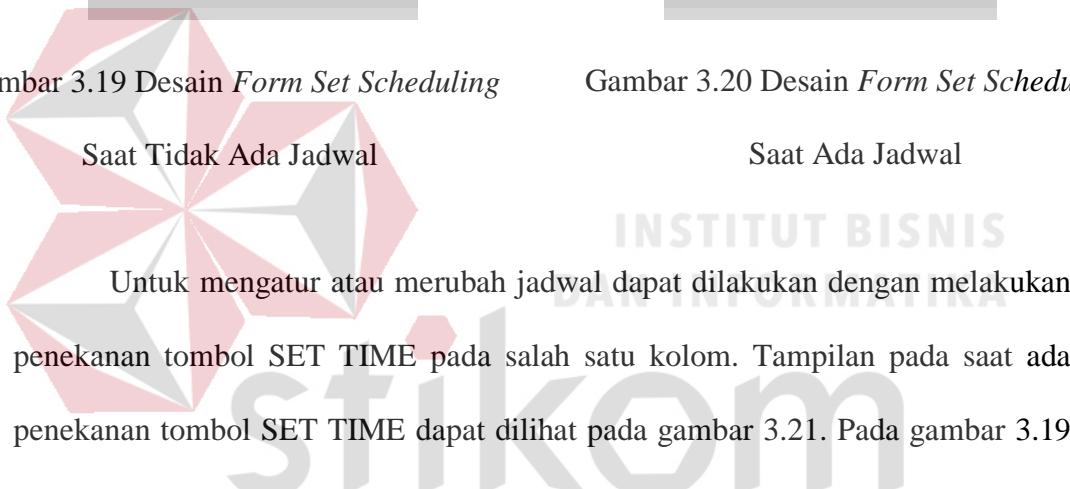
Form Set Scheduling akan tampil ketika ada penekanan tombol Atur Jadwal yang tersedia pada *Form Scheduling*. Pada *Form* ini *user* dapat melakukan pengaturan terhadap jadwal pemberian pakan pada alat sesuai keinginan, seperti menambah, merubah, mengakifkan dan menonaktifkan jadwal. Pada *Form* ini terdapat satu tabel yang terdiri dari empat kolom dengan beberapa pengaturan tertentu pada setiap kolom. Serta satu tombol SAVE yang berfungsi untuk menyimpan pengaturan yang sudah dilakukan.



Gambar 3.19 Desain Form Set Scheduling



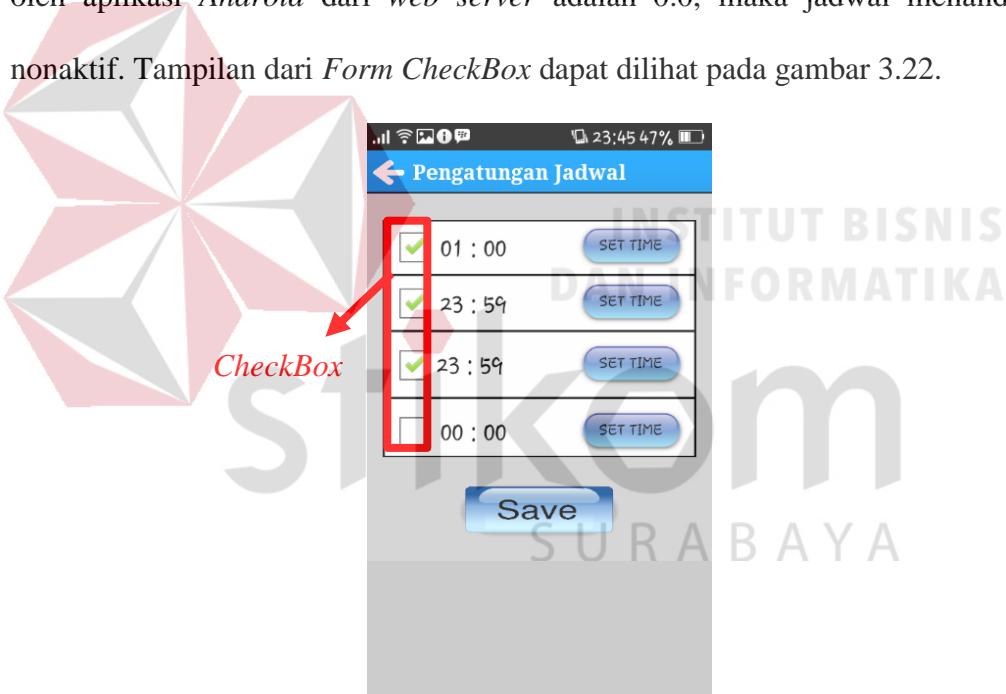
Gambar 3.20 Desain Form Set Scheduling



Gambar 3.21 Tampilan Pengaturan atau Perubahan Jadwal

3.4.1.6 CheckBox

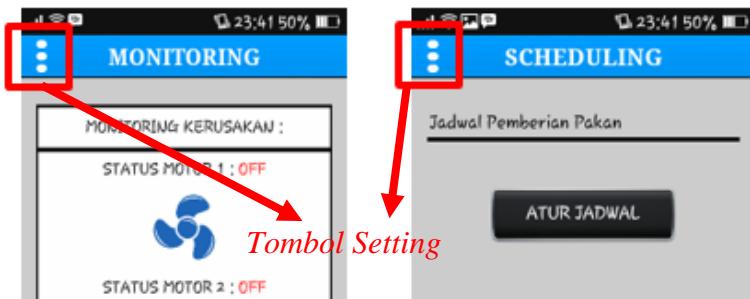
CheckBox pada *Form Set Scheduling* berfungsi untuk mengaktifkan ataupun menonaktifkan jadwal. Apabila *user* ingin menonaktifkan jadwal pemberian pakan pada alat, *user* hanya perlu menghilangkan *CheckList* pada *CheckBox*. Sebaliknya ingin mngaktifkan jadwal, *user* hanya perlu melakukan *CheckList* pada *CheckBox*. *CheckBox* ini akan bekerja secara otomatis mengikuti data penjadwalan yang didapatkan dari *web server*. Apabila jadwal yang didapatkan oleh aplikasi *Android* dari *web server* adalah 0:0, maka jadwal menandakan nonaktif. Tampilan dari *Form CheckBox* dapat dilihat pada gambar 3.22.



Gambar 3.22 *CheckBox* Pada *Form Set Scheduling*

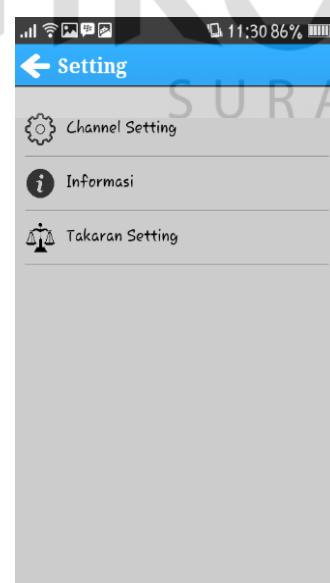
3.4.1.7 Form Setting

Form Setting akan tampil ketika ada penekanan tombol *setting* yang berada di pojok kiri atas pada saat berada di *Form Monitoring* atau *Form Scheduling*. Tombol *setting* dapat dilihat pada gambar 3.23 pada tanda merah.



Gambar 3.23 Desain Tombol Untuk *Form Setting*

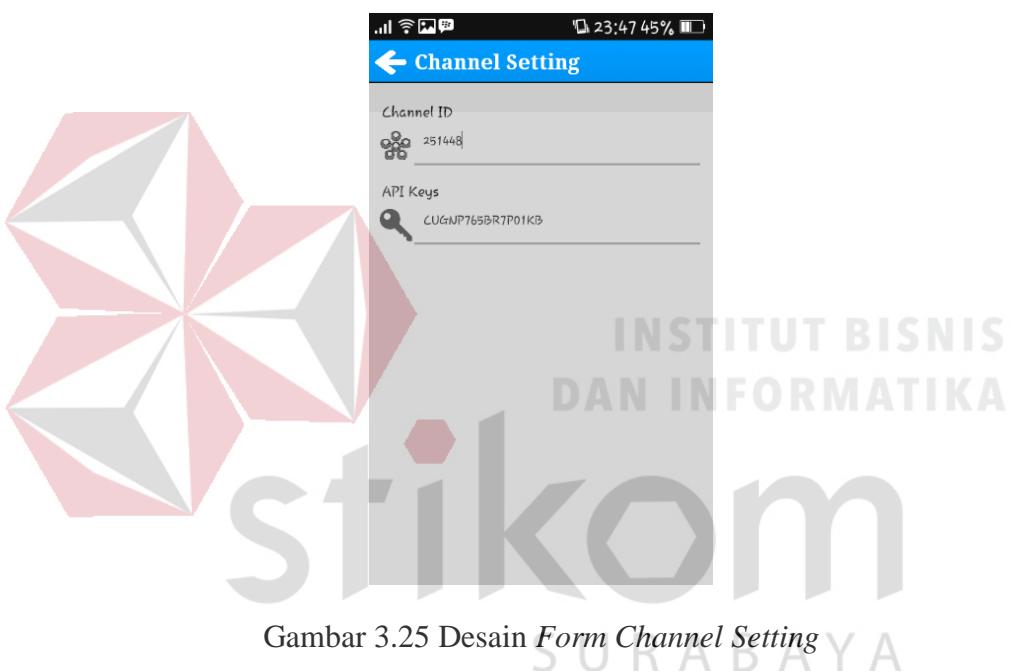
Pada *Form Setting* terdapat tiga menu, yaitu *Channel Settings*, *Informasi* dan *Takaran Setting*. *Channel Settings* berfungsi untuk melakukan pengaturan terhadap *API Key* dan *Channel ID*. *Informasi* berfungsi untuk mengetahui informasi tentang aplikasi. *Takaran Settings* berfungsi untuk melakukan pengaturan terhadap *Takaran Pemberian Pakan* Pada Alat Pemberi Pakan Ikan Otomatis. Desain dari *Form Setting* dapat dilihat pada gambar 3.24.



Gambar 3.24 Desain *Form Setting*

3.4.1.8 Form Channel Setting

Pada *Form Channel Setting* terdapat dua *textbox* yang berfungsi sebagai media *input API Key* dan *Channel ID*. Untuk menyimpan perubahan yang telah dilakukan *user* hanya perlu melakukan penekanan tombol *Back* pada *smartphone Android* maupun tombol *Back* yang berada pada pojok kiri atas *Form*. Desain dari *Form Channel Setting* dapat dilihat pada gambar 3.25.



3.4.1.9 Form Takaran Setting

Pada *Form Takaran Setting* terdapat dua *textbox* yang berfungsi sebagai media informasi Takaran Lama dan sebagai media *input* Takaran Baru. Untuk menyimpan perubahan Takaran pemberian pakan yang telah dilakukan, *user* hanya perlu melakukan penekanan tombol *Back* pada *smartphone Android* maupun tombol *Back* yang berada pada pojok kiri atas *Form*. Desain dari *Form Channel Setting* dapat dilihat pada gambar 3.26.



Gambar 3.26 Desain Form Takaran Setting

3.4.1.10 Taskbar

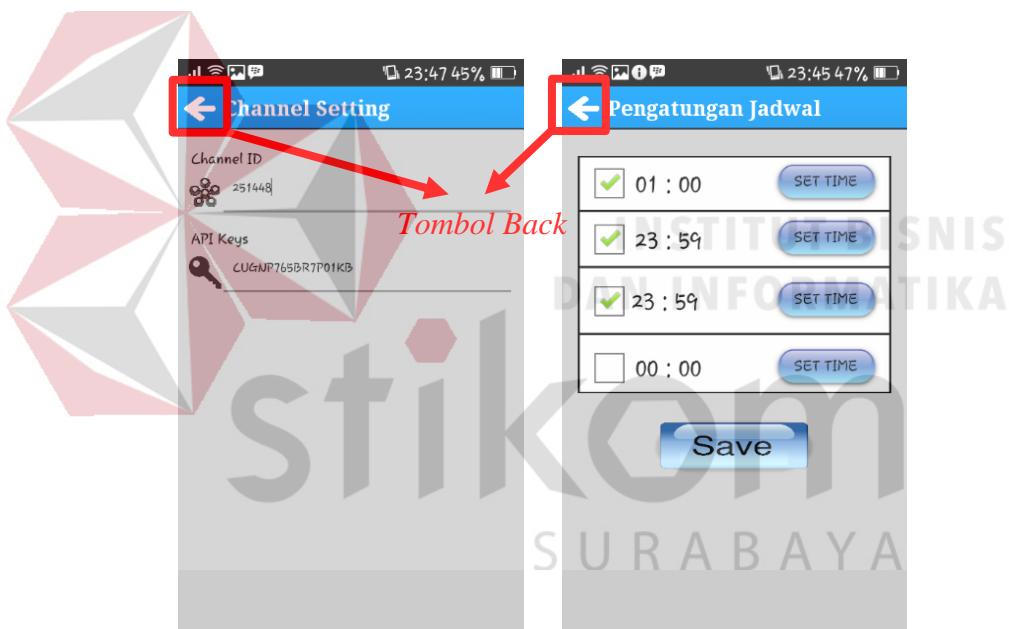
Taskbar berfungsi untuk membantu *user* dalam navigasi menu atau untuk membuka menu *monitoring* dan menu *scheduling*. Desain Taskbar dapat dilihat pada gambar 3.27.



Gambar 3.27 Desain Taskbar

3.4.1.11 Tombol Back

Tombol *Back* berfungsi untuk kembali menuju *Form* sebelumnya, tombol ini juga dapat berfungsi sebagai pengganti tombol *SAVE* pada *Form* dan kondisi tertentu. Ketika *user* berada pada *Form Set Scheduling*, *Form Channel Setting* atau *Form* Takaran *Setting* dan melakukan perubahan data, maka tombol *Back* dapat berfungsi sebagai pengganti tombol *SAVE*. Jika tidak, maka tombol *Back* akan berfungsi seperti pada umumnya. Desain dapat dilihat pada gambar 3.28.

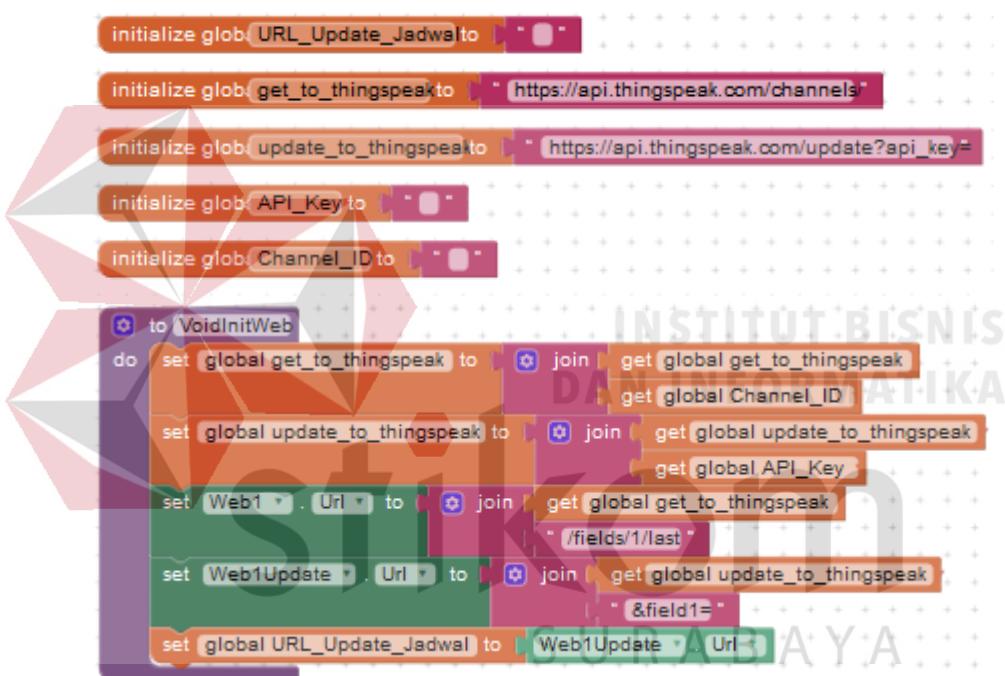


Gambar 3.28 Desain Tombol *Back*

3.4.2 Perancangan *Blocks* Ambil dan Update Data *Web Server*

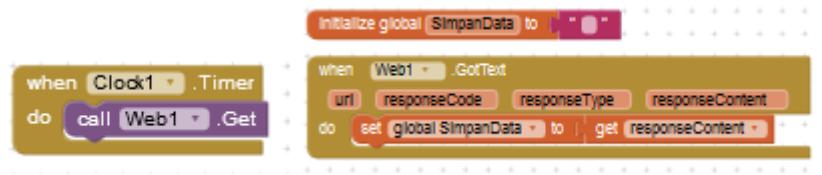
Setelah perancangan desain sudah selesai, selanjutnya akan dilakukan perancangan *Blocks* yaitu istilah pembuatan program yang terdapat pada App Inventor. Bentuk *code program* pada App Inventor berbentuk *puzzle* yaitu hanya dengan cara *drag and drop* untuk membuat program didalamnya. Perancangan ini

digunakan untuk mengakses dan mengambil data dari *web server*. Untuk dapat mengakses dan mengambil data dari *web server* maka harus menggunakan salah satu *component* pada *App Inventor* yaitu **web**. Setelah itu panggil **web.Url** untuk mengatur *Url component* **web** dengan *Url* dari *web server*, yaitu *Url* tertentu dari *api.thingspeak.com*. Untuk melakukan inisialisasi *component* **web** lihat pada gambar 3.29.



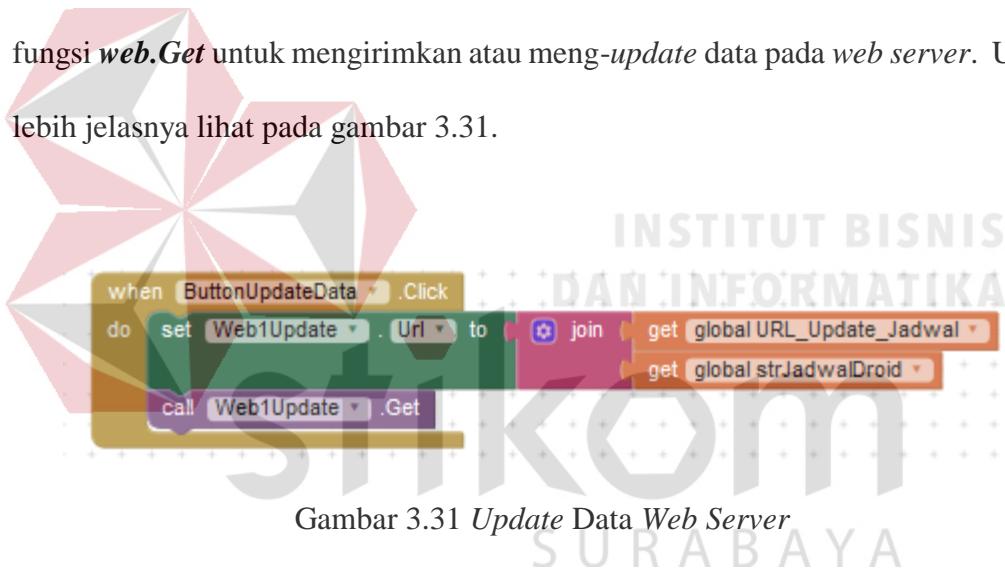
Gambar 3.29 Inisialisasi *Component web*

Setelah melakukan inisialisasi, langkah selanjutnya tinggal memanggil perintah **web.Get** untuk mengakses *web server*. Untuk membaca dan menyimpan data dari *web server* dapat menggunakan fungsi **web.GotText**. Sampai disini data dari *web server* sudah dapat dibaca, disimpan dan selanjutnya diproses untuk menjadi data Jadwal, Pemantauan dan *Request*. Untuk lebih jelasnya lihat pada gambar 3.30.



Gambar 3.30 Pengaksesan dan Penyimpanan Data *Web Server*

Untuk melakukan *update* data *web server*, dapat dilakukan dengan memanggil *web.Url* untuk menambahkan data yang ingin di *update* pada *web server*. Setelah data ditambahkan pada *Url*, langkah selanjutnya adalah memanggil fungsi *web.Get* untuk mengirimkan atau meng-*update* data pada *web server*. Untuk lebih jelasnya lihat pada gambar 3.31.

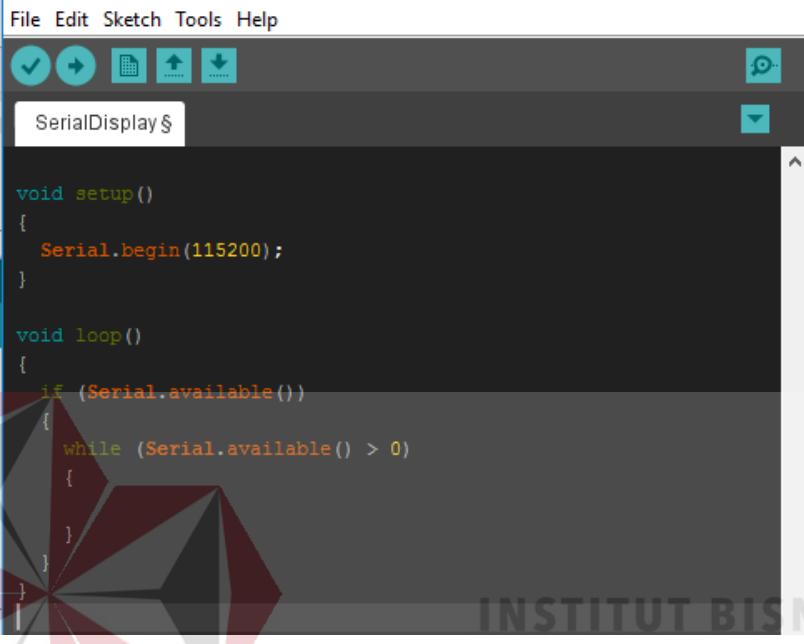


Gambar 3.31 *Update Data Web Server*

3.4.3 Perancangan Komunikasi *Serial* pada *Arduino* dan *Module WiFi NodeMCU ESP8266*

Perancangan ini digunakan sebagai komunikasi untuk saling berkirim data antara *Arduino* dengan *module* *ESP8266*. Transmisi data pada komunikasi serial dilakukan per-bit, dan untuk dapat salin berkomunikasi antara satu dengan yang satunya maka membutuhkan dua jalur yaitu *transmit* (Tx) dan *receive* (Rx). Dimana *transmit* (TX) berfungsi sebagai pengirim dan *receive* (RX) berfungsi sebagai

penerima. Pengaktifan Komunikasi *Serial* pada *Arduino* maupun *module WiFi ESP8266* pada *Arduino* IDE dapat dilihat dibawah ini.



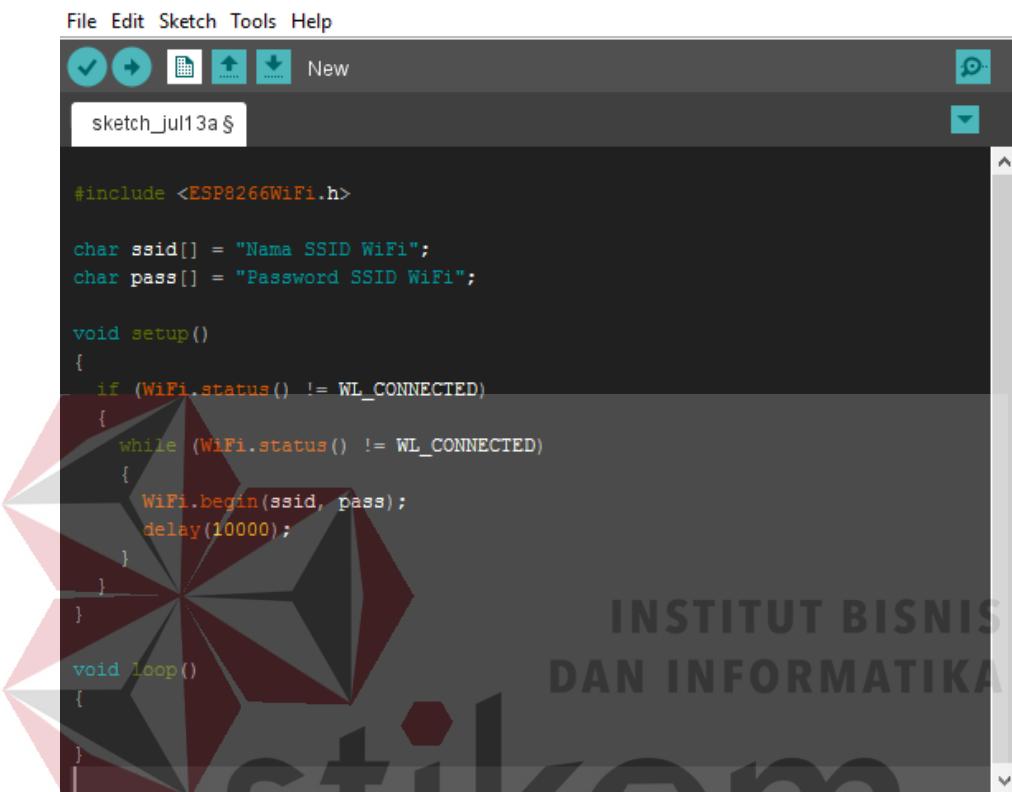
Gambar 3.32 Kode Komunikasi *Serial* *Arduino* dan *Module WiFi ESP8266*

Pada gambar 3.32 *Serial.begin()*; digunakan untuk mengatur kecepatan transmisi data, sedangkan *Serial.available()* digunakan sebagai *buffer* penerima data.

3.4.4 Perancangan Program Koneksi *Module WiFi NodeMCU ESP8266 Dengan WiFi*

Perancangan ini digunakan untuk menghubungkan atau mengkoneksikan *module WiFi ESP8266* dengan *WiFi*. Hal ini berfungsi supaya alat khususnya *Arduino* dan *module WiFi ESP8266* dapat mengakses *internet*. Cara

menghubungkan *module WiFi ESP8266* dengan *WiFi SSID* tertentu pada *Arduino* IDE dapat dilihat dibawah ini.



```

File Edit Sketch Tools Help
New
sketch_jul13a §
#include <ESP8266WiFi.h>

char ssid[] = "Nama SSID WiFi";
char pass[] = "Password SSID WiFi";

void setup()
{
    if (WiFi.status() != WL_CONNECTED)
    {
        while (WiFi.status() != WL_CONNECTED)
        {
            WiFi.begin(ssid, pass);
            delay(10000);
        }
    }
}

void loop()
{
}

```

Gambar 3.33 Kode Koneksi *Module WiFi ESP8266* Dengan *WiFi*

Pada gambar 3.33 pemanggilan *library* `#include <ESP8266WiFi.h>` digunakan supaya dapat menggunakan fungsi *WiFi* pada *module*. Untuk perintah *WiFi.status()* berfungsi untuk melihat *status WiFi* pada *module*. Untuk perintah *WiFi.begin(ssid, pass);* berfungsi untuk menghubungkan *module* dengan *SSID WiFi*, dimana *ssid* adalah nama dari *SSID WiFi* dan *pass* adalah *Pasword* dari *WiFi* yang akan dihubungkan dengan *module WiFi ESP8266*.

3.4.5 Perancangan Program *Module WiFi ESP8266* Untuk Ambil Data Dari *Web Server*

Perancangan ini digunakan oleh *module WiFi ESP8266* untuk mengambil data dari *web server*. *Web server* yang digunakan pada tugas akhir ini adalah <https://thingspeak.com/>. Program ini hanya dapat bekerja sebagaimana mestinya jika *module WiFi ESP8266* sudah terhubung dengan *WiFi* dan dapat digunakan untuk mengakses *internet*. Program dapat dilihat pada *Arduino IDE* dapat dilihat dibawah ini.



```

File Edit Sketch Tools Help
sketch_jul13a §
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>

WiFiClient client;

unsigned long ChannelID = 251448;
const char* APIKey = "CUGNP765BR7P01KB";
int Field = 1;

String strField;

void setup()
{
    ThingSpeak.begin(client);
}

void loop()
{
    strField = ThingSpeak.readStringField(ChannelID, Field, APIKey);
}

```

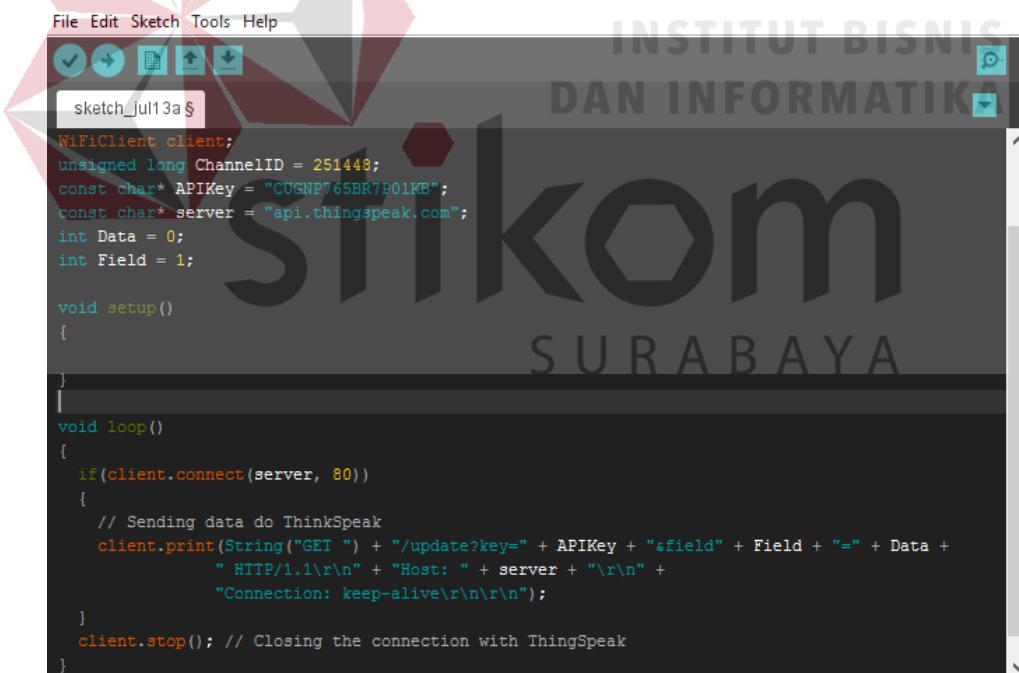
Gambar 3.34 Kode Ambil Data *Web Server* Pada *Module WiFi ESP8266*

Pada gambar 3.34 pemanggilan *library* `#include "ThingSpeak.h"` bertujuan supaya dapat menggunakan fungsi dari *Thingspeak*. `WiFiClient client;` berfungsi untuk deklarasi *client* pada *module*. Untuk perintah `strField = ThingSpeak.readStringField(ChannelID, Field, APIKey);` berfungsi untuk mengakses dan mengambil data dari *web server thingspeak*, dengan ketentuan *ChannelID* dan *APIKey* adalah *Channel ID* dan *API Key* yang sudah didapatkan

dari *web server thingspeak*. Sedangkan *Field* adalah nomor *Field* yang ingin dibaca dan diambil datanya. Pembacaan dan pengambilan data adalah data *last* atau data terakhir pada *web server* dengan nomor *Field* tertentu.

3.4.6 Perancangan Program *Module WiFi ESP8266* Untuk *Update Data Web Server*

Perancangan ini digunakan oleh *module WiFi ESP8266* untuk *update* atau mengirim data pada *web server*. *Web server* dan syarat untuk menggunakan fungsi ini sama dengan waktu program ambil data. Program dapat dilihat pada *Arduino* IDE dapat dilihat dibawah ini.



```

File Edit Sketch Tools Help
sketch_jul13a:sketch_jul13a.ino
WiFiClient client;
unsigned long ChannelID = 251448;
const char* APIKey = "CUGNP765BR7B01KB";
const char* server = "api.thingspeak.com";
int Data = 0;
int Field = 1;

void setup()
{
}

void loop()
{
    if(client.connect(server, 80))
    {
        // Sending data do ThinkSpeak
        client.print(String("GET ") + "/update?key=" + APIKey + "&field" + Field + "=" + Data +
                    " HTTP/1.1\r\n" + "Host: " + server + "\r\n" +
                    "Connection: keep-alive\r\n\r\n");
    }
    client.stop(); // Closing the connection with ThinkSpeak
}

```

Gambar 3.35 Kode *Update Data Web Server* Pada *Module WiFi ESP8266*

Pada gambar 3.35 pemanggilan *library* masih tetap sama pada saat ambil data. *Client.connect(server, 80)* berfungsi untuk melihat apakah *client* atau *module WiFi ESP8266* dapat terhubung pada *server* dengan 80 sebagai *port*. Untuk *server*

yang digunakan adalah "api.thingspeak.com". Untuk mengirim atau update data dapat menggunakan kode `client.print();`. Setelah membuka dan mengirim atau *update* data pada *server*, tutup kembali koneksi dengan kode `client.stop();`.

3.5 Pengaturan Web Server

Web server pada tugas akhir ini menggunakan <https://thingspeak.com/> yang berfungsi untuk menyimpan data dan sebagai perantara *monitoring* dan *scheduling*. Pertama yang dilakukan agar bisa mengakses *web server* tersebut yaitu mendaftarkan *account* baru.



Gambar 3.36 Sign Up thingspeak.com

Pada gambar 3.36 setelah masuk pada link <https://thingspeak.com/> klik pada *Sign Up* untuk melakukan pendaftaran *account* baru. Isikan data seperti *User ID*, *email*, *Time Zone*, dan *Password*. Setelah melengkapi data tersebut selanjutnya tekan *Create Account*. Setelah itu akan muncul halaman baru seperti pada gambar 3.37 dibawah ini.



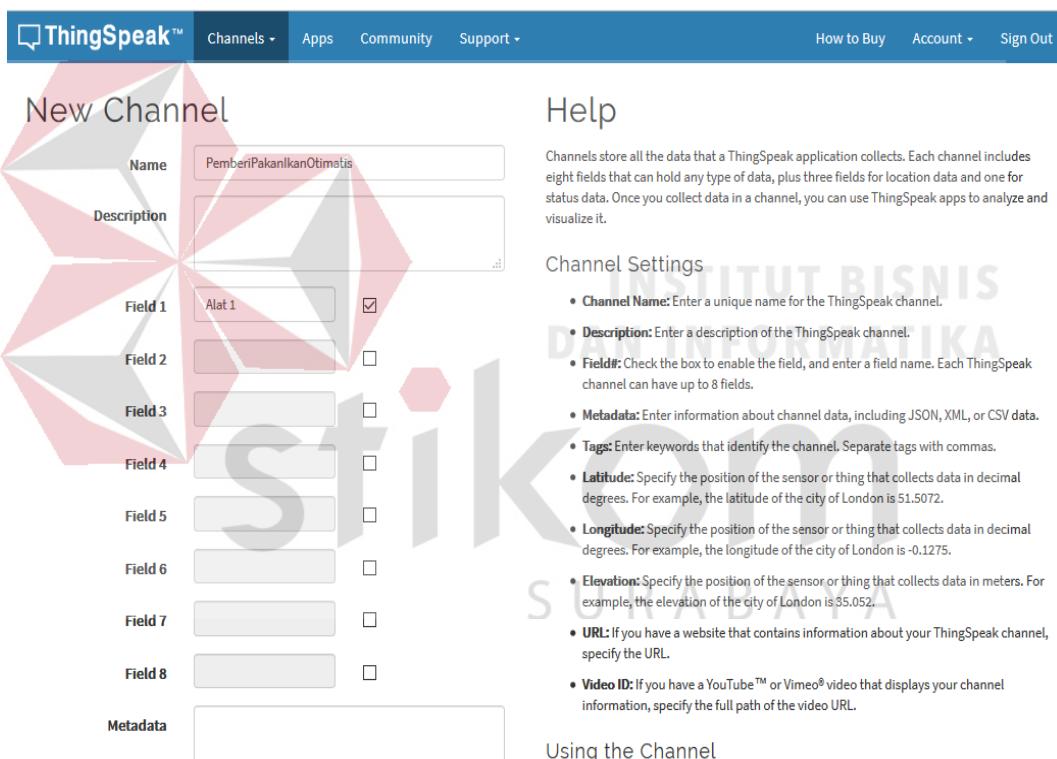
My Channels

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Gambar 3.37 Berhasil Membuat *Account* dan Masuk Pada *Web Server*

Klik New Channel seperti pada gambar 3.37 diatas. Setelah itu akan terbuka halaman baru seperti gambar 3.38 dibawah ini.



Gambar 3.38 Halaman *Channel* Baru

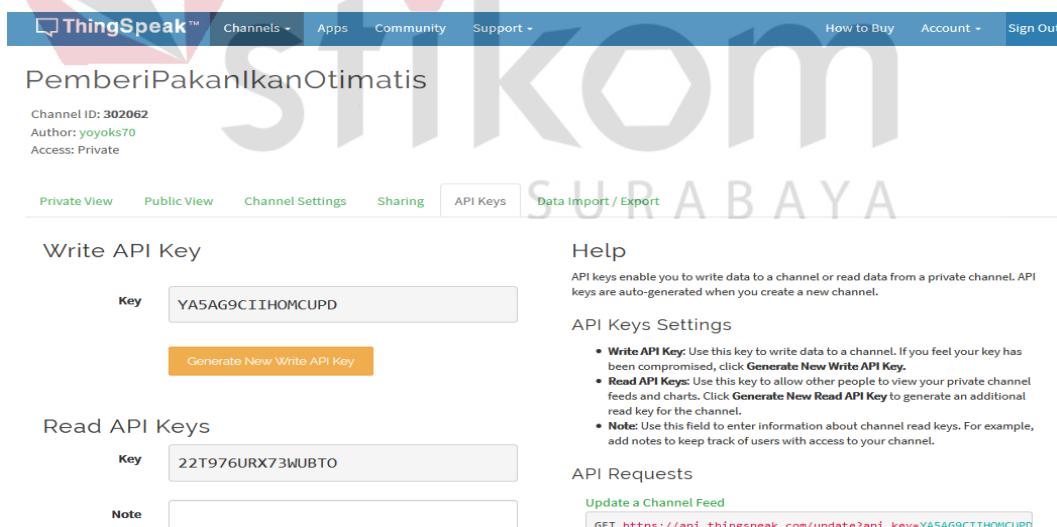
Pada gambar 3.38 diatas yaitu *input* nama dengan nama Pemberi Pakan Ikan Otomatis kemudian gunakan *Field 1* dengan nama Alat. Setelah itu klik *save channel* pada bagian bawah halaman untuk menyimpan *channel* baru tersebut dan

channel yang telah dibuat sudah bisa digunakan untuk menyimpan data.



Gambar 39. Tampilan Awal *Channel* Baru

Setelah membuat *channel* baru seperti pada gambar 3.39 diatas, maka akan mendapat *API key* dan *Channel ID* tertentu. *API Key* dan *Channel ID* tersebut sangat penting agar dapat melakukan pengiriman dan penerimaan data. Selanjutnya *web server* telah siap untuk digunakan. Berikut adalah tampilan saat melihat *API key* dan *Channel ID* pada gambar 3.40.



Gambar 3.40 Tampilan *API Keys* dan *Channel ID*

BAB IV

HASIL PENGUJIAN DAN PENGAMATAN

Dalam bab ini penulis akan menguraikan dan menjelaskan hasil analisis pengujian dari hasil penelitian tugas akhir ini yang telah dilakukan, pengujian dilakukan dalam beberapa bagian yang disusun dalam urutan dari yang sederhana menuju sistem yang lengkap. Pengujian dilakukan meliputi pengujian perangkat lunak (*software*) aplikasi *Smartphone Android* dan perangkat keras (*hardware*) meliputi *Arduino* dan *Module WiFi NodeMCU ESP8266*, diharapkan hasil yang didapat adalah suatu sistem yang dapat menjalankan rancangan alat berjalan dengan baik, optimal, dan bermanfaat.

4.1 Pengujian *Arduino*

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

4.1.1 Tujuan Pengujian *Arduino*

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

4.1.2 Alat Yang Dibutuhkan Pengujian *Arduino*

1. Komputer/Laptop
2. *Arduino Mega*
3. Kabel *serial usb type B*

4.1.3 Prosedur Pengujian Pengujian *Arduino*

1. Hubungkan *Arduino Mega* dengan kabel *serial usb type B*.
2. Selanjutnya nyalakan komputer/laptop dan jalankan program *Arduino IDE*.
3. Sambungkan kabel *serial* dengan komputer/laptop.
4. Buka *sketch* yang akan digunakan untuk di *upload* kedalam *Arduino Mega*.
5. *Setting board, serial port* dan *programmer* sesuai dengan yang digunakan.
6. Kemudian *upload sketch* dan tunggu hingga selesai.
7. Setelah *upload* selesai akan diketahui apakah program berhasil di *download* atau tidak oleh *Arduino Mega 2560*.

4.1.4 Hasil Pengujian *Arduino*

Dari percobaan di atas dapat diperoleh hasil dari proses *upload* pada jendela *comment* *Arduino IDE*. Apabila pada saat proses *upload* program tidak ada *comment* yang menunjukan kegagalan atau tidak ada *comment error* dalam sambungan antara kabel serial dan *Arduino Mega*, hal itu menandakan bahwa proses berjalan dengan baik. Proses *upload* dapat dilihat pada gambar 4.1. Apabila

proses *upload* program berjalan dengan baik maka di tandai dengan tampil *comment* seperti yang di tunjukan pada gambar 4.2.



Gambar 4.1 Tampilan Proses *Upload* Dari Arduino IDE Kepada Arduino



Gambar 4.2 Tampilan *Comment* Saat Berhasil *Upload* Kepada Arduino

4.2 Pengujian *Module WiFi NodeMCU ESP8266*

Pada sub bab ini akan menjelaskan tentang pengujian pada perangkat keras *Module WiFi NodeMCU ESP8266* yang telah dilakukan. Penjelasan meliputi tujuan

pengujian, alat yang dibutuhkan, prosedur pengujian dan hasil yang didapatkan dari pengujian.

4.2.1 Tujuan Pengujian *Module WiFi NodeMCU ESP8266*

Pengujian perangkat *Module WiFi NodeMCU ESP8266* bertujuan untuk mengetahui apakah *Module WiFi NodeMCU ESP8266* dapat berjalan dengan baik dan memastikan *Module WiFi NodeMCU ESP8266* yang dipakai tidak ada kerusakan sehingga dapat digunakan sesuai yang diharapkan.

4.2.2 Alat Yang Dibutuhkan Pengujian *Module WiFi NodeMCU ESP8266*

1. Komputer/Laptop
2. *Module WiFi NodeMCU ESP8266*
3. Kabel *serial micro usb*

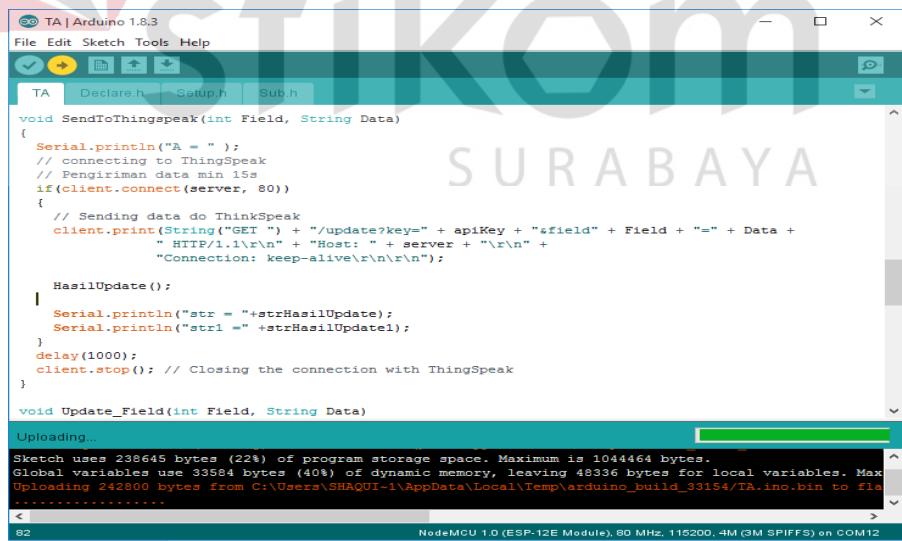
4.2.3 Prosedur Pengujian *Module WiFi NodeMCU ESP8266*

1. Hubungkan *Module WiFi ESP8266* dengan kabel *serial micro usb*.
2. Selanjutnya nyalakan komputer/laptop dan jalankan program *Arduino IDE*.
3. Sambungkan kabel *serial micro usb* dengan komputer/laptop.
4. Buka *sketch* yang akan digunakan untuk di *upload* kedalam *Module WiFi NodeMCU ESP8266*.
5. *Setting board, serial port* dan *programmer* sesuai dengan yang digunakan.

8. Kemudian *upload sketch* dan tunggu hingga selesai.
9. Setelah *upload* selesai akan diketahui apakah program berhasil di *download* atau tidak oleh *Module WiFi NodeMCU ESP8266*.

4.2.4 Hasil Pengujian *Module WiFi NodeMCU ESP8266*

Dari percobaan di atas dapat diperoleh hasil dari proses *upload* pada jendela *comment* *Arduino* IDE. Apabila pada saat proses *upload* program tidak ada *comment* yang menunjukkan kegagalan atau tidak ada *comment error*, hal itu menandakan bahwa proses berjalan dengan baik. Proses *upload* dapat dilihat pada gambar 4.3. Apabila proses *upload* program berjalan dengan baik maka di tandai dengan tampil *comment* seperti yang di tunjukan pada gambar 4.4.



```

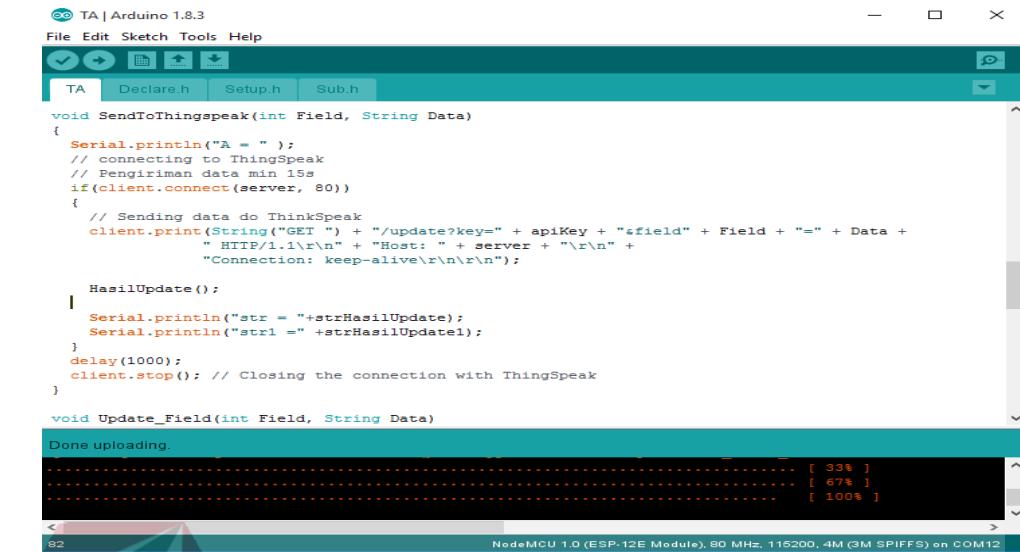
  void SendToThingSpeak(int Field, String Data)
  {
    Serial.println("A = " );
    // connecting to ThingSpeak
    // Pengiriman data min 1s
    if(client.connect(server, 80))
    {
      // Sending data do ThinkSpeak
      client.print("GET ") + "/update?key=" + apiKey + "&field" + Field + "=" + Data +
      " HTTP/1.1\r\n" + "Host: " + server + "\r\n\r\n" +
      "Connection: keep-alive\r\n\r\n";
    }

    HasilUpdate();
    Serial.println("str = "+strHasilUpdate);
    Serial.println("str1 = "+strHasilUpdate1);
  }
  delay(1000);
  client.stop(); // Closing the connection with ThingSpeak
}

void Update_Field(int Field, String Data)
{
  Uploading...
  Sketch uses 238645 bytes (22%) of program storage space. Maximum is 1044464 bytes.
  Global variables use 33584 bytes (40%) of dynamic memory, leaving 48336 bytes for local variables. Max
  Uploading 242800 bytes from C:\Users\SHAQUI-1\AppData\Local\Temp\arduino_build_33154\TA.ino to file
  .....
}

```

Gambar 4.3 Tampilan Proses *Upload* dari *Arduino IDE* Kepada *Module WiFi NodeMCU ESP8266*



```

TA | Arduino 1.8.3
File Edit Sketch Tools Help
TA Declare.h Setup.h Sub.h
void SendToThingspeak(int Field, String Data)
{
    Serial.println("A = ");
    // connecting to ThingSpeak
    // Pengiriman data min 15s
    if(client.connect(server, 80))
    {
        // Sending data do ThinkSpeak
        client.print("GET ") + "/update?key=" + apiKey + "&field" + Field + "=" + Data +
        " HTTP/1.1\r\n" + "Host: " + server + "\r\n" +
        "Connection: keep-alive\r\n\r\n";
    }
    HasilUpdate();
    Serial.println("str = "+strHasilUpdate);
    Serial.println("str1 = "+strHasilUpdate1);
}
delay(1000);
client.stop(); // Closing the connection with ThingSpeak
}

void Update_Field(int Field, String Data)
Done uploading.

```

NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (3M SPIFFS) on COM12

Gambar 4.4 Tampilan Comment Saat Berhasil Upload Kepada Module WiFi NodeMCU ESP8266

4.3 Pengujian Pemantauan Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi *Android*

Pada sub bab ini akan menjelaskan tentang pengujian pemantauan kerusakan yang terjadi pada Alat Pemberi Pakan Ikan Otomatis dari aplikasi *Android* yang telah dilakukan. Penjelasan meliputi tujuan pengujian, alat yang dibutuhkan, prosedur pengujian dan hasil yang didapatkan dari pengujian.

4.3.1 Tujuan Pengujian Pemantauan Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi *Android*

Pengujian ini dilakukan pada aplikasi *Android* yang telah dibuat, dengan nama *Smart Monitoring and Scheduling* atau dapat disingkat “SMaS.apk”.

Pengujian bertujuan untuk mengetahui apakah data pemantauan kerusakan yang terjadi pada Alat Pemberi Pakan Ikan Otomatis sesuai dan akurat atau tidak dengan data pemantauan kerusakan yang didapatkan oleh aplikasi *Android*.

4.3.2 Alat yang Dibutuhkan Pengujian Pemantauan Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi *Android*

1. Laptop/komputer
2. *Smartphone Android*
3. Aplikasi “**SMaS.apk**”
4. *Arduino Mega*
5. *Module WiFi NodeMCU ESP8266*
6. *Modem WiFi (Mobile WiFi)*
7. Kabel *Jumper*

4.3.3 Prosedur Pengujian Pemantauan Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi *Android*

1. Hidupkan *Modem WiFi (Mobile WiFi)*.
2. Hubungkan *port serial* pada *Arduino Mega 2560* dengan *port serial* pada *Module WiFi NodeMCU ESP8266* menggunakan kabel *jumper*.
3. Hidupkan Alat Pemberi Pakan Ikan Otomatis.
4. Pada Alat Pemberi Pakan Ikan Otomatis masukkan data Berat Pakan, Takaran, dan penjadwalan sesuai keinginan dengan menekan *keypad* pada

alat. Selanjutnya pilih START untuk memulai dan menjalankan alat, serta mengirim data pada *server*.

5. Siapkan *Smartphone Android* yang dapat mengakses *internet* dan sudah terinstall aplikasi “**SMaS.apk**”.
6. Buka aplikasi “**SMaS.apk**” dan buka menu *Monitoring*.

4.3.4 Hasil Pengujian Pemantauan Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi *Android*

Dari percobaan diatas apabila pemantauan kerusakan pada Alat Pemberi Pakan Ikan Otomatis berhasil dilakukan, hasil dari pemantauan kerusakan dapat dilihat pada menu *monitoring* aplikasi “**SMaS.apk**”. Aplikasi akan menampilkan informasi Motor DC 1 dan Motor DC 2 berjalan atau tidak serta akan tampil informasi terjadi kerusakan atau tidak pada alat.



Gambar 4.5 Tampilan Menu *Monitoring* Kerusakan

Pada gambar 4.5 diatas adalah salah satu hasil dari pengujian pemantauan kerusakan. Apabila terjadi kerusakan pada Alat Pemberi Pakan Ikan Otomatis, maka pada *smartphone android* akan mendapatkan *Notifikasi* atau pemberitahuan seperti gambar 4.6, bahwa sedang terjadi kerusakan pada penggerak dari Alat Pemberi Pakan Ikan Otomatis.



Gambar 4.6 Tampilan *Notifikasi* Terjadi Kerusakan

Tabel 4.1 Parameter Kerusakan Pada Alat Pemberi Pakan Ikan Otomatis

No.	Status Alat	Status Motor DC 1	Status Motor DC 2	Status Kerusakan
1.	ON	ON	ON	TIDAK TERJADI KERUSAKAN
2.	ON	ON	OFF	TERJADI KERUSAKAN
3.	ON	OFF	ON	TERJADI KERUSAKAN
4.	ON	OFF	OFF	TERJADI KERUSAKAN
5.	OFF	OFF	OFF	TIDAK TERJADI KERUSAKAN
6.	OFF	ON	OFF	TERJADI KERUSAKAN
7.	OFF	OFF	ON	TERJADI KERUSAKAN
8.	OFF	ON	ON	TERJADI KERUSAKAN

Tabel 4.1 diatas adalah parameter kerusakan pada Alat Pemberi Pakan Ikan Otomatis. Hasil pengujian pemantauan kerusakan yang terjadi pada penggerak dari

Alat Pemberi Pakan Ikan Otomatis pada aplikasi *Smartphone Android* adalah seperti pada tabel 4.2 yang dilakukan sebanyak 30 kali.

Tabel 4.2 Hasil Pengujian Pemantauan Kerusakan Pada Penggerak Alat Pemberi Pakan Ikan Otomatis Dengan Aplikasi *Android*

No.	Status Alat	Status Motor DC 1 Alat	Status Motor DC 2 Alat	Status Motor DC 1 Aplikasi <i>Android</i>	Status Motor DC 2 Aplikasi <i>Android</i>	Status Kerusakan Pada Aplikasi <i>Android</i>	Status Data Akurat
1.	ON	ON	OFF	ON	OFF	TERJADI KERUSAKAN	Akurat
2.	ON	ON	OFF	ON	OFF	TERJADI KERUSAKAN	Akurat
3.	ON	ON	OFF	ON	OFF	TERJADI KERUSAKAN	Akurat
4.	ON	OFF	ON	OFF	ON	TERJADI KERUSAKAN	Akurat
5.	ON	OFF	ON	OFF	ON	TERJADI KERUSAKAN	Akurat
6.	ON	OFF	ON	OFF	ON	TERJADI KERUSAKAN	Akurat
7.	ON	OFF	OFF	OFF	OFF	TERJADI KERUSAKAN	Akurat
8.	ON	OFF	OFF	OFF	OFF	TERJADI KERUSAKAN	Akurat
9.	ON	OFF	OFF	OFF	OFF	TERJADI KERUSAKAN	Akurat
10.	ON	ON	ON	ON	ON	TIDAK TERJADI KERUSAKAN	Akurat
11.	ON	ON	ON	OFF	OFF	TERJADI KERUSAKAN	Tidak Akurat
12.	ON	ON	ON	ON	ON	TIDAK TERJADI KERUSAKAN	Akurat
13.	ON	ON	ON	ON	ON	TIDAK TERJADI KERUSAKAN	Akurat
14.	ON	ON	ON	ON	ON	TIDAK TERJADI KERUSAKAN	Akurat
15.	ON	ON	ON	ON	ON	TIDAK TERJADI KERUSAKAN	Akurat
16.	OFF	ON	OFF	ON	OFF	TERJADI KERUSAKAN	Akurat
17.	OFF	ON	OFF	ON	OFF	TERJADI KERUSAKAN	Akurat
18.	OFF	ON	OFF	ON	OFF	TERJADI KERUSAKAN	Akurat

Tabel 4.2 Lanjutan

No.	Status Alat	Status Motor DC 1 Alat	Status Motor DC 2 Alat	Status Motor DC 1 Aplikasi Android	Status Motor DC 2 Aplikasi Android	Status Kerusakan Pada Aplikasi Android	Status Data Akurat
19.	OFF	OFF	ON	OFF	ON	TERJADI KERUSAKAN	Akurat
20.	OFF	OFF	ON	OFF	ON	TERJADI KERUSAKAN	Akurat
21.	OFF	OFF	ON	OFF	OFF	TIDAK TERJADI KERUSAKAN	Tidak Akurat
22.	OFF	ON	ON	OFF	OFF	TIDAK TERJADI KERUSAKAN	Tidak Akurat
23.	OFF	ON	ON	OFF	OFF	TIDAK TERJADI KERUSAKAN	Tidak Akurat
24.	OFF	ON	ON	ON	ON	TERJADI KERUSAKAN	Akurat
25.	OFF	ON	ON	ON	ON	TERJADI KERUSAKAN	Akurat
26.	OFF	ON	ON	ON	ON	TERJADI KERUSAKAN	Akurat
27.	OFF	OFF	OFF	OFF	OFF	TIDAK TERJADI KERUSAKAN	Akurat
28.	OFF	OFF	OFF	OFF	OFF	TIDAK TERJADI KERUSAKAN	Akurat
29.	OFF	OFF	OFF	OFF	OFF	TIDAK TERJADI KERUSAKAN	Akurat
30.	OFF	OFF	OFF	OFF	OFF	TIDAK TERJADI KERUSAKAN	Akurat

Dari 30 pengujian pemantauan kerusakan pada Alat Pemberi Pakan Ikan

Otomatis yang telah dilakukan. Hasil pengujian diatas didapatkan dari pengamatan yang dilakukan terhadap status alat ON/OFF atau waktunya pemberian pakan atau tidak dan kedua status Motor DC pada alat ON/OFF yang kemudian dibandingkan dengan data status Motor DC pada aplikasi *Android* ON/OFF, serta melihat tabel 4.1 sebagai acuan indikasi kerusakan. Sehingga didapatkan hasil seperti pada tabel 4.2 diatas.

Hasil pengujian pada tabel diatas dapat dilihat bahwa ada empat data pengujian pemantauan kerusakan yang tidak akurat antara Alat Pemberi Pakan Ikan Otomatis dengan aplikasi “SMaS.apk”. Data menjadi tidak akurat pada saat kondisi jaringan *internet* yang digunakan tidak stabil. Persentase tingkat keakuratan data dari 30 pengujian pemantauan kerusakan yang dikirim oleh Alat Pemberi Pakan Ikan Otomatis, dengan data kerusakan yang diterima oleh aplikasi “SMaS.apk” adalah 86.6%. Nilai persentase tersebut diperoleh dari perhitungan sebagai berikut.

$$\text{Persentase} = (\text{Data Akurat}/\text{Jumlah Pengujian}) \times 100$$

$$\text{Persentase} = (26/30) \times 100$$

$$\text{Persentase} = 86.6 \%$$

4.4 Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi *Android*

Pada sub bab ini akan menjelaskan tentang pengujian pemantauan ketersediaan pakan pada Alat Pemberi Pakan Ikan Otomatis dari aplikasi *Android* yang telah dilakukan. Pemantauan ketersediaan pakan meliputi berat pakan, persentase dan takaran pemberian pakan. Penjelasan meliputi tujuan pengujian, alat yang dibutuhkan, prosedur pengujian dan hasil yang didapatkan dari pengujian.

4.4.1 Tujuan Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi *Android*

Pengujian ini dilakukan pada aplikasi *Android* yang telah dibuat, dengan nama *Smart Monitoring and Scheduling* atau dapat disingkat “*SMaS.apk*”. Pengujian bertujuan untuk mengetahui apakah data pemantauan ketersediaan pakan pada Alat Pemberi Pakan Ikan Otomatis sesuai dan akurat atau tidak dengan data yang didapatkan oleh aplikasi *Android*.

4.4.2 Alat yang Dibutuhkan Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi *Android*

1. Laptop/komputer
2. *Smartphone Android*
3. Aplikasi “*SMaS.apk*”
4. *Arduino Mega*
5. *Module WiFi NodeMCU ESP8266*
6. *Modem WiFi (Mobile WiFi)*
7. Kabel *Jumper*

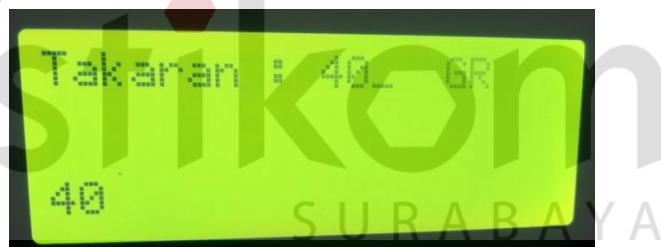
4.4.3 Prosedur Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi *Smartphone Android*

1. Hidupkan Modem WiFi (Mobile WiFi).

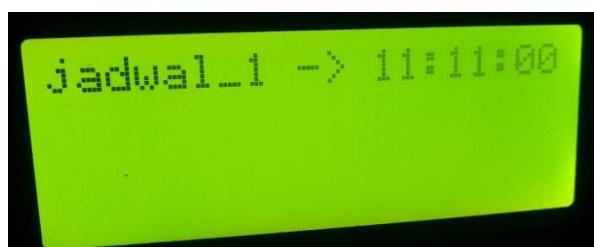
2. Hubungkan port serial pada Arduino Mega 2560 dengan port serial pada Module WiFi NodeMCU ESP8266 menggunakan kabel jumper.
3. Hidupkan Alat Pemberi Pakan Ikan Otomatis.
4. Pada Alat Pemberi Pakan Ikan Otomatis masukkan data Berat Pakan, Takaran, dan penjadwalan sesuai keinginan dengan menekan *keypad* pada alat. Selanjutnya pilih START untuk memulai dan menjalankan alat, serta mengirim data pada *server*.



Gambar 4.7 Tampilan *Input* Berat Pakan Pada *Display LCD*



Gambar 4.8 Tampilan *Input* Takaran Pada *Display LCD*



Gambar 4.9 Tampilan *Input* Jadwal Pada *Display LCD*



Gambar 4.10 Tampilan Input Start Pada *Display LCD*

5. Siapkan *Smartphone Android* yang dapat mengakses *internet* dan sudah terinstall aplikasi “**SMaS.apk**”.
6. Buka aplikasi “**SMaS.apk**” dan buka menu *Monitoring*.

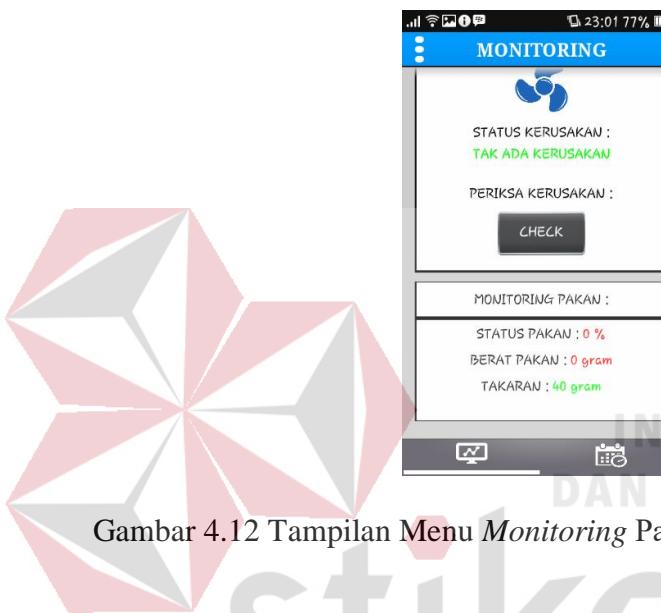


Gambar 4.11 Tampilan Menu *Monitoring Pakan*

4.4.4 Hasil Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dari Aplikasi *Android*

Dari percobaan diatas apabila pemantauan ketersediaan pakan pada Alat Pemberi Pakan Otomatis berhasil dilakukan, maka pada menu *monitoring* aplikasi

“SMaS.apk” akan menampilkan informasi ketersediaan pakan yang sesuai dan akurat dengan data ketersediaan yang terdapat pada Alat Pemberi Pakan Ikan Otomatis. Dimana informasi meliputi Berat Pakan, Persentase Pakan dan Takaran pemberian pakan seperti pada gambar 4.12.



Gambar 4.12 Tampilan Menu *Monitoring* Pada Aplikasi “SMaS.apk”

Tabel 4.3 Hasil Pengujian Pemantauan Ketersediaan Pakan Pada Alat Pemberi Pakan Ikan Otomatis Dengan Aplikasi *Android*

No.	Keterangan Alat	Tampilan <i>Display</i> Alat	Tampilan Aplikasi <i>Android</i>	Status Data Akurat
1.	Berat Pakan	1000 gram	1000 gram	Akurat
	Takaran	40 gram	40 gram	
2.	Berat Pakan	960 gram	960 gram	Akurat
	Takaran	40 gram	40 gram	
3.	Berat Pakan	920 gram	920 gram	Akurat
	Takaran	40 gram	40 gram	
4.	Berat Pakan	2000 gram	2000 gram	Akurat
	Takaran	80 gram	80 gram	
5.	Berat Pakan	1800 gram	1800 gram	Akurat
	Takaran	60 gram	60 gram	

Tabel 4.3 Lanjutan

No.	Keterangan Alat	Tampilan <i>Display</i> Alat	Tampilan Aplikasi <i>Android</i>	Status Data Akurat
6.	Berat Pakan	2000 gram	2000 gram	Akurat
	Takaran	100 gram	100 gram	
7.	Berat Pakan	5000 gram	5000 gram	Akurat
	Takaran	200 gram	200 gram	
8.	Berat Pakan	5000 gram	5000 gram	Akurat
	Takaran	120 gram	120 gram	
9.	Berat Pakan	500 gram	500 gram	Akurat
	Takaran	50 gram	50 gram	
10.	Berat Pakan	599 gram	500 gram	Tidak Akurat
	Takaran	56 gram	50 gram	
11.	Berat Pakan	5500 gram	5500 gram	Akurat
	Takaran	160 gram	160 gram	
12.	Berat Pakan	5340 gram	5340 gram	Akurat
	Takaran	200 gram	200 gram	
13..	Berat Pakan	5140 gram	5140 gram	Akurat
	Takaran	240 gram	240 gram	
14.	Berat Pakan	4900 gram	5140 gram	Tidak Akurat
	Takaran	80 gram	240 gram	
15.	Berat Pakan	9999 gram	9999 gram	Akurat
	Takaran	1000 gram	1000 gram	
16.	Berat Pakan	1234 gram	1234 gram	Akurat
	Takaran	123 gram	123 gram	
17.	Berat Pakan	7000 gram	7000 gram	Akurat
	Takaran	25 gram	25 gram	
18.	Berat Pakan	800 gram	800 gram	Akurat
	Takaran	9 gram	9 gram	
19.	Berat Pakan	6000 gram	6000 gram	Akurat
	Takaran	300 gram	300 gram	
	Persentase Pakan		100 %	
20.	Berat Pakan	6666 gram	6666 gram	Akurat
	Takaran	66 gram	66 gram	
21.	Berat Pakan	9988 gram	9988 gram	Akurat
	Takaran	89 gram	89 gram	
22.	Berat Pakan	3700 gram	3700 gram	Akurat
	Takaran	70 gram	70 gram	

Tabel 4.3 Lanjutan

No.	Keterangan	Tampilan <i>Display</i> Alat	Tampilan Aplikasi <i>Android</i>	Status Data Akurat
23.	Berat Pakan	4700 gram	4700 gram	Akurat
	Takaran	100 gram	100 gram	
24.	Berat Pakan	5656 gram	5656 gram	Akurat
	Takaran	1212 gram	1212 gram	
25.	Berat Pakan	6565 gram	6565 gram	Akurat
	Takaran	2121 gram	2121 gram	
26.	Berat Pakan	7878 gram	7878 gram	Akurat
	Takaran	87 gram	87 gram	
27.	Berat Pakan	1000 gram	1000 gram	Akurat
	Takaran	68 gram	68 gram	
28.	Berat Pakan	4321 gram	4321 gram	Akurat
	Takaran	32 gram	32 gram	
29.	Berat Pakan	5634 gram	5634 gram	Akurat
	Takaran	91 gram	91 gram	
30.	Berat Pakan	8998 gram	8998 gram	Akurat
	Takaran	121 gram	121 gram	

Dari 30 pengujian pemantauan ketersediaan pakan pada Alat Pemberi Pakan Ikan Otomatis yang telah dilakukan. Hasil pengujian diatas didapatkan dari pengamatan yang dilakukan terhadap *input* Berat Pakan dan Takaran pemberian pakan yang telah dilakukan oleh *user* pada alat, kemudian dibandingkan dengan informasi data Berat Pakan dan Takaran pemberian pakan pada aplikasi *Android*. Sehingga didapatkan hasil seperti pada tabel 4.3 diatas.

Hasil pengujian pada tabel diatas dapat dilihat bahwa ada dua data pengujian pemantauan ketersediaan pakan yang tidak akurat antara Alat Pemberi Pakan Ikan Otomatis dengan aplikasi “SMaS.apk”. Data menjadi tidak akurat pada saat kondisi jaringan *internet* yang digunakan tidak stabil. Persentase tingkat

keakuratan data dari 30 pengujian pemantauan ketersediaan pakan yang dikirim oleh Alat Pemberi Pakan Ikan Otomatis, dengan data ketersediaan pakan yang diterima oleh aplikasi “SMaS.apk” adalah 93.3%. Nilai persentase tersebut diperoleh dari perhitungan sebagai berikut.

$$\text{Persentase} = (\text{Data Akurat}/\text{Jumlah Pengujian}) \times 100$$

$$\text{Persentase} = (28/30) \times 100$$

$$\text{Persentase} = 93.3 \%$$

4.5 Pengujian Penjadwalan Dari Aplikasi *Android*

Pada sub bab ini akan menjelaskan tentang pengujian penjadwalan dari aplikasi *Android* kepada Alat Pemberi Pakan Ikan Otomatis yang telah dilakukan. Penjelasan meliputi tujuan pengujian, alat yang dibutuhkan, prosedur pengujian dan hasil yang didapatkan dari pengujian.

4.5.1 Tujuan Pengujian Penjadwalan Dari Aplikasi *Android*

Pengujian ini dilakukan pada aplikasi *Android* yang telah dibuat, dengan nama *Smart Monitoring and Scheduling* atau dapat disingkat “SMaS.apk”. Pengujian bertujuan untuk mengetahui apakah data penjadwalan yang dikirimkan oleh aplikasi *Android* sesuai dan akurat atau tidak dengan data yang diterima oleh Alat Pemberi Pakan Ikan Otomatis.

4.5.2 Alat yang Dibutuhkan Pengujian Penjadwalan Dari Aplikasi *Android*

1. Laptop/komputer
2. *Smartphone Android*
3. Aplikasi “**MSaS.apk**”
4. *Arduino Mega*
5. *Module WiFi NodeMCU ESP8266*
6. *Modem WiFi (Mobile WiFi)*
7. Kabel *Jumper*

4.5.3 Prosedur Pengujian Penjadwalan Dari Aplikasi *Android*

1. Hidupkan *Modem WiFi (Mobile WiFi)*.
2. Hubungkan *port serial* pada *Arduino Mega 2560* dengan *port serial* pada *Module WiFi NodeMCU ESP8266* menggunakan kabel *jumper*.
3. Hidupkan Alat Pemberi Pakan Ikan Otomatis.
4. Siapkan *Smartphone Android* yang dapat mengakses *internet* dan sudah terinstall aplikasi “**MSaS.apk**”.
5. Buka aplikasi “**MSaS.apk**” dan buka menu *Scheduling*.
6. Pada menu *Scheduling* tekan tombol **ATUR JADWAL**, selanjutnya atur jadwal pemberian pakan ikan sesuai keinginan.



Gambar 4.13 Tampilan Atur Jadwal

7. Setelah itu tekan **SAVE** atau **Back** untuk menyimpan dan mengirimkan data penjadwalan.

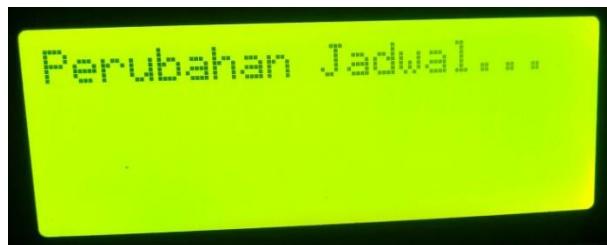


Gambar 4.14 Tampilan Simpan Dan *Update* Jadwal

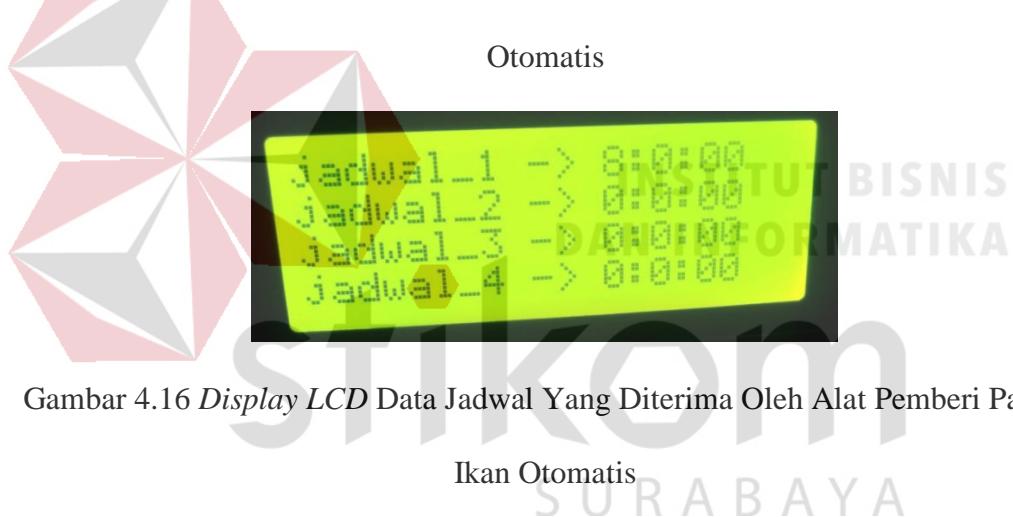
4.5.4 Hasil Pengujian Penjadwalan Dari Aplikasi *Android*

Dari pengujian diatas apabila perubahan jadwal dari aplikasi “SMaS.apk” berhasil dilakukan, maka pada *Display LCD* Alat Pemberi Pakan Otomatis akan

menampilkan informasi perubahan jadwal yang sesuai dan akurat dengan yang telah dilakukan pada aplikasi *Android*. Informasi perubahan jadwal pada alat dapat dilihat pada gambar 4.15 dan gambar 4.16.



Gambar 4.15 *Display LCD* Perubahan Jadwal Pada Alat Pemberi Pakan Ikan



Gambar 4.16 *Display LCD* Data Jadwal Yang Diterima Oleh Alat Pemberi Pakan

Ikan Otomatis

Tabel 4.4 Hasil Pengujian Penjadwalan Dari Aplikasi *Android*

No.	Keterangan	Tampilan Aplikasi <i>Android</i>	Tampilan Display Alat	Status Data Akurat
1.	Jadwal 1	08:00	08:00:00	Akurat
	Jadwal 2	00:00	00:00:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
2.	Jadwal 1	08:07	08:07:00	Akurat
	Jadwal 2	00:00	00:00:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	

Tabel 4.4 Lanjutan

No.	Keterangan	Tampilan Aplikasi Android	Tampilan Display Alat	Status Data Akurat
3.	Jadwal 1	08:18	08:18:00	Akurat
	Jadwal 2	09:03	09:03:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
4.	Jadwal 1	09:26	09:26:00	Tidak Akurat
	Jadwal 2	00:00	00:00:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:26:00	
5.	Jadwal 1	09:30	09:30:00	Akurat
	Jadwal 2	10:13	10:13:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
6.	Jadwal 1	07:30	07:30:00	Akurat
	Jadwal 2	13:13	13:13:00	
	Jadwal 3	16:41	16:41:00	
	Jadwal 4	00:00	00:00:00	
7.	Jadwal 1	07:30	07:30:00	Akurat
	Jadwal 2	13:13	13:13:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
8.	Jadwal 1	07:30	07:30:00	Akurat
	Jadwal 2	00:00	00:00:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
9.	Jadwal 1	05:05	05:05:00	Akurat
	Jadwal 2	11:05	11:05:00	
	Jadwal 3	14:07	14:07:00	
	Jadwal 4	18:00	18:00:00	
10.	Jadwal 1	06:05	06:05:00	Akurat
	Jadwal 2	12:05	12:05:00	
	Jadwal 3	15:07	15:07:00	
	Jadwal 4	19:00	19:00:00	
11.	Jadwal 1	07:05	07:05:00	Akurat
	Jadwal 2	13:05	13:05:00	
	Jadwal 3	16:07	16:07:00	
	Jadwal 4	20:00	20:00:00	
12.	Jadwal 1	07:14	07:14:00	Akurat
	Jadwal 2	13:07	13:07:00	

Tabel 4.4 Lanjutan

No.	Keterangan	Tampilan Aplikasi Android	Tampilan Display Alat	Status Data Akurat
	Jadwal 3	16:14	16:14:00	
	Jadwal 4	20:02	20:02:00	
13.	Jadwal 1	07:17	07:17:00	Akurat
	Jadwal 2	13:08	13:08:00	
	Jadwal 3	16:17	16:17:00	
	Jadwal 4	20:11	20:11:00	
14.	Jadwal 1	07:29	07:29:00	Akurat
	Jadwal 2	16:18	16:18:00	
	Jadwal 3	16:25	16:25:00	
	Jadwal 4	20:19	20:19:00	
15.	Jadwal 1	09:29	09:29:00	Akurat
	Jadwal 2	11:18	11:18:00	
	Jadwal 3	17:25	17:25:00	
	Jadwal 4	23:19	23:19:00	
16.	Jadwal 1	00:29	00:29:00	Akurat
	Jadwal 2	00:00	00:00:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
17.	Jadwal 1	00:34	00:34:00	Akurat
	Jadwal 2	00:00	00:00:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
18.	Jadwal 1	23:34	23:34:00	Akurat
	Jadwal 2	07:00	07:00:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
19.	Jadwal 1	23:34	23:34:00	Akurat
	Jadwal 2	09:00	09:00:00	
	Jadwal 3	08:00	08:00:00	
	Jadwal 4	00:00	00:00:00	
20.	Jadwal 1	23:34	23:34:00	Akurat
	Jadwal 2	09:00	09:00:00	
	Jadwal 3	08:02	08:02:00	
	Jadwal 4	00:00	00:00:00	
21.	Jadwal 1	23:34	23:34:00	Akurat
	Jadwal 2	05:00	05:00:00	
	Jadwal 3	08:02	08:02:00	
	Jadwal 4	00:00	00:00:00	

Tabel 4.4 Lanjutan

No.	Keterangan	Tampilan Aplikasi Android	Tampilan Display Alat	Status Data Akurat
22.	Jadwal 1	23:34	23:34:00	Akurat
	Jadwal 2	05:00	05:00:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
23.	Jadwal 1	00:02	00:02:00	Akurat
	Jadwal 2	00:00	00:00:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
24.	Jadwal 1	00:02	00:02:00	Akurat
	Jadwal 2	01:24	01:24:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
25.	Jadwal 1	00:02	00:02:00	Akurat
	Jadwal 2	02:29	02:29:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
26.	Jadwal 1	00:02	00:02:00	Akurat
	Jadwal 2	02:36	02:36:00	
	Jadwal 3	02:35	02:35:00	
	Jadwal 4	00:00	00:00:00	
27.	Jadwal 1	00:02	00:02:00	Tidak Akurat
	Jadwal 2	02:36	02:36:00	
	Jadwal 3	02:35	02:35:00	
	Jadwal 4	01:40	00:00:00	
28.	Jadwal 1	01:46	01:46:00	Akurat
	Jadwal 2	02:36	02:36:00	
	Jadwal 3	02:35	02:35:00	
	Jadwal 4	01:46	01:46:00	
29.	Jadwal 1	01:46	01:46:00	Akurat
	Jadwal 2	02:36	02:36:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	
30.	Jadwal 1	01:46	01:46:00	Akurat
	Jadwal 2	00:00	00:00:00	
	Jadwal 3	00:00	00:00:00	
	Jadwal 4	00:00	00:00:00	

Dari 30 pengujian penjadwalan dari aplikasi “SMaS.apk” yang telah dilakukan. Hasil pengujian diatas didapatkan dari pengamatan yang dilakukan terhadap *input* Jadwal pemberian pakan yang telah dilakukan oleh *user* pada aplikasi “SMaS.apk”, kemudian dibandingkan dengan informasi data Jadwal pemberian pakan pada Alat Pemberi Pakan Ikan Otomatis. Sehingga didapatkan hasil seperti pada tabel 4.4 diatas.

Hasil pengujian pada tabel diatas dapat dilihat bahwa ada dua data pengujian penjadwalan pemberian pakan yang tidak akurat antara Alat Pemberi Pakan Ikan Otomatis dengan aplikasi “SMaS.apk”. Data menjadi tidak akurat pada saat pengiriman data secara *serial* antara *Module WiFi NodeMCU ESP8266* dengan *Arduino Mega*, serta data menjadi tidak akurat pada saat kondisi jaringan *internet* yang digunakan tidak stabil. Persentase tingkat keakuratan data dari 30 pengujian penjadwalan pemberian pakan yang dikirim oleh aplikasi “SMaS.apk”, dengan data ketersediaan pakan yang diterima oleh Alat Pemberi Pakan Ikan Otomatis adalah 93.3%. Nilai persentase tersebut diperoleh dari perhitungan sebagai berikut.

$$\text{Persentase} = (\text{Data Akurat}/\text{Jumlah Pengujian}) \times 100$$

$$\text{Persentase} = (28/30) \times 100$$

$$\text{Persentase} = 93.3 \%$$

4.6 Pengujian Penjadwalan Dari Alat Pemberi Pakan Ikan Otomatis

Pada sub bab ini akan menjelaskan tentang pengujian penjadwalan dari Alat Pemberi Pakan Ikan Otomatis kepada aplikasi *Android* yang telah dilakukan.

Penjelasan meliputi tujuan pengujian, alat yang dibutuhkan, prosedur pengujian dan hasil yang didapatkan dari pengujian.

4.6.1 Tujuan Pengujian Penjadwalan Dari Alat Pemberi Pakan Ikan Otomatis

Pengujian ini dilakukan pada aplikasi *Android* yang telah dibuat, dengan nama *Smart Monitoring and Scheduling* atau dapat disingkat “*SMaS.apk*”.

Pengujian bertujuan untuk mengetahui apakah data penjadwalan yang dikirimkan oleh Alat Pemberi Pakan Ikan Otomatis sesuai dan akurat atau tidak dengan data yang diterima oleh aplikasi *Android*.

4.6.2 Alat yang Dibutuhkan Pengujian Penjadwalan Dari Alat Pemberi Pakan Ikan Otomatis

1. Laptop/komputer
2. *Smartphone Android*
3. Aplikasi “**MSaS.apk**”
4. *Arduino Mega*
5. *Module WiFi NodeMCU ESP8266*
6. *Modem WiFi (Mobile WiFi)*
7. Kabel *Jumper*

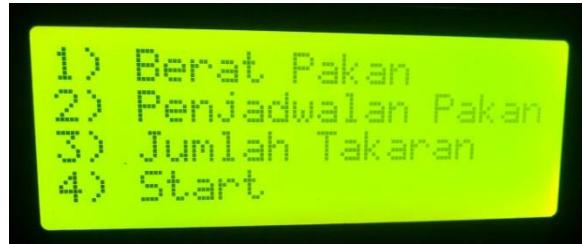
4.6.3 Prosedur Pengujian Penjadwalan Dari Alat Pemberi Pakan Ikan Otomatis

1. Hidupkan *Modem WiFi (Mobile WiFi)*.
2. Hubungkan *port serial* pada *Arduino Mega 2560* dengan *port serial* pada *Module WiFi NodeMCU ESP8266* menggunakan kabel *jumper*.
3. Hidupkan Alat Pemberi Pakan Ikan Otomatis.
4. Siapkan *Smartphone Android* yang dapat mengakses *internet* dan sudah terinstall aplikasi “**SMaS.apk**”.
5. Buka aplikasi “**SMaS.apk**” dan buka menu *Scheduling*.



Gambar 4.17 Tampilan Menu *Scheduling*

6. Pada Alat Pemberi Pakan Ikan Otomatis masukkan data Berat Pakan, Takaran, dan penjadwalan sesuai keinginan dengan menekan *keypad* pada alat. Selanjutnya pilih **START** untuk memulai dan menjalankan alat, serta mengirim data pada *server*



Gambar 4.18 Tampilan *Input* Menu Pada *Display LCD*



Gambar 4.19 Tampilan *Input* Jadwal Pada *Display LCD*



Gambar 4.20 Tampilan *Start* Pada *Display LCD*

4.6.4 Hasil Pengujian Penjadwalan Dari Alat Pemberi Pakan Ikan Otomatis

Dari pengujian diatas, apabila perubahan jadwal dari Alat Pemberi Pakan Otomatis berhasil dilakukan, maka pada menu *Scheduling* aplikasi “SMaS.apk” akan menampilkan informasi perubahan jadwal yang sesuai dan akurat dengan yang telah dilakukan pada Alat Pemberi Pakan Otomatis. Informasi perubahan jadwal pada aplikasi “SMaS.apk” dapat dilihat pada gambar 4.21.



Gambar 4.21 Tampilan Menu *Scheduling* Pada Aplikasi “SMaS.apk”

Tabel 4.5 Hasil Pengujian Penjadwal Dari Alat Pemberi Pakan Ikan Otomatis Kepada Aplikasi *Android*

No.	Keterangan	Tampilan Display Alat	Tampilan Aplikasi <i>Android</i>	Status Data Akurat
1.	Jadwal 1	11:11:00	11:11	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
2.	Jadwal 1	11:16:00	11:16	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
3.	Jadwal 1	22:22:00	22:22	Akurat
	Jadwal 2	33:33:00	33:33	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
4.	Jadwal 1	01:01:00	01:01	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
5.	Jadwal 1	09:30:00	09:30	Akurat
	Jadwal 2	09:09:00	09:09	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:30:00	00:00	

Tabel 4.5 Lanjutan

No.	Keterangan	Tampilan Display Alat	Tampilan Aplikasi Android	Status Data Akurat
6.	Jadwal 1	03:03:00	03:03	Akurat
	Jadwal 2	22:22:00	22:22	
	Jadwal 3	33:33:00	33:33	
	Jadwal 4	00:00:00	00:00	
7.	Jadwal 1	04:04:00	04:04	Akurat
	Jadwal 2	11:11:00	11:11	
	Jadwal 3	11:11:00	11:11	
	Jadwal 4	18:00:00	18:00	
8.	Jadwal 1	23:22:00	23:22	Akurat
	Jadwal 2	23:25:00	23:25	
	Jadwal 3	23:28:00	23:28	
	Jadwal 4	00:00:00	00:00	
9.	Jadwal 1	23:35:00	23:35	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
10.	Jadwal 1	23:42:00	23:41	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
11.	Jadwal 1	23:47:00	23:47	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
12.	Jadwal 1	23:50:00	23:47	Tidak Akurat
	Jadwal 2	23:56:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
13.	Jadwal 1	00:09:00	00:09	Akurat
	Jadwal 2	00:14:00	00:14	
	Jadwal 3	00:17:00	00:17	
	Jadwal 4	00:22:00	00:22	
14.	Jadwal 1	00:30:00	00:30	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	

Tabel 4.5 Lanjutan

No.	Keterangan	Tampilan Display Alat	Tampilan Aplikasi Android	Status Data Akurat
15.	Jadwal 1	00:37:00	00:30	Tidak Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
16.	Jadwal 1	00:44:00	00:44	Akurat
	Jadwal 2	00:48:00	00:48	
	Jadwal 3	22:00:00	22:00	
	Jadwal 4	00:00:00	00:00	
17.	Jadwal 1	00:53:00	00:53	Akurat
	Jadwal 2	00:58:00	00:58	
	Jadwal 3	01:01:00	01:01	
	Jadwal 4	00:00:00	00:00	
18.	Jadwal 1	01:04:00	01:04	Akurat
	Jadwal 2	01:08:00	01:08	
	Jadwal 3	01:13:00	01:13	
	Jadwal 4	01:18:00	01:18	
19.	Jadwal 1	01:22:00	01:22	Akurat
	Jadwal 2	01:27:00	01:27	
	Jadwal 3	01:32:00	01:32	
	Jadwal 4	01:37:00	01:37	
20.	Jadwal 1	01:48:00	01:48	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
21.	Jadwal 1	01:58:00	01:58	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
22.	Jadwal 1	02:02:00	02:02	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
23.	Jadwal 1	02:08:00	02:08	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	

Tabel 4.5 Lanjutan

No.	Keterangan	Tampilan Display Alat	Tampilan Aplikasi Android	Status Data Akurat
24.	Jadwal 1	02:11:00	02:11	Akurat
	Jadwal 2	22:22:00	22:22	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
25.	Jadwal 1	02:17:00	02:17	Akurat
	Jadwal 2	22:22:00	22:22	
	Jadwal 3	23:23:00	23:23	
	Jadwal 4	00:00:00	00:00	
26.	Jadwal 1	02:22:00	02:22	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
27.	Jadwal 1	02:26:00	02:26	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
28.	Jadwal 1	02:30:00	02:30	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
29.	Jadwal 1	02:38:00	02:38	Akurat
	Jadwal 2	00:00:00	00:00	
	Jadwal 3	00:00:00	00:00	
	Jadwal 4	00:00:00	00:00	
30.	Jadwal 1	02:42:00	02:30	Akurat
	Jadwal 2	02:45:00	02:45	
	Jadwal 3	02:48:00	02:48	
	Jadwal 4	02:51:00	02:51	

Dari 30 pengujian penjadwalan dari Alat Pemberi Pakan Ikan Otomatis yang telah dilakukan. Hasil pengujian diatas didapatkan dari pengamatan yang dilakukan terhadap *input* Jadwal pemberian pakan yang telah dilakukan oleh *user* pada Alat Pemberi Pakan Ikan Otomatis, kemudian dibandingkan dengan informasi

data Jadwal pemberian pakan pada aplikasi “SMaS.apk”. Sehingga didapatkan hasil seperti pada tabel 4.5 diatas.

Hasil pengujian pada tabel diatas dapat dilihat bahwa ada dua data pengujian penjadwalan pemberian pakan yang tidak akurat antara Alat Pemberi Pakan Ikan Otomatis dengan aplikasi “SMaS.apk”. Data menjadi tidak akurat pada saat kondisi jaringan *internet* yang digunakan tidak stabil. Persentase tingkat keakuratan data dari 30 pengujian penjadwalan pemberian pakan yang dikirim oleh Alat Pemberi Pakan Ikan Otomatis, dengan data ketersediaan pakan yang diterima oleh aplikasi “SMaS.apk” adalah 93.3%. Nilai persentase tersebut diperoleh dari perhitungan sebagai berikut.

$$\text{Persentase} = (\text{Data Akurat}/\text{Jumlah Pengujian}) \times 100$$

$$\text{Persentase} = (28/30) \times 100$$

$$\text{Persentase} = 93.3 \%$$

4.7 Pengujian Perubahan Takaran Pemberian Pakan Dari Aplikasi

Android Kepada Alat Pemberi Pakan Ikan Otomatis

Pada sub bab ini akan menjelaskan tentang pengujian perubahan takaran dari aplikasi *Android* kepada Alat Pemberi Pakan Ikan Otomatis yang telah dilakukan. Penjelasan meliputi tujuan pengujian, alat yang dibutuhkan, prosedur pengujian dan hasil yang didapatkan dari pengujian.

4.7.1 Tujuan Pengujian Perubahan Takaran Pemberian Pakan Dari Aplikasi *Android* Kepada Alat Pemberi Pakan Ikan Otomatis

Pengujian ini dilakukan pada aplikasi *Android* yang telah dibuat, dengan nama *Smart Monitoring and Scheduling* atau dapat disingkat “*SMaS.apk*”. Pengujian bertujuan untuk mengetahui apakah data penjadwalan yang dikirimkan oleh aplikasi *Android* sesuai dan akurat atau tidak dengan data yang diterima oleh Alat Pemberi Pakan Ikan Otomatis.

4.7.2 Alat yang Dibutuhkan Pengujian Perubahan Takaran Pemberian Pakan Dari Aplikasi *Android* Kepada Alat Pemberi Pakan Ikan Otomatis

1. Laptop/komputer
2. *Smartphone Android*
3. Aplikasi “*MSaS.apk*”
4. *Arduino Mega*
5. *Module WiFi NodeMCU ESP8266*
6. *Modem WiFi (Mobile WiFi)*
7. Kabel *Jumper*

4.7.3 Prosedur Pengujian Perubahan Takaran Pemberian Pakan Dari Aplikasi *Android* Kepada Alat Pemberi Pakan Ikan Otomatis

1. Hidupkan *Modem WiFi (Mobile WiFi)*.

2. Hubungkan *port serial* pada *Arduino Mega 2560* dengan *port serial* pada *Module WiFi NodeMCU ESP8266* menggunakan kabel *jumper*.
3. Hidupkan Alat Pemberi Pakan Ikan Otomatis.
4. Siapkan *Smartphone Android* yang dapat mengakses *internet* dan sudah terinstall aplikasi “**SMaS.apk**”.
5. Buka aplikasi “**SMaS.apk**”, buka menu *Setting* dan buka menu *Takaran Setting*, selanjutnya atur takaran pemberian pakan ikan sesuai keinginan.



Gambar 4.22 Tampilan Menu *Setting*



Gambar 4.23 Tampilan Menu *Takaran Setting*

- Setelah itu tekan **Back** untuk menyimpan dan mengirimkan data takaran.

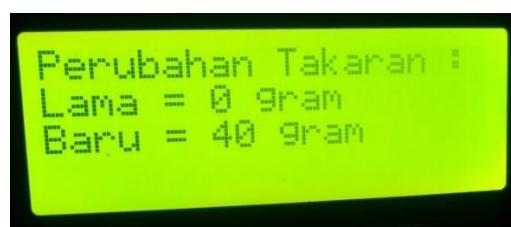


Gambar 4.24 Tampilan Simpan Dan *Update* DataTakaran

4.7.4 Hasil Pengujian Perubahan Takaran Pemberian Pakan Dari Aplikasi

Android Kepada Alat Pemberi Pakan Ikan Otomatis

Dari pengujian diatas apabila perubahan takaran dari aplikasi “SMaS.apk” berhasil dilakukan, maka pada *Display LCD* Alat Pemberi Pakan Otomatis akan menampilkan informasi perubahan takaran yang sesuai dan akurat dengan yang telah dilakukan pada aplikasi *Android*. Informasi perubahan jadwal pada alat dapat dilihat pada gambar 4.25.



Gambar 4.25 *Display LCD* Data Takaran Yang Diterima Oleh Alat Pemberi Pakan Ikan Otomatis

Tabel 4.6 Hasil Pengujian Perubahan Takaran Pada Alat Pemberi Pakan Ikan Otomatis Dengan Aplikasi *Android*

No.	Tampilan Aplikasi <i>Android</i>	Tampilan Display Alat	Status Data Akurat
1.	40 gram	40 gram	Akurat
2.	80 gram	80 gram	Akurat
3.	120 gram	120 gram	Akurat
4.	160 gram	160 gram	Akurat
5.	200 gram	200 gram	Akurat
6.	240 gram	240 gram	Akurat
7.	25 gram	25 gram	Akurat
8.	35 gram	25 gram	Tidak Akurat
9.	45 gram	25 gram	Tidak Akurat
10.	1000 gram	1000 gram	Akurat
11.	500 gram	500 gram	Akurat
12.	123 gram	123 gram	Akurat
13.	60 gram	60 gram	Akurat
14.	70 gram	70 gram	Akurat
15.	800 gram	800 gram	Akurat
16.	10 gram	10 gram	Akurat
17.	28 gram	28 gram	Akurat
18.	340 gram	340 gram	Akurat
19.	380 gram	380 gram	Akurat
20.	321 gram	321 gram	Akurat
21.	400 gram	400 gram	Akurat
22.	99 gram	99 gram	Akurat
23.	999 gram	999 gram	Akurat
24.	54 gram	54 gram	Akurat
25.	20 gram	20 gram	Akurat
26.	1040 gram	1040 gram	Akurat
27.	130 gram	130 gram	Akurat
28.7	110 gram	110 gram	Akurat
29.	210 gram	210 gram	Akurat
30.	310 gram	310 gram	Akurat

Dari 30 pengujian penjadwalan dari aplikasi “SMaS.apk” yang telah dilakukan. Hasil pengujian diatas didapatkan dari pengamatan yang dilakukan terhadap *input* takaran pemberian pakan yang telah dilakukan oleh *user* pada

aplikasi “SMaS.apk”, kemudian dibandingkan dengan informasi data takaran pemberian pakan pada Alat Pemberi Pakan Ikan Otomatis. Sehingga didapatkan hasil seperti pada tabel 4.6 diatas.

Hasil pengujian pada tabel diatas dapat dilihat bahwa ada dua data pengujian perubahan takaran pemberian pakan yang tidak akurat antara Alat Pemberi Pakan Ikan Otomatis dengan aplikasi “SMaS.apk”. Data menjadi tidak akurat pada saat kondisi jaringan *internet* yang digunakan tidak stabil. Persentase tingkat keakuratan data dari 30 pengujian penjadwalan pemberian pakan yang dikirim oleh aplikasi “SMaS.apk”, dengan data ketersediaan pakan yang diterima oleh Alat Pemberi Pakan Ikan Otomatis adalah 93.3%. Nilai persentase tersebut diperoleh dari perhitungan sebagai berikut.

$$\text{Persentase} = (\text{Data Akurat}/\text{Jumlah Pengujian}) \times 100$$

$$\text{Persentase} = (28/30) \times 100$$

$$\text{Persentase} = 93.3 \%$$

4.8 Analisa Hasil Pengujian

Pada sub bab ini akan menjelaskan tentang hasil dari beberapa pengujian pada sistem yang telah dilakukan. Pengujian meliputi pengujian pemantauan kerusakan pada Alat Pemberi Pakan Ikan Otomatis, pengujian pemantauan ketersediaan pakan pada Alat Pemberi Pakan Ikan Otomatis, pengujian penjadwalan dari aplikasi *Android*, pengujian penjadwalan dari pada Alat Pemberi

Pakan Ikan Otomatis dan pengujian perubahan takaran dari aplikasi *Android*. Hasil pengujian didapatkan dengan cara melakukan pengamatan keakuratan data antara Alat Pemberi Pakan Ikan Otomatis dan aplikasi *Android*.

Dari hasil pengujian yang telah dilakukan, didapatkan hasil bahwa tidak semua data akurat, antara data yang dikirimkan oleh aplikasi *Android* dengan data yang didapatkan Alat Pemberi Pakan Ikan Otomatis maupun sebaliknya. Dari pengamatan pada pengujian sistem yang telah dibuat, diketahui bahwa keakuratan data sangat terpengaruh oleh kondisi jaringan *internet* yang tidak stabil, dan pengiriman data secara *serial* antara *Module WiFi NodeMCU ESP8266* dengan *Arduino Mega 2560*.

Dari beberapa pengujian yang telah dilakukan pada sistem ini, didapatkan rata-rata persentase keakuratan data sebesar 91.96%. Nilai persentase tersebut didapatkan dari perhitungan rata-rata dari 5 pengujian utama, yaitu pengujian pemantauan kerusakan pada Alat Pemberi Pakan Ikan Otomatis dengan persentase keakuratan data 86.6%, pengujian pemantauan ketersediaan pakan pada Alat Pemberi Pakan Ikan Otomatis dengan persentase keakuratan data 93.3%, pengujian penjadwalan dari aplikasi *Android* dengan persentase keakuratan data 93.3%, pengujian penjadwalan dari pada Alat Pemberi Pakan Ikan Otomatis dengan persentase keakuratan data 93.3% dan pengujian perubahan takaran dari aplikasi *Android* dengan persentase keakuratan data 93.3%. Sehingga didapatkan perhitungan sebagai berikut.

$$Rata - rata = \frac{(Percentase Pengujian)}{Jumlah Pengujian}$$

$$Rata - rata = \frac{(86.6 + 93.3 + 93.3 + 93.3 + 93.3)}{5}$$

$$Rata - rata = 91.96 \%$$



BAB V

PENUTUP

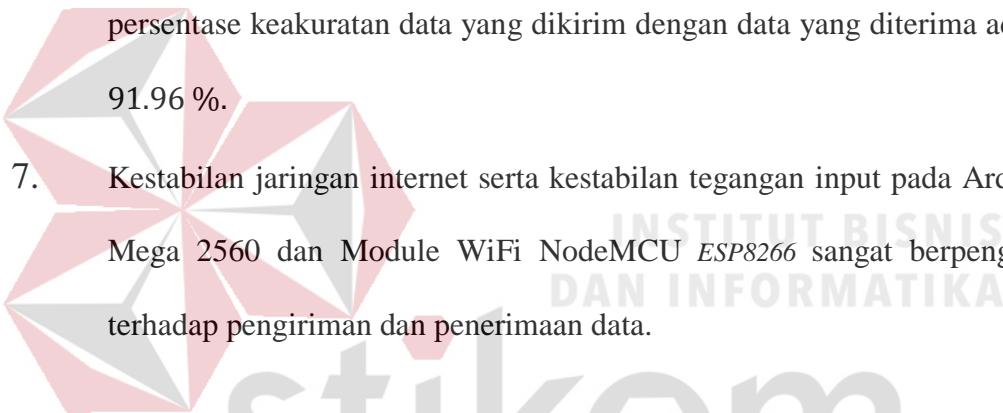
5.1. Kesimpulan

Berdasarkan hasil perancangan sistem dan seluruh pengujian yang telah dilakukan untuk semua kondisi yang mungkin terjadi pada aplikasi *Smartphone Android* dan Alat Pemberi Pakan Ikan Otomatis, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Aplikasi *Android* “**SMaS.apk**” yang telah dibuat dapat melakukan pemantauan kerusakan pada Alat Pemberi Pakan Ikan Otomatis secara jarak jauh melalui media internet. Dari 30 pengujian yang telah dilakukan, persentase keakuratan data adalah 86.6%.
2. Aplikasi *Android* “**SMaS.apk**” yang telah dibuat dapat melakukan pemantauan ketersediaan pakan pada Alat Pemberi Pakan Ikan Otomatis secara jarak jauh melalui media internet. Dari 30 pengujian yang telah dilakukan, persentase keakuratan data adalah 93.3%.
3. Aplikasi *Android* “**SMaS.apk**” yang telah dibuat dapat melakukan penjadwalan pada Alat Pemberi Pakan Ikan Otomatis secara jarak jauh melalui media internet. Dari 30 pengujian yang telah dilakukan, persentase keakuratan data adalah 93.3%.
4. Aplikasi *Android* “**SMaS.apk**” yang telah dibuat dapat menerima penjadwalan yang dilakukan pada Alat Pemberi Pakan Ikan Otomatis secara

jarak jauh melalui media internet. Dari 30 pengujian yang telah dilakukan, persentase keakuratan data adalah 93.3%.

5. Aplikasi *Android* “**SMaS.apk**” yang telah dibuat dapat melakukan perubahan takaran pemberian pakan pada Alat Pemberi Pakan Ikan Otomatis secara jarak jauh melalui media internet. Dari 30 pengujian yang telah dilakukan, persentase keakuratan data adalah 93.3%.
6. Analisis dari beberapa hasil pengujian yang telah dilakukan, rata-rata persentase keakuratan data yang dikirim dengan data yang diterima adalah 91.96 %.
7. Kestabilan jaringan internet serta kestabilan tegangan input pada Arduino Mega 2560 dan Module WiFi NodeMCU *ESP8266* sangat berpengaruh terhadap pengiriman dan penerimaan data.



5.2 Saran

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

1. Diperlukannya koneksi jaringan *internet* dan *power supply* yang lebih stabil.
2. Aplikasi “**SMaS.apk**” dapat dikembangkan lagi seperti, penambahan menu untuk memilih Alat Pemberi Pakan Ikan Otomatis yang masih dalam satu area atau lokasi yang ingin dipantau dan dikontrol, serta pengembangan supaya aplikasi *Android* dapat tetap berjalan dan memberian informasi kepada *user* walaupun aplikasi tidak sedang dijalankan atau dibuka.

DAFTAR PUSTAKA

- Arduino. (2017). Retrieved from Arduino: <https://www.arduino.cc/>
- Aristo, F. (2016). *Perancangan Dan Implementasi Sistem Kendali Lampu Dengan Arduino Uno Melalui ESP8266 Wifi Module Berbasis Android.*
- Ecadio. (2015, 5 23). *Apakah Arduino itu?* Retrieved from Ecadio: <http://ecadio.com/apakah-arduino-itu>
- Fathurohim, M. S. (2015). *Aplikasi Android Untuk Manajemen Pakan Kolam Ikan.*
- Novianty, S. J. (2016). *Prototype Monitoring Ketinggian Air Pada Bak Penampung Berbasis NodeMCU Pada Perguruan Tinggi Raharja.* Tangerang.
- Noviata, M. A., & Setyaningsih, E. (2015). *Sistem Informasi Monitoring Kereta Api Berbasis Web Server Menggunakan Layanan.*
- Nurahman, A. (2012). *Instalasi Dan Trouble Shooting Speedy.*
- Pratama, R. M. (2017). *Pengontrolan Otomatis Suhu Air Pada Kolam Pembelahan Ikan Berbasis Komputer Mini.*
- Saragih, A. R. (2016). *Rancang Bangun Perangkat Pemberi Pakan Ikan Otomatis Pada Kolam Pembelahan Ikan Berbasis Arduino.*
- Sensor Putaran. (2012). Retrieved from <http://elektronika-dasar.web.id/membuat-sensor-putaran-kecepatan/>
- Shneiderman, B. (2000). Universal Usability. *Communications of the ACM* Vol. 43, 85-91.
- Sulton, D. R. (2016). *Monitoring Dan Setting Greenhouse Berbasis Android.*