

**MODIFIKASI RANCANG BANGUN ALAT PEMBERI PAKAN UDANG
VANAME (*SEMI AUTOMATIC FEEDER*) DIBAWAH UMUR 30 HARI
BERBASIS *INTERNET OF THINGS (IOT)***

PROYEK AKHIR

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



Disusun Oleh :

HANIFAH THOHIROH

NIRM : 0031712

HARINDRA BAYU AJI SUSANTO

NIRM : 0031713

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG**

2020

LEMBAR PENGESAHAN

MODIFIKASI RANCANG BANGUN ALAT PEMBERI PAKAN UDANG VANAME (*SEMI AUTOMATIC FEEDER*) DIBAWAH UMUR 30 HARI BERBASIS *INTERNET OF THINGS (IOT)*

Oleh:

Hanifah Thohiroh	NIRM	0031712
Harindra Bayu Aji. S	NIRM	0031713

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

Menyetujui,

Pembimbing 1



Irwan, M. Sc, Ph. D

Pembimbing 2



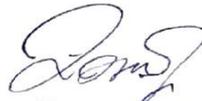
Riki Afriansyah, M.T

Penguji 1



Surojo, M. T

Penguji 2



Zanu Saputra, M. Tr. T

Penguji 3



M. Iqbal Nugraha, M. Eng

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini :

Nama Mahasiswa 1 : Hanifah Thohiroh NIRM : 0031712

Nama Mahasiswa 2 : Harindra Bayu Aji Susanto NIRM : 0031713

Dengan Judul : MODIFIKASI RANCANG BANGUN ALAT PEMBERI
PAKAN UDANG VANAME (*SEMI AUTOMATIC FEEDER*)
DIBAWAH UMUR 30 HARI BERBASIS *IOT (INTERNET*
OF THINGS)

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

Nama Mahasiswa

1. Hanifah Thohiroh

2. Harindra Bayu Aji Susanto

Sungailiat, Agustus 2020

Tanda Tangan

The image shows two handwritten signatures in blue ink. The first signature is for Hanifah Thohiroh and the second is for Harindra Bayu Aji Susanto. Each signature is positioned above a horizontal dotted line.

ABSTRAK

Pada saat ini telah banyak dibuat dan dikembangkan tambak udang vaname khususnya di Kepulauan Bangka Belitung. Salah satu masalah yang saat ini dihadapi bagi para pembudidaya udang yaitu proses pemberian pakan yang dilakukan secara manual dinilai kurang produktif karena memerlukan banyak tenaga kerja dan waktu yang cukup banyak. Selain itu pemberian pakan secara manual dapat menyebabkan sisa pakan menumpuk pada satu titik, hal ini dapat menyebabkan kualitas air pada tambak menurun. Jika hal ini terjadi maka dapat membuat pertumbuhan udang menjadi kurang baik. Dari persoalan tersebut, maka dibuatlah sebuah alat yang bernama automatic feeder berbasis Internet of Things (IoT). Tujuan dri penelitian ini adalah untuk mempermudah petani dalam proses pemberian pakan di tambak udang. Sistem ini menggunakan Arduino Mega 2560 dan NodeMCU sebagai komponen utama dalam hal mengendalikan motor DC 12 volt dan 24 volt. Sementara itu untuk memonitoring proses makan udang di tambak serta mengetahui indikator jumlah pakan udang menggunakan RTC dan sensor ultrasonik. automatic feeder ini dapat dikontrol secara otomatis dan manual melalui PC, serta dapat di monitoring lewat PC dan smartphone dengan menggunakan website. Alat ini juga dapat memonitoring jumlah pakan yang masih tersisa serta dapat mengatur waktu dalam proses pemberian pakan dan data jumlah pakan dapat disimpan dalam bentuk database. Serta alat ini juga memberikan indikator berupa lampu saat sedang melakukan proses maupun setelah selesai melakukan pemberian pakan sesuai jam yang telah diatur. Dengan penggunaan PWM pada driver motor dapat membuat pakan terbagi secara merata.

Kata Kunci : *Udang Vaname, automatic feeder, Internet of Things, Database*

ABSTRACT

At this time many vannamei shrimp ponds have been made and developed, especially in the Bangka Belitung Islands. One of the problems currently faced by shrimp cultivators is that the manual feeding process is considered less productive because it requires a lot of labor and a lot of time. In addition, manual feeding can cause leftover feed to accumulate at one point, this can cause the water quality in the pond to decrease. If this happens it can make shrimp growth less well. From these problems, a tool called automatic feeder based on the Internet of Things (IoT) was created. The aim of this research is to facilitate farmers in the process of feeding in shrimp ponds. This system uses Arduino Mega 2560 and NodeMCU as the main components in controlling 12 volt and 24 volt DC motors. Meanwhile, to monitor the feeding process of shrimp in the ponds and to determine the indicator for the amount of shrimp feed using RTC and ultrasonic sensors. This automatic feeder can be controlled automatically and manually via PC, and can be monitored via PC and smartphone using the website. This tool can also monitor the amount of feed that is still remaining and can adjust the time in the feeding process and data on the amount of feed can be stored in the form of a database. And this tool also provides an indicator in the form of a light while doing the process or after completing feeding according to the set hour. With the use of PWM on the motor driver can make the feed evenly distributed.

Keywords: *Vaname Shrimp, automatic feeder, Internet of Things, Database*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan laporan Proyek Akhir ini dengan baik dan tepat waktu. Laporan ini disusun sebagai sebagai salah satu persyaratan mahasiswa dalam menyelesaikan Proyek Akhir ini.

Dalam Proyek Akhir ini penulis membuat sebuah alat yang bernama *automatic feeder*. Dimana alat ini berfungsi untuk memberikan pakan udang di tambak. Sistem kontrol yang digunakan pada alat ini yaitu menggunakan Arduino, serta untuk penyimpanan data berupa *database* menggunakan NodeMCU. Dalam pembuatan Proyek Akhir ini penulis mengakui bahwa selesainya Proyek Akhir ini tidak lepas dari bantuan dan dukungan dari pihak-pihak tertentu. Oleh karena itu

penulis mengucapkan terima kasih kepada :

1. Bapak Irwan, M. Sc, Ph. D, selaku pembimbing 1 dalam Proyek Akhir ini.
2. Bapak Riki Afriansyah, M. T, selaku pembimbing 2 dalam Proyek Akhir ini.
3. Bapak I Made Andik Setiawan, M.Eng, Ph. D selaku Direktur Politeknik Manufaktur Bangka Belitung.
4. Bapak Muhammad Iqbal Nugraha, M. Eng, selaku Kepala Jurusan Teknik Elektronika dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
5. Bapak Ocsirendi, M. T, selaku Kepala Prodi Teknik Elektronika dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
6. Seluruh staf pengajar dan karyawan di Politeknik Manufaktur Negeri Bangka Belitung.
7. Orang tua beserta keluarga yang lainnya, yang selalu memberikan kasih sayang, doa serta dukungan kepada penulis.
8. Teman-teman yang telah ikut mendukung dan memberikan bantuan serta masukan dalam pembuatan Proyek Akhir ini.
9. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung yang tidak dapat disebutkan satu persatu.

Dalam laporan Proyek Akhir ini penulis menyadari masih terdapat kekurangan, oleh karena itu penulis sangat mengharapkan saran, kritik dan masukan yang bersifat membangun dalam rangka perbaikan laporan ini. Demikianlah laporan ini dibuat dan semoga laporan ini dapat bermanfaat dan menambah wawasan bagi para pembaca. Akhir kata penulis mengucapkan terima kasih.

Sungailiat, Agustus 2020

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	2
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN.....	xii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Proyek Akhir	3
BAB II DASAR TEORI	4
1.1. Udang Vaname	4
1.2. <i>Automatic Feeder</i>	5
1.2.1. <i>Internet of Things (IoT)</i>	5
1.2.2. Cara kerja <i>Automatic Feeder</i>	7
1.3. <i>Database</i>	8
BAB III METODE PELAKSANAAN	10
1.1. Tahap Pertama.....	10
1.2. Tahap Kedua.....	10

1.3.	Tahap Ketiga	11
1.4.	Tahap Keempat.....	12
1.5.	Tahap Kelima	13
1.6.	Tahap Keenam.....	13
BAB IV PEMBAHASAN.....		14
4.1.	Deskripsi Alat.....	14
4.2.	Sistem Mekanik <i>automatic feeder</i>	15
4.2.1.	Wadah Penampung Pakan.....	17
4.2.2.	Corong Penyalur Pakan.....	18
4.2.3.	Corong Penyalur Pakan.....	18
4.2.4.	Penampung dan Penyalur Pakan	19
4.2.5.	Motor DC 12V	19
4.2.6.	Pelontar Pakan.....	20
4.2.7.	Motor DC 24 volt	21
4.2.8.	<i>Power Supply</i>	21
4.2.9.	Pompa Air	22
4.2.10.	Penyangga Kaki Pompa air	22
4.3.	Rangkaian Kontrol dan Komunikasi	23
4.3.1	Program Penyalan motor DC 24V dan 12V.....	27
4.3.2	Program Pengiriman Data Serial Sensor Ultrasonik ke NodeMCU	27
4.4.	Uji Coba <i>Hardware</i>	27
4.4.1.	Pengujian Motor DC 12 volt & 24 volt.....	28
4.4.2.	Pengujian Jarak Lontaran Pakan	28
4.4.3.	Pengujian Keluaran Pakan.....	29
4.5.	Uji Coba <i>Software</i>	30
4.5.1.	Pengujian antara Arduino, <i>Real Time Clock</i> (RTC)	31
4.5.2.	Pengujian antara <i>Modul Wifi NodeMCU</i> , <i>Real Time Clock</i> (RTC), Sensor Ultrasonik dan <i>Database</i>	33
BAB V PENUTUP.....		36
5.1.	Kesimpulan.....	36
5.2.	Saran	36

DAFTAR PUSTAKA 38

DAFTAR TABEL

Tabel 4. 1 Pengujian Jarak Lontaran Pakan secara manual	29
Tabel 4. 2 Pengujian Jarak Lontaran Pakan menggunakan mesin	29
Tabel 4. 3 Pengujian Keluaran Pakan	30

DAFTAR GAMBAR

Gambar 2. 1 Udang Vaname	5
Gambar 2. 2 <i>automatic feeder</i>	8
Gambar 3. 1 Diagram blok <i>automatic feeder</i> secara umum	11
Gambar 4. 1 Diagram blok <i>automatic feeder</i>	15
Gambar 4. 2 Konstruksi <i>automatic feeder</i>	16
Gambar 4. 3 Wadah penampung pakan	17
Gambar 4. 4 Corong penyalur pakan	18
Gambar 4. 5 Penguat sisi corong	18
Gambar 4. 6 Penampung dan penyalur pakan.....	19
Gambar 4. 7 Peletakkan motor DC 12 volt	20
Gambar 4. 8 Pelontar pakan	20
Gambar 4. 9 Penempatan motor DC 24 volt Pengukuran menggunakan Tacho meter.....	21
Gambar 4. 10 Power Supply	21
Gambar 4. 11 Pompa Air	22
Gambar 4. 12 Dudukan Pompa Air.....	22
Gambar 4. 13 Konstruksi assembly <i>automatic feeder</i>	23
Gambar 4. 14 Rangkaian Kontrol dan Komunikasi	24
Gambar 4. 15 Flowchart Progres kerja Arduino	25
Gambar 4. 16 Flowchart progress kerja NodeMCU	26
Gambar 4. 17 Blok diagram uji coba penyalaan motor DC 12 volt & 24 volt.	28
Gambar 4. 18 hasil pengujian RTC sistem kontrol hidup.....	32
Gambar 4. 19 Tampilan Database di Smartphone dan PC.....	33
Gambar 4. 20 Tampilan menu sisa pakan	34
Gambar 4. 21 Progres bar.....	34
Gambar 4.22 Menu cek histori.....	35
Gambar 4.23 Menu cek histori.....	35

DAFTAR LAMPIRAN

LAMPIRAN 1 : Daftar Riwayat Hidup

LAMPIRAN 2 : Desain *automatic feeder*

LAMPIRAN 3 : Program Arduino Mega 2560

LAMPIRAN 4 : Program NodeMCU

BAB I

PENDAHULUAN

1.1. Latar Belakang

Budidaya udang saat ini berkembang dengan pesat, dan banyak sekali pengusaha kecil maupun besar yang terjun ke pembudidayaan udang tersebut. Dengan berkembangnya budidaya udang tersebut, ada beberapa faktor yang menjadi kendala bagi pembudidaya. Mengutip dari DJPB (Direktorat Jendral Perikanan Budidaya Kementerian Kelautan dan Perikanan) salah satu kendala yang masih menjadi masalah bagi para pembudidaya yaitu terjadinya ketidak seimbangan kondisi lingkungan dan menurunnya daya dukung lingkungan sebagai akibat pengelolaan budidaya yang tidak terkontrol menjadi faktor utama penyebab munculnya masalah dalam usaha budidaya udang saat ini. Oleh karena itu, perlu dilakukan peningkatan penerapan teknologi yang inovatif, aplikatif, dan efesensi, sebagai upaya untuk meningkatkan produktifitas tambak dan perbaikan kualitas lingkungan. Lingkungan yang dimaksud yaitu air yang terdapat pada tambak tersebut. Semakin bagus kualitas air semakin baik pula lingkungan yang dihasilkan. Tetapi kualitas air juga bergantung dari jumlah pakan yang ditebar, semakin banyak pakan yang ditebar maka penurunan kualitas air akan semakin cepat. Salah satu upaya yang dapat dilakukan agar kualitas air tetap terjaga yaitu dengan menebar pakan secara merata dengan menggunakan mesin *automatic feeder*. Dan penelitian yang telah dilakukan menurut DJPB dengan menggunakan *automatic feeder* secara nyata mampu mengurangi masa pemeliharaan udang *vanamei* secara intensif, yaitu dapat dipersingkat hingga 25% dibandingkan dengan cara manual. Sebagai gambaran, untuk mencapai ukuran udang 35 gram biasanya diperlukan waktu 120 hari, namun dengan menggunakan *automatic feeder* bisa dipersingkat sekitar 90 hari (DJPB, 2017).

Begitu pula dengan tambak udang yang berlokasi di desa Rebo Kab. Bangka. Tambak tersebut sudah menggunakan *automatic feeder* sebagai alat bantu pemberian pakan udang mereka. Setelah melakukan survey ke tambak

tersebut, kami melihat bagaimana bentuk serta cara kerja dari mesin itu sendiri. Yang menjadi permasalahan adalah kurangnya jarak lontar dari mesin *automatic feeder* tersebut. Proyek akhir tahun kemarin sudah pernah dibuat mesin yang sama tapi untuk jarak lontarnya masih belum bisa terpenuhi (M. Anzullah dan Rachmah Suci Saputri , 2019).

Sama halnya dengan tambak udang yang ada di kabupaten Bangka, kecamatan Rebo, tambak ini juga telah menggunakan *automatic feeder* sebagai alat bantu dalam pemberian makan udang. Saat melakukan survei ke tambak, dapat dilihat bagaimana bentuk serta cara kerja dari mesin *automatic feeder* yang telah ada sebelumnya. Setelah diperhatikan, kelemahan *automatic feeder* ini adalah pakan langsung jatuh dari wadah kedalam pelontar, sehingga membuat pakan tersebut menumpuk sebelum sempat terlontarkan ke luar. Kemudian bagian alat yang lainnya terdiri dari wadah penampung pakan berbentuk pelet, karena mesin ini digunakan untuk udang yang berusia diatas 30 hari. Wadah yang berbentuk tong ini memiliki kapasitas tampung ± 50 kg. Pemutaran wadah dilakukan dengan sebuah motor AC 3 *phase* dengan putaran sebanyak ± 1500 RPM. Dengan putaran tersebut, pakan dapat menyebar ke titik 10 m dari posisi mesin dan ketinggian *automatic feeder* tidak dapat diatur. Dari survei yang dilakukan ini, inovasi yang akan dilakukan untuk mengembangkan *automatic feeder* ini dengan penambahan sistem kontrol berbasis *Internet of Things* (IoT), serta untuk pemberian pakan udang yang berusia dibawah 30 hari, dan penambahan sistem *database* untuk pakan udang selama proses pembesaran 30 hari kedepan. Karena untuk udang di usia tersebut pakan yang digunakan bukan menggunakan pelet yang besar, melainkan pelet yang berukuran lebih kecil.

Dengan adanya alat ini dapat mempersingkat waktu para pekerja tambak dan menebar pakan udang secara merata agar pertumbuhan udang dapat berjalan dengan semestinya sehingga meminimalisir kerugian bagi para pembudidaya udang.

1.2. Rumusan Masalah

Berikut adalah permasalahan yang terjadi di tambak udang yang menyebabkan masalah bagi para pembudidaya udang yaitu :

1. Bagaimana proses penebaran pakan udang yang berupa pelet ukuran kecil dapat tertebar ke kolam.
2. Dengan penggunaan *automatic feeder* bagaimana cara agar dapat mengefesienkan waktu para pekerja.
3. Bagaimana agar proses penyimpanan jumlah pakan yang telah di digunakan selama proses pembesaran dapat tersimpan ke dalam *database*.

1.3. Batasan Masalah

Berikut adalah permasalahan yang terjadi di tambak udang yang menyebabkan masalah bagi para pembudidaya udang yaitu :

1. Jarak lontar pakan terbatas maksimal 1,5 meter.
2. Pemberian pakan dengan *automatic feeder* ini untuk usia udang di bawah 30 hari.
3. Kapasitas wadah hanya dapat menampung pakan sebanyak 50 kg (1 kali makan sebanyak 9 kg).
4. Untuk akses kedalam *database* hanya menggunakan satu jaringan koneksi yang sama.

1.4. Tujuan Proyek Akhir

Berikut adalah tujuan dibuat *automatic feeder* pada proyek akhir ini yaitu :

1. Membantu para pembudidaya udang agar pemberian pakan dapat dilakukan tepat waktu.
2. Mempermudah para pekerja untuk menyimpan data jumlah pakan yang telah di digunakan selama proses pembesaran ke dalam *database*.

BAB II

DASAR TEORI

1.1. Udang Vaname

Udang merupakan salah satu komoditas perikanan Indonesia yang berpotensi besar untuk dikembangkan. Saat ini komoditas udang bernilai ekonomi mencapai USD250 miliar atau sekitar Rp3,6 triliun setiap tahun. Indonesia sendiri menempati urutan ketiga terbesar sebagai negara pengekspor udang di pasar dunia setelah Thailand dan India. Jenis udang yang diekspor oleh Indonesia di antaranya adalah udang windu, udang vaname, dan jenis udang lainnya.

Khususnya mengenai udang vaname (*Litopenaeus vannamei*), rata-rata jenis udang ini memiliki kontribusi volume ekspor mencapai 85%. Udang vaname memiliki karakteristik spesifik, seperti mampu hidup pada kisaran salinitas yang luas, mampu beradaptasi dengan lingkungan bersuhu rendah, memiliki tingkat keberlangsungan hidup yang tinggi, dan memiliki ketahanan yang cukup baik terhadap penyakit sehingga cocok untuk dibudidayakan di tambak.

Udang vaname merupakan udang yang berasal dari daerah subtropis pantai barat Amerika, mulai dari Teluk California di Mexico bagian utara sampai pantai barat Guatemala, El Salvador, Nicaragua, Kosta Rika di Amerika Tengah hingga ke Peru di Amerika Selatan. Di Indonesia udang vaname mulai banyak di budidayakan dan dijadikan sebagai pengganti udang windu (*Penaeus monodon*), dimana produksi udang windu menurun sejak 1996 akibat penurunan kualitas lingkungan dan sering mengalami kematian massal akibat penyakit dan virus.

Dalam Surat Keputusan (SK) Menteri Kelautan dan Perikanan RI. No. 41/2001 tentang Pelepasan Varietas Udang Vaname Sebagai Varietas Unggul, bahwa udang vaname merupakan udang varietas unggulan yang mampu meningkatkan produksi, pendapatan dan kesejahteraan petani ikan.

Udang vaname mempunyai keunggulan diantaranya dapat mencapai ukuran besar, dapat tumbuh secepat udang windu (3 g/minggu), dapat dibudidayakan pada kisaran salinitas yang lebar (0,5- 45 ppt /part per thousand),

kebutuhan protein yang lebih rendah (20-35%) dibanding udang windu dan dapat ditebar dengan kepadatan tinggi hingga lebih dari 150 ekor/m². Di bawah ini adalah gambar udang vaname yang banyak dibudidayakan, terdapat pada gambar 2.1 berikut ini :



Gambar 2. 1 Udang Vaname

1.2. Automatic Feeder

Automatic feeder adalah pemberi pakan otomatis udang yang bekerja secara otomatis dalam hal pengaturan jadwal, frekuensi pemberian pakan, dosis pakan serta dapat beroperasi selama 24 jam secara terus-menerus. Semakin banyak pakan yang ditebar maka penurunan kualitas air akan semakin cepat, sehingga permasalahan dalam budidaya semakin cepat. Salah satu upaya yang bisa dilakukan untuk menjaga kualitas air agar tidak cepat menurun adalah penggunaan mesin *automatic feeder*. Dengan menggunakan *automatic feeder*, maka kesalahan cara pemberian pakan secara manual yang ditebar secara keliling bisa dihindari, dengan kata lain alat ini sangat efektif dalam memperbaiki manajemen pemberian pakan.

1.2.1. Internet of Things (IoT)

Internet of Things (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. IoT telah berkembang

dari konvergensi teknologi nirkabel, *micro-electromechanical systems* (MEMS), dan Internet.

“A *Things*” pada *Internet of Things* dapat didefinisikan sebagai subjek misalkan orang dengan monitor implant jantung, hewan peternakan dengan transponder biochip, sebuah mobil yang telah dilengkapi built-in sensor untuk memperingatkan pengemudi ketika tekanan ban rendah. Sejauh ini, IoT paling erat hubungannya dengan komunikasi *machine-to-machine* (M2M) di bidang manufaktur dan listrik, perminyakan, dan gas. Produk dibangun dengan kemampuan komunikasi M2M yang sering disebut dengan sistem cerdas atau “*smart*”. Sebagai contoh yaitu smart kabel, smart meter, smart grid sensor.

Penelitian pada IoT masih dalam tahap pengembangan. Berikut adalah beberapa definisi alternatif dikemukakan untuk memahami *Internet of Things* (IoT).

Definisi awal IoT adalah *Internet of Things* memiliki potensi untuk mengubah dunia seperti pernah dilakukan oleh Internet, bahkan mungkin lebih baik. Pernyataan tersebut diambil dari artikel sebagai berikut: “Hari ini komputer dan manusia, hampir sepenuhnya tergantung pada Internet untuk segala informasi yang semua terdiri dari sekitar 50 petabyte (satu petabyte adalah 1.024 terabyte) data yang tersedia pada Internet dan pertama kali digagas dan diciptakan oleh manusia. Dari mulai magnetik, menekan tombol rekam, mengambil gambar digital atau memindai kode bar (Ashton, 2009).

Diagram konvensional dari Internet meninggalkan *router* menjadi bagian terpenting dari semuanya. Masalahnya adalah orang memiliki waktu, perhatian dan akurasi terbatas. Mereka semua berarti tidak sangat baik dalam menangkap berbagai data tentang hal di dunia nyata.

Dari segi fisik dan begitu juga lingkungan kita. Gagasan dan informasi begitu penting, tetapi banyak lagi hal yang penting. Namun teknologi informasi saat ini sangat tergantung pada data yang berasal dari orang-orang sehingga komputer kita tahu lebih banyak tentang semua ide dari hal-hal tersebut.

1.2.2. Cara kerja *Automatic Feeder*

Automatic feeder merupakan suatu alat yang difungsikan untuk membantu atau mempermudah para petani atau pembudidaya udang dalam pemberian pakan yang menggunakan pelet agar pemberian pakan dapat dilakukan tepat waktu dan merata, yang telah diatur dan mengacu pada jadwal makan per hari, waktu dan jumlah pakan sesuai dengan yang diinginkan. Alat ini bekerja secara otomatis dalam proses pengeluaran dan penyebaran pakan dengan kontrol Arduino. Arduino ini berfungsi sebagai media kontrol jalur keluar pakan, setingan waktu makan, jarak lontaran pakan dan banyaknya pakan yang dikeluarkan.

Jalur keluarnya pakan secara otomatis di atur menggunakan motor DC yang dikontrol melalui Arduino. Sistem kontrol Arduino Mega 2560 ini diprogram dengan menggunakan komputer yang sudah dilengkapi dengan *software* Arduino.

Alat ini akan beroperasi secara otomatis dengan menggunakan *Real Time Clock* (RTC). Sementara untuk pemonitorannya menggunakan tampilan web yang terkoneksi dengan sensor ultrasonik dan NodeMCU. Pada sistem ini sensor ultrasonik di letakkan di dalam penutup wadah pakan, yang berfungsi untuk memonitor jumlah pakan. *Output* pada alat ini akan mengaktifkan sebuah motor DC 24 volt yang berfungsi sebagai pelontar pakan dan sebuah motor DC 12 volt yang berfungsi sebagai pemutar penampung pakan berdasarkan waktu yang telah diatur pada RTC. RTC berfungsi sebagai pewaktu alat. Sehingga pengguna dapat memonitoring alat dari kejauhan.

Untuk penggunaan motor DC, kecepatan motor DC ini akan diatur menggunakan *pulse width modulation* (PWM) yang kecepataannya dapat diatur melalui program, hal ini bertujuan agar pemberian pakan terjadi secara merata. Berikut ini adalah Gambar 2.2 *automatic feeder* yang telah ada.



Gambar 2. 2 automatic feeder

1.3. Database

Database atau sering kita kenal basis data merupakan sekumpulan data yang tersusun dan tersimpan rapi dalam komputer, dan dapat diolah maupun dimanipulasi dengan menggunakan *software* atau perangkat lunak untuk dijadikan sebagai informasi. *Database* adalah kumpulan informasi atau data yang tersimpan secara sistematis sehingga temu kembali informasinya menjadi mudah dan cepat (Kusmayadi, 2011).

Fathansyah (2004) menyatakan beberapa pengertian *database* yaitu sebagai berikut :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganiasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Connolly dan Begg (2010) berkata *database* adalah sekumpulan data tersebar yang berhubungan secara logis, dan penjelasan dari data ini dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi.

Inmon (2005) berkata *database* adalah sekumpulan data yang saling berhubungan yang disimpan (biasanya dengan redudansi yang terkontrol dan

terbatas) berdasarkan skema. Sebuah *database* dapat melayani *single* atau *multiple applications*.

Gottschalk dan Saether (2010) berkata *database* adalah sekumpulan data yang terorganisir untuk mendukung banyak aplikasi secara efisien dengan memusatkan data dan mengontrol data *redundant*.

Al-Bahra Bin Ladjamudin (2005) menyatakan terdapat beberapa definisi basis data dari beberapa orang ahli basis data sebagai berikut :

1. *Database* adalah sekumpulan data store (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk*, *optical disk*, *magnetic drum* atau media penyimpanan sekunder lainnya.
2. *Database* adalah sekumpulan program – program aplikasi umum yang bersifat “*batch*” yang mengeksekusi dan memproses data secara umum (seperti pencarian, peremajaan, penambahan, dan penghapusan terhadap data).
3. *Database* terdiri dari data yang akan digunakan atau diperuntukan terhadap banyak *user*, dimana masing – masing *user* (baik menggunakan teknik pemrosesan yang bersifat *batch* atau *on-line*) akan menggunakan data tersebut sesuai dengan tugas dan fungsinya, dan *user* lain dapat juga menggunakan data tersebut dalam waktu yang bersamaan.
4. *Database* adalah koneksi terpadu dari data – data yang saling berkaitan dari suatu *enterprise* (perusahaan, instansi pemerintah atau swasta).
5. *Database* adalah koleksi atau kumpulan data yang mekanis, terbagi/shared, terdefinisi secara formal dan dikontrol terpusat pada organisasi (*Everest*).

Dari pengertian di atas dapat dinyatakan bahwa database yaitu kumpulan-kumpulan data yang berisi informasi yang terhubung satu sama lain yang diorganisasikan dengan struktur tertentu serta dapat ditemukan dengan mudah dan cepat menggunakan bantuan komputer.

BAB III

METODE PELAKSANAAN

Dalam proses pembuatan Proyek Akhir yang berjudul “Modifikasi Rancang Bangun Alat Pemberi Pakan Udang Vaname (*Semi automatic feeder*) Dibawah Umur 30 Hari Berbasis *Internet Of Things (IoT)*” terdiri dari beberapa tahapan berikut :

1. Studi literatur dan pengumpulan data
2. Desain sistem *automatic feeder*
3. Pembuatan sistem mekanik dan elektronik.
4. Desain dan pembuatan perangkat lunak.
5. Pembuatan *database*
6. Pengujian.

Penjelasan lebih lanjut dari tahapan di atas dapat dilihat pada langkah-langkah berikut ini:

1.1. Tahap Pertama

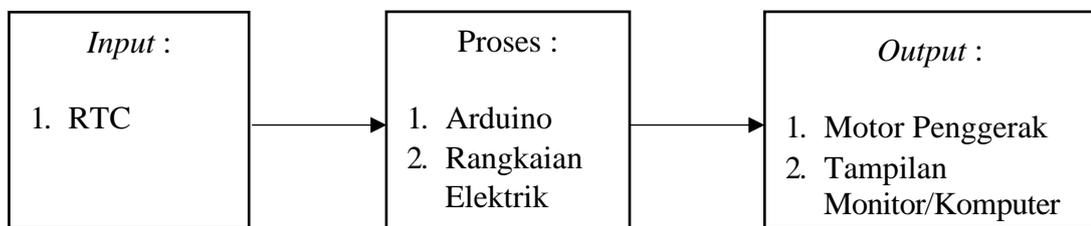
Melakukan studi literatur dan survei ke lapangan tentang *automatic feeder* pada pakan udang dan *database* yang diperlukan di lapangan. Studi literatur dan pengumpulan data ini bertujuan untuk mengetahui secara garis besar tentang *automatic feeder* yang digunakan ditambah udangdan untuk mengatasi permasalahan yang terjadi, serta memenuhi apa yang diperlukan di lapangan. Data yang telah terkumpul dijadikan acuan untuk tahapan proses pembuatan *automatic feeder* dan *database* selanjutnya.

1.2. Tahap Kedua

Mendesain proyek akhir “Modifikasi Rancang Bangun Alat Pemberi Pakan Udang Vaname (*Semi automatic feeder*) Dibawah Umur 30 Hari Berbasis

internet of things (IoT)” baik secara *hardware* dan *software*. Tujuannya agar mempermudah dalam proses pembuatan tugas akhir ini. Perancangan *hardware* ini berupa perancangan mekanik dengan menggunakan *software* yang ada, dan perancangan elektrik yang dibutuhkan dalam *automatic feeder* itu sendiri menggunakan *software* Arduino sebagai sistem kontrol nya. Serta pembuatan *database* menggunakan aplikasi XAMPP dan untuk menampilkannya menggunakan *website*.

Berikut ini adalah diagram blok secara umum dari *automatic feeder* yang akan dibuat :



Gambar 3. 1 Diagram blok *automatic feeder* secara umum

1.3. Tahap Ketiga

Pembuatan *hardware* dan *software*. Untuk pembuatan *hardware* berupa pembuatan konstruksi mekanik yang berupa penentuan bahan yang digunakan, bagian-bagian alat serta ukuran yang digunakan. Perencanaan ini harus dilakukan dan difikirkan secara matang-matang hal ini bertujuan untuk mengurangi kesalahan pada saat pembuatan alat nantinya.

Berikut adalah proses pembuatan alat pada bagian mekanik yang meliputi:

1. Penentuan dan pembuatan wadah penampung pakan dengan menggunakan plat dengan ketebalan 0.6 mm.
2. Pembuatan pipa penampung dan penyalur pakan, dengan menggunakan plastik 3D *printing*.
3. Pelontar pakan menggunakan alat yang sudah ada, yaitu *steam* pencucian.
4. Pemasangan tata letak motor DC 12 volt sebagai media pemutar

penampung pakan.

5. Pemasangan tata letak motor DC 24 volt sebagai penggerak pelontar.
6. Pembuatan kaki atau tiang penyangga *automatic feeder* beserta penguncinya.
7. Perakitan bagian-bagian mekanik *automatic feeder*.

Sedangkan proses pembuatan alat pada bagian sistem kontrol meliputi :

1. Pemrograman *ON/OFF automatic feeder*
2. Pengaturan putaran motor DC 12 volt dan 24 volt
3. Pemrograman kontrol monitoring jumlah pakan menggunakan sensor ultrasonik
4. Pemrograman monitoring waktu pemberian pakan menggunakan RTC
5. Pemrograman *database* menggunakan MySQL

1.4. Tahap Keempat

Setelah proses pembuatan *hardware* dan *software* selesai, langkah selanjutnya yaitu proses uji coba alat. Uji coba ini bertujuan untuk mengetahui kerja alat apakah bekerja secara optimal dan berfungsi sesuai dengan fungsi yang diinginkan. Uji coba ini terdapat dua tahapan :

- Uji Coba *Hardware*

Uji coba *hardware* yang dilakukan meliputi:

1. Uji coba jarak lontaran pakan
2. Uji coba keluaran pakan

- Uji Coba *Software*

Uji coba *software* yang dilakukan meliputi:

1. Uji coba koneksi antara Arduino, komputer, dan Node MCU
2. Uji coba koneksi antara Arduino dan RTC
3. Uji coba koneksi antara NodeMCU dan sensor ultrasonik
4. Uji coba koneksi *database* ke alat

- Uji Coba Keseluruhan

Uji coba secara keseluruhan ini dilakukan untuk mengetahui hasil pengujian dari keseluruhan alat yang telah dibuat.

1.5. Tahap Kelima

Tahap ini merupakan tahap pembuatan *database* dimana *software* yang digunakan adalah XAMPP. *Database* ini berfungsi sebagai penyimpan data sisa pakan dari sensor.

1.6. Tahap Keenam

Pada tahap ini merupakan tahap analisis data pada hasil uji coba yang telah dilakukan. Hal ini bertujuan untuk melihat kekurangan terhadap alat yang telah dibuat baik dari segi konstruksi, rangkaian kontrol, *database* serta program yang dibuat.

BAB IV

PEMBAHASAN

Bab pembahasan ini menguraikan proses pengerjaan proyek akhir ini berdasarkan metode yang telah dipaparkan pada bab sebelumnya. Secara umum bab ini menguraikan tentang:

1. Deskripsi alat
2. Sistem mekanik dalam pembuatan *automatic feeder*
3. Rangkaian kontrol dan komunikasi
4. Pengujian *hardware*
5. Pengujian *software*

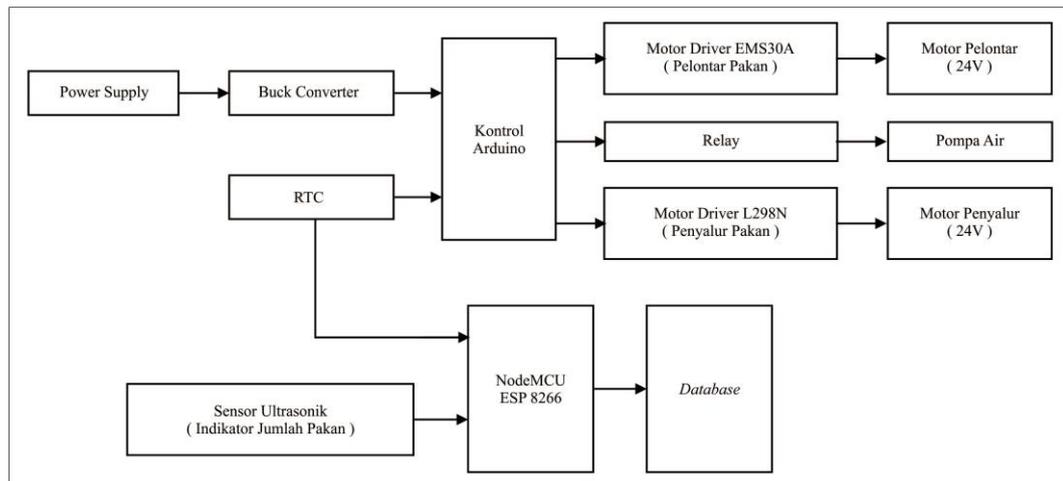
4.1. Deskripsi Alat

Automatic feeder berbasis IoT ini adalah sebuah alat yang digunakan untuk memberikan pakan udang di tambak udang. Alat ini akan beroperasi secara otomatis dengan diaktifkan melalui Arduino Mega 2560 menggunakan RTC. Sementara untuk pengontrolan dan pemonitorannya menggunakan sebuah Arduino Mega 2560 yang telah terprogram beserta sebuah NodeMCU ESP 8266 sebagai pengirim data yang akan disimpan di *database* dan ditampilkan di PC. Pada sistem ini menggunakan sebuah sensor ultrasonik yang di letakkan di dalam penutup wadah pakan yang berfungsi untuk memonitoring jumlah sisa pakan.

Output pada Arduino akan mengaktifkan sebuah motor DC 24 volt yang berfungsi sebagai pelontar pakan dan sebuah motor DC 12 volt yang berfungsi sebagai pemutar penampung pakan berdasarkan waktu yang telah diatur pada RTC. RTC berfungsi sebagai pewaktu yang disertai dengan pemberitahuan. Pemberitahuan ini berupa indikator lampu yang menyala pada mesin, sebagai penanda alat sedang bekerja/proses sedang berjalan dan notifikasi bahwa proses telah selesai dilaksanakan pada waktu yang sudah ditentukan. Perbedaan

dengan *automatic feeder* sebelumnya yaitu pada *automatic feeder* yang diperbaharui ini digunakan untuk udang yang berumur dibawah 30 hari.

Berikut adalah gambar diagram blok *automatic feeder* berbasis *Internet of Things* yang terdapat pada gambar 4.1.



Gambar 4. 1 Diagram blok *automatic feeder*

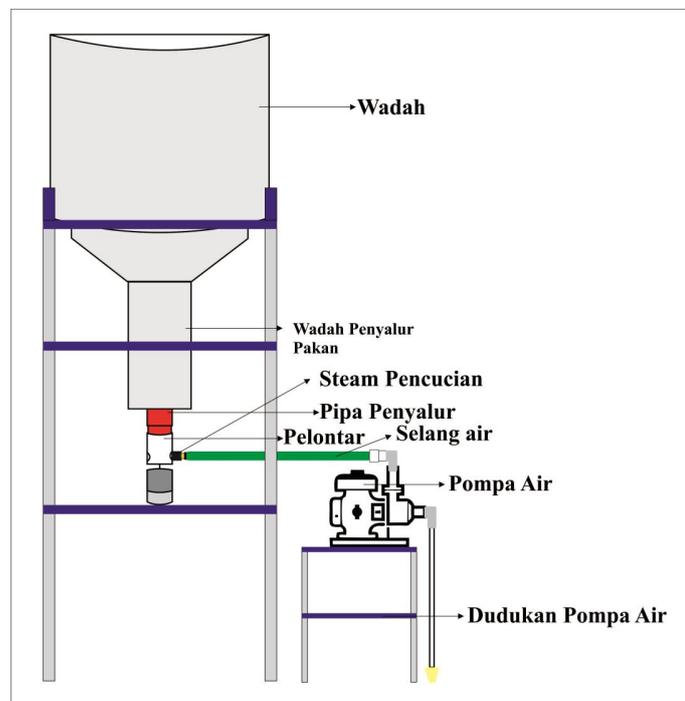
PC sebagai monitoring data sisa pakan dari sensor ultasonik yang dikirim melalui NodeMCU ESP8266. *Power supply* pada rangkaian kontrol ini berfungsi sebagai sumber dari tegangan 220V AC ke 12V DC. Keluaran *power supply* 12 volt DC kemudian terhubung ke rangkaian *Buck Converter*. RTC pada rangkaian kontrol berfungsi sebagai *input* Arduino dan NodeMCU. *Output power supply* 12 volt juga masuk ke dalam rangkaian motor *driver* L298N, motor driver EMS30A, dan *output buck converter* masuk ke Arduino Mega 2560. Tegangan *output* dari Arduino Mega 2560 sebesar 5V terhubung ke driver motor untuk mengaktifkan motor pengatur pakan dan motor pelontar pakan sesuai dengan waktu yang telah ditentukan.

4.2. Sistem Mekanik *automatic feeder*

Untuk perancangan mekanik pada *automatic feeder* ini kami menggunakan aplikasi *solidwork 17* dan *coreldraw x7*. Rancangan ini menggambarkan secara garis besar bagaimana konstruksi yang akan dibuat. Mekanisme kerja alat ini

adalah saat *automatic feeder* diaktifkan, komponen elektrik seperti sensor ultrasonik, driver motor dan relay akan bekerja pada saat waktu yang telah ditentukan menggunakan RTC. Relay disini berfungsi untuk menghidupkan pompa air, yang dimana pompa air akan mendorong pakan yang mengalir dari penyalur ke pelontar, setelah itu pelontar akan melontarkan pakan mulai dari waktu yang telah ditentukan pada RTC.

Perancangan sistem mekanik ini dimulai dari menentukan bagian-bagian pada alat serta menentukan bentuk yang sesuai dengan bagian alat yang ingin dibuat. Gambar 4.2 di bawah ini menunjukkan gambar secara keseluruhan sistem mekanik *automatic feeder* pada proyek akhir ini.



Gambar 4. 2 Konstruksi *automatic feeder*

Pembuatan sistem mekanik dilakukan dalam beberapa tahapan. Tahapan ini berfungsi sebagai acuan dalam proses pembuatan proyek akhir. Berikut adalah tahapan-tahapan dalam proses pembuatan mekanik *automatic feeder*.

1. Perancangan mekanik *automatic feeder*
2. Penentuan wadah penampung pakan
3. Pembuatan corong penyalur pakan

4. Pembuatan penguat sisi keliling corong
5. Pembuatan penampung dan penyalur pakan
6. Peletakan motor DC 12 volt
7. Pembuatan pelontar pakan
8. Peletakan motor DC 24 volt
9. Peletakan *power supply*
10. Pembuatan pengunci wadah
11. Pembuatan tiang pengunci
12. Peletakan pompa air
13. Pembuatan rangka untuk dudukan pompa air
14. Pembuatan pengaman motor DC
15. Perakitan bagian-bagian mekanik

Untuk penjelasan lebih lanjut tentang pembuatan mekanik *automatic feeder* dapat dilihat pada pembahasan di bawah ini:

4.2.1. Wadah Penampung Pakan

Wadah penampung pakan merupakan tempat untuk meletakkan pakan udang yang akan ditebar di kolam tambak. Untuk bahan material wadah penampung pakan ini menggunakan material plat dengan ketebalan 0.6 mm. Diameter wadah ini sebesar 50 cm dan ketinggian wadah 40 cm. Wadah ini dapat menampung pakan sebanyak 35 kg. Gambar 4.3 menunjukkan gambar wadah penampung pakan.



Gambar 4. 3 Wadah penampung pakan

4.2.2. Corong Penyalur Pakan

Corong penyalur pakan ini merupakan corong yang berfungsi sebagai tempat penyalur pakan setelah wadah. Pembuatan corong penyalur pakan ini meliputi pemilihan bahan material yang tepat dan sesuai. Oleh karena itu dipilihlah tutup dari dandang nasi yang berbentuk krucut dengan ukuran diameter 40 cm dan plat dengan ketebalan 0.8 mm, yang kemudian digabungkan hingga menjadi corong penyalur pakan yang sesuai dengan kebutuhan yang diinginkan. Gambar 4.4 di bawah ini merupakan gambar dari corong penyalur pakan.



Gambar 4. 4 Corong penyalur pakan

4.2.3. Corong Penyalur Pakan

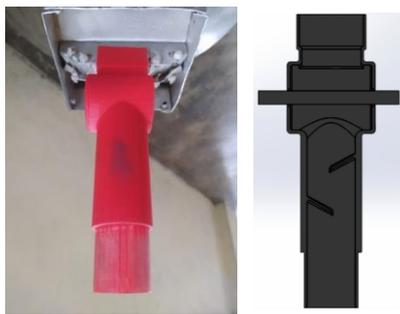
Setelah corong selesai, langkah selanjutnya yaitu membuat penguat sisi keliling corong. Penguat ini menggunakan material berupa plat besi yang diletakkan di sekeliling bagian atas dan samping corong. Penguat ini berfungsi untuk melindungi corong agar tidak mudah penyok saat tertekan beban pakan. Proses pembuatan penguat ini menggunakan bahan material plat dengan ketebalan 2 mm. Gambar penguat pada sisi keliling corong dapat dilihat pada gambar 4.5 di bawah ini.



Gambar 4. 5 Penguat sisi corong

4.2.4. Penampung dan Penyalur Pakan

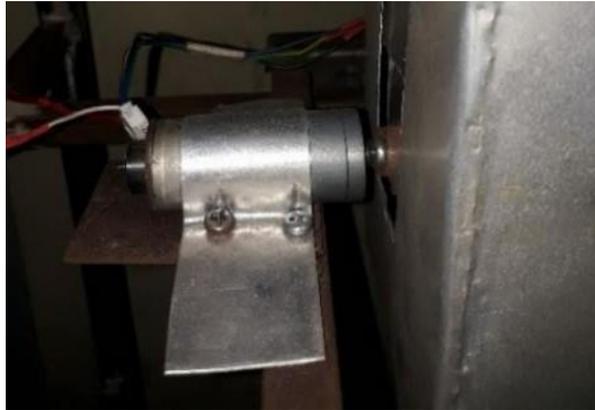
Penampung dan penyalur pakan merupakan bagian yang berfungsi sebagai penyalur pakan dari corong, kemudian menyalurkan pakan tersebut ke dalam pelontar pakan. Untuk itu kami menggunakan bahan plastik, yang dibuat dengan 3D Printing, dikarenakan bahan ini terbilang ringan dan telah memiliki bentuk sebagaimana yang diinginkan, dan dibagian penampung pakan terdapat ruang yang berfungsi sebagai pengatur keluaran pakan yang berbentuk seperti pintu. Kemudian, pada bagian bawah penyalur pakan dibuat *space* dengan kemiringan 45 derajat agar pakan yang turun dari penampung tidak langsung jatuh ke pelontar yang akan membuat pelontar tersumbat. Gambar 4.6 di bawah ini merupakan gambar penampung dan penyalur pakan.



Gambar 4. 6 Penampung dan penyalur pakan

4.2.5. Motor DC 12V

Motor DC 12 volt ini berfungsi sebagai penggerak penampung dan penyalur pakan, jadi pakan yang disalurkan dari corong akan masuk kedalam penampung pakan, setelah itu motor DC 12 volt ini akan memutar penampung pakan sehingga pakan di dalam penampung akan tersalurkan menuju pelontar pakan. Motor DC ini diletakkan di samping penampung pakan. Kecepatan putaran motor DC ini sebesar 100 RPM. Motor DC ini dapat menahan beban maksimal 3 kg. Gambar 4.7 di bawah ini merupakan penempatan motor DC 12 volt.



Gambar 4. 7 Peletakkan motor DC 12 volt

4.2.6. Pelontar Pakan

Pelontar pakan merupakan bagian alat yang berfungsi sebagai pelontar pakan udang. Proses pembuatan pelontar pakan ini menggunakan *3D printing* sebagai *basenya*(warna putih). Bagian pendorong pakan menggunakan kepala stem pencucian yang dimasukkan kedalam base, lalu di lem mneggunakan lem pipa dan lem tembak seabai perekat. Gambar 4.8 di bawah ini adalah gambar pelontar pakan.



Gambar 4. 8 Pelontar pakan

4.2.7. Motor DC 24 volt

Motor DC 24 volt berfungsi sebagai penggerak pelontar. Motor DC 24 volt ini diletakkan di bagian bawah pelontar pakan. RPM maksimal motor sebesar 21.000 pada tegangan maksimal . Karena menggunakan *power supply* 12V, maka RPM yang digunakan menjadi setengah dari RPM maksimal yaitu 10.500 dan ini sudah dibuktikan menggunakan alat ukur tacho meter. Kecepatan motor DC ini diatur *PWM high* atau maksimal 255. Hal ini bertujuan agar pakan yang dilontarkan tertebat secara merata. Gambar 4.9 di bawah ini adalah gambar peletakan motor DC 24 volt pada *automatic feeder* dan bukti pengukuran menggunakan tacho meter.



Gambar 4. 9 Penempatan motor DC 24 volt Pengukuran menggunakan Tacho meter

4.2.8. Power Supply

Power supply berfungsi sebagai pemberi suplai sumber untuk menghidupkan komponen seperti motor DC 12V, DC 24V, dan Buck Converter spek keluaran power supply ini yaitu 12V dan 20A. Power supply ini akan diletakan di bawah kaki penyanggah pompa air. Gambar 4.10 di bawah ini adalah gambar *Power Supply*.



Gambar 4. 10 Power Supply

4.2.9. Pompa Air

Pompa air disini berfungsi sebagai pendorong pakan yang turun dari penyalur. *datasheet* dari pompa air ini memiliki keluaran debit air sebesar 54 liter/menit. Tegangan yang digunakan yaitu 220V AC. Dengan spesifikasi daya *output* 125 watt, daya *input* start 450 watt, daya hisap 9m, dan daya dorong 30m. Gambar 4.12 di bawah ini adalah pompa air.



Gambar 4. 11 Pompa Air

4.2.10. Penyangga Kaki Pompa air

Penyangga kaki pompa air diletakan di bagian belakang konstruksi utama. Pada bagian ini box kontrol juga diletakan pada bagian atas penyangga agar mempermudah dalam perakitan rangkaian kontrol. Tujuan dibuatnya kaki penyanggah, untuk mempermudah keluaran pompa air yang dikeluarkan dari atas melalui bawah, Sehingga pergerakannya lancar. Gambar 4.13 dibawah ini merupakan dudukan pompa air.



Gambar 4. 12 Dudukan Pompa Air

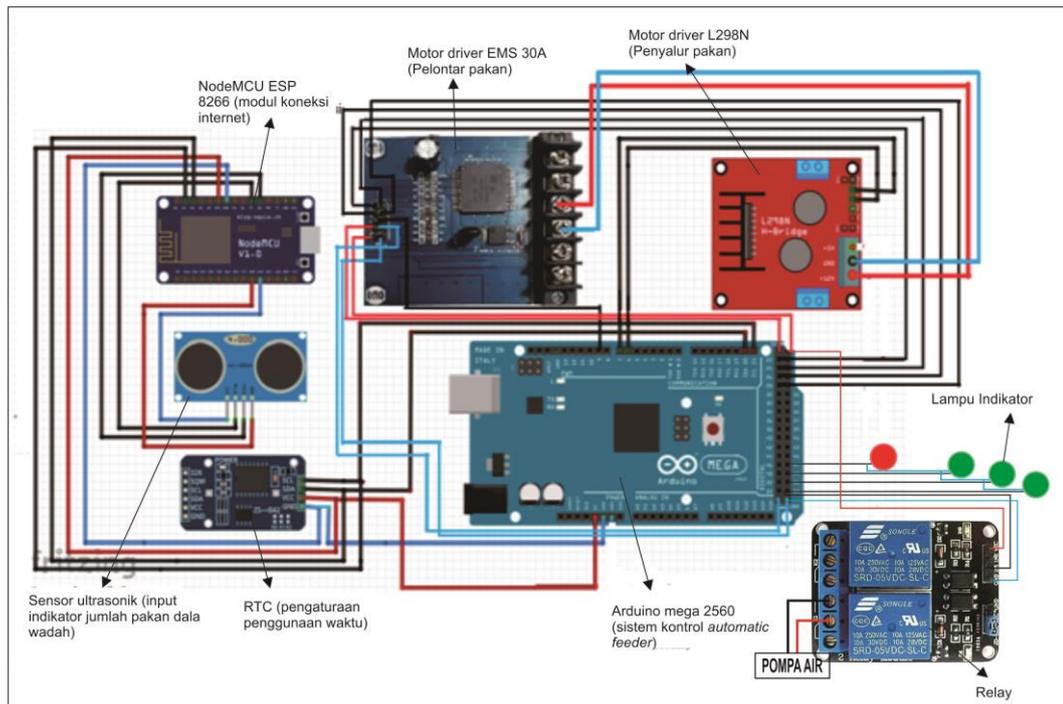
Setelah bagian-bagian pada alat mekanik telah selesai dibuat, maka langkah selanjutnya yaitu menggabungkan bagian mekanik tersebut. Pada tahap ini bagian- bagian yang telah dibuat di awal disatukan menjadi satu kesatuan. Gambar secara keseluruhan *automatic feeder* ini dapat dilihat pada gambar 4.14 berikut ini.



Gambar 4. 13 Konstruksi assembly automatic feeder

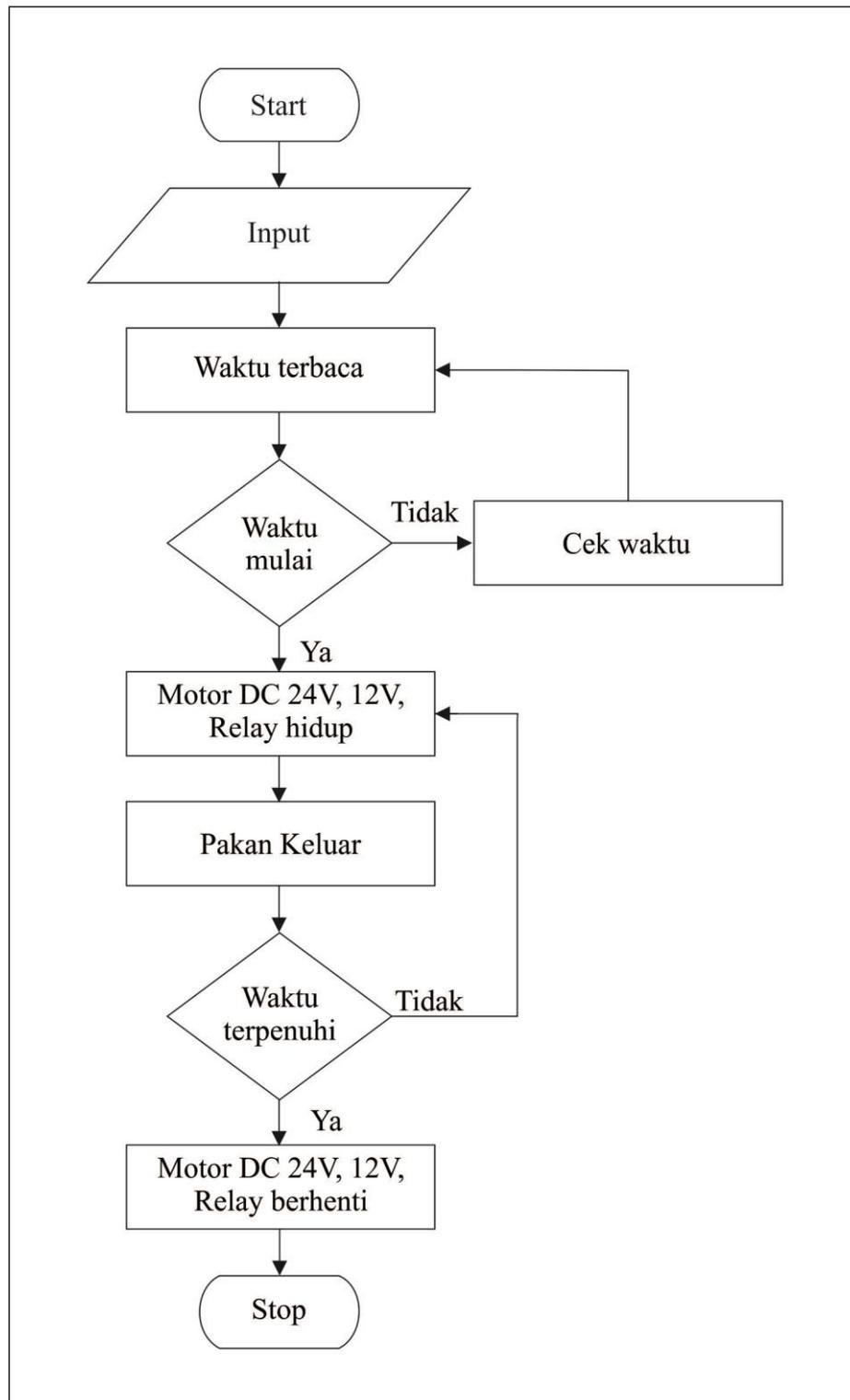
4.3. Rangkaian Kontrol dan Komunikasi

Rangkaian kontrol mesin *automatic feeder* berbasis IoT ini dirangkai di dalam sebuah box panel dengan ukuran 19 x 27 cm. Di dalam rangkaian ini terdapat sebuah Arduino Mega 2560, RTC, motor driver, sensor ultrasonik dan NodeMCU sebagai prantara komunikasi. Komponen - komponen tersebut disambungkan ke Arduino Mega 2560 sebagai sistem kontrol pada alat ini. Sehingga setiap komponen memiliki pin-pin tersendiri untuk disambungkan ke Arduino Mega 2560, pin-pin yang disambungkan ke Arduino ini harus sama ditulis persis seperti di dalam program Arduino. Jika pin di dalam program tidak sama antara sambungan komponen-komponen ke Arduino Mega 2560, maka alat juga tidak bisa berfungsi. Oleh karena itu diperlukannya ketelitian dan konsentrasi yang baik dalam perangkaian komponen serta pembuatan program Arduino untuk sistem kontrol *automatic feeder* ini. Untuk rangkaian kontrol dan komunikasi dapat dilihat pada gambar 4.15 di bawah ini.



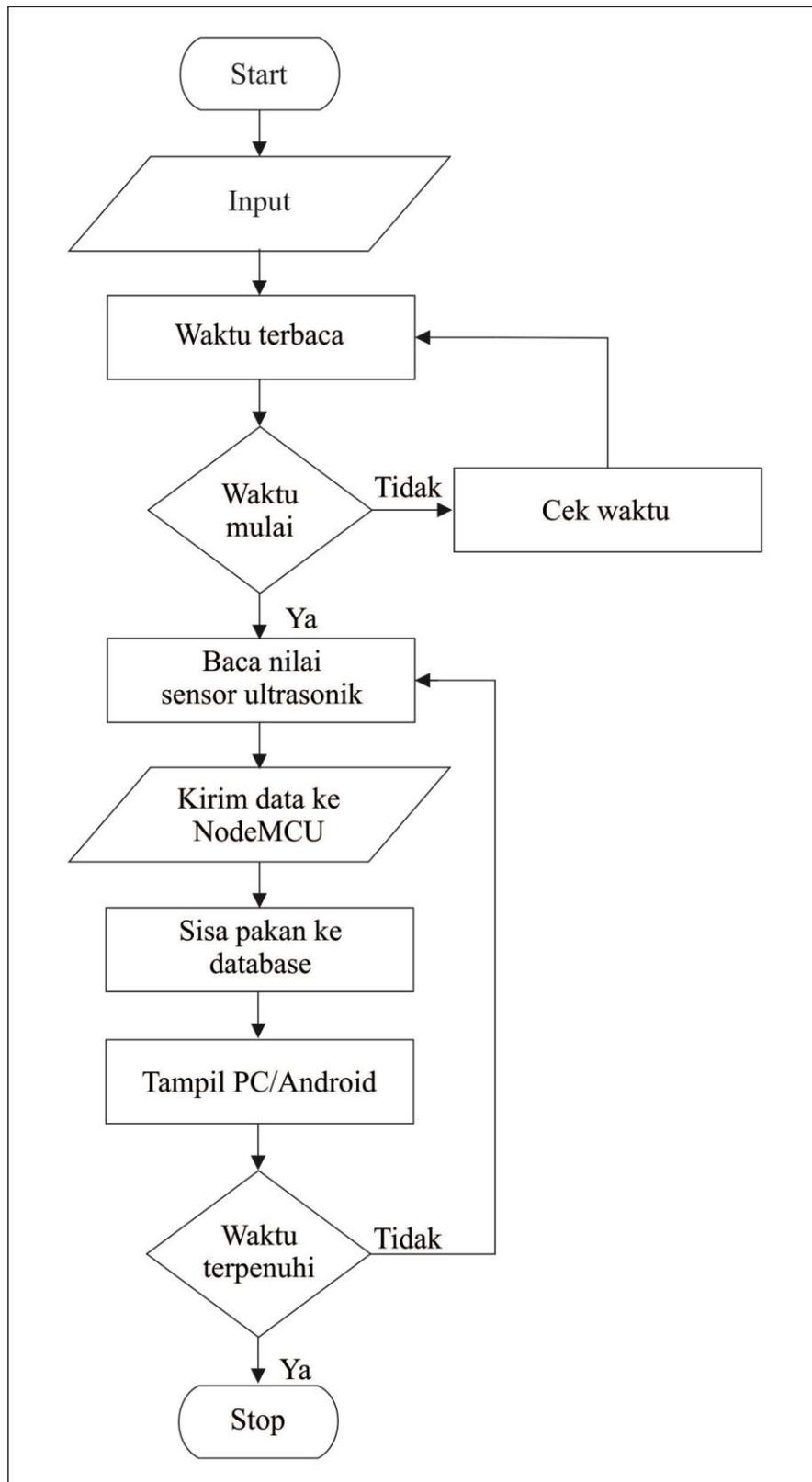
Gambar 4. 14 Rangkaian Kontrol dan Komunikasi

Skematik di atas menjelaskan rangkaian kontrol *hardware* yang digunakan pada alat *automatic feeder* berbasis IoT. Sistem kontrol disini menggunakan Arduino Mega 2560. Untuk penggerak motor DC 12 volt menggunakan motor driver L298N sedangkan untuk 24 volt menggunakan motor driver EMS30A, kemudian untuk pengiriman data sensor ke *database* menggunakan sebuah modul wifi yaitu NodeMCU ESP 8266. Setelah itu ada sensor ultrasonik yang berfungsi sebagai *input* untuk mengetahui sisa pakan yang masih ada di dalam wadah, kemudian untuk pengaturan waktu menggunakan 1 buah RTC. Berikut merupakan *flowchart* proses kerja Arduino terdapat pada gambar 4.16 di bawah ini.



Gambar 4. 15 Flowchart Progres kerja Arduino

Berikut merupakan flowchart progress kerja NodeMCU terdapat pada gambar 4.17 berikut ini.



Gambar 4. 16 Flowchart progress kerja NodeMCU

Pada flowchart progress kerja NodeMCU data didapatkan melalui sensor ultrasonik. Kemudian jika NodeMCU terhubung dengan internet maka NodeMCU akan mengirim data yang didapat dari sensor ke *database* yang akan ditampilkan di komputer melalui web.

4.3.1 Program Penyalan motor DC 24V dan 12V

Pemrograman utama ini terdapat pada Arduino yaitu pembacaan RTC dan pengontrolan motor. Untuk pemrograman penyalan motor DC dapat dilihat pada lampiran. Pada pemrograman penyalan motor DC dapat dijelaskan bahwa waktu untuk pemberian pakan secara otomatis telah terprogram di dalam RTC. Sehingga secara otomatis motor DC akan aktif saat waktu yang telah ditentukan terbaca. Jika waktu telah terpenuhi maka secara otomatis motor DC akan berhenti berputar, dan akan kembali menyala saat waktu yang telah ditentukan kembali terbaca.

4.3.2 Program Pengiriman Data Serial Sensor Ultrasonik ke NodeMCU

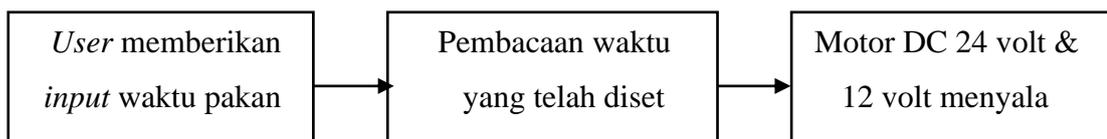
Pemrograman ini digunakan untuk mengirimkan data yang dibaca oleh NodeMCU kemudian akan dikirimkan ke *smartphone/PC*. Program ini bertujuan untuk menampilkan nilai dari sensor jumlah pakan yang akan dikonversikan ke dalam bentuk persen, serta waktu dan tanggal yang terprogram oleh RTC. Kemudian data dari sensor akan dikirim ke *database* dan kemudian ditampilkan di web melalui *PC/smartphone*.

4.4. Uji Coba Hardware

Pada bagian pengujian alat ini, dapat dilakukan hanya jika konstruksinya telah selesai dibuat. Pengujian yang dilakukan adalah sebagai berikut.

4.4.1. Pengujian Motor DC 12 volt & 24 volt

Uji coba yang dilakukan pada motor DC 24 volt ini untuk mengetahui apakah motor dapat menyala saat diberikan tegangan dari *power supply*. Setelah masuk ke dalam program, apakah motor dapat berputar seperti perintah yang telah terprogram. Kemudian saat waktu yang telah ditentukan tercapai, apakah motor akan berhenti. Kedua motor DC ini memiliki kecepatan putar/RPM yang berbeda. Motor DC 12 volt memiliki RPM sebesar 100 RPM dan motor DC 24 volt sebesar 21.000 RPM. Tahapan untuk uji coba motor DC ini dimulai dari pemberian waktu makan oleh *user*, saat waktu makan tercapai motor DC 24 volt akan berputar cw dan ccw, sesuai perintah yang diberikan. Untuk pergerakan putaran motor DC 24 volt diset 180 derajat menggunakan *delay* pada program Arduino. Kemudian motor DC 12 volt dan DC 24 volt akan mulai berputar pada saat bersamaan. Jika telah memenuhi kriteria yang diinginkan ini maka motor DC 12 dan 24 volt telah berfungsi dengan baik. Tahap terakhirnya yaitu saat waktu yang diberikan telah habis maka motor DC 12 volt dan 24 volt akan berhenti secara bersamaan, jika motor DC berhenti secara bersamaan maka pengujian motor DC 12 volt dan 24 volt ini telah berhasil. Gambar 4.18 di bawah ini merupakan blok diagram uji coba penyalan motor DC 12 volt & 24 volt.



Gambar 4. 17 Blok diagram uji coba penyalan motor DC 12 volt & 24 volt.

4.4.2. Pengujian Jarak Lontaran Pakan

Uji coba jarak lontaran pakan ini bertujuan untuk mencapai jarak lontaran pakan maksimum. Karena semakin jauh jarak lontaran maksimum nya, maka semakin baik. Selain itu juga lontaran maksimum ini terdapat di dalam tuntutan proyek akhir ini. Untuk hasil pengujian jarak lontaran pakan secara manual dapat dilihat di tabel 4.1 berikut ini.

Tabel 4. 1 Pengujian Jarak Lontaran Pakan secara manual

Pelontar	Ukuran (diameter)	Jarak Lontar	Tinggi pelontar dari tanah
Selang	2 cm	1 m	90 cm
	5 mm	2,5 m	90 cm
	2 mm	5 m	90 cm
Kepala Steam	-	6 m	90 cm

Maksud dari uji coba secara manual ini sebagai acuan, apakah lontaran pakan tersebut bisa mencapai jarak yang diinginkan. Pengambilan data tersebut dilakukan secara manual dengan satu orang memegang pelontar mengikuti dari data yang tertera diatas. Untuk hasil pengujian jarak lontaran pakan menggunakan mesin dapat dilihat di tabel 4.2 berikut ini.

Tabel 4. 2 Pengujian Jarak Lontaran Pakan menggunakan mesin

Pelontar	Ukuran (diameter)	Jarak Lontar	Tinggi pelontar dari tanah
Kepala Steam	-	1,5 m	45 cm

Pada pengujian ini dapat dibandingkan hasil pengujian menggunakan mesin dan secara manual. Pengujian menggunakan mesin mengalami penurunan karena beberapa hal yang menyebabkan lontaran pakan tidak maksimal seperti, penggunaan motor penggerak yang tidak sesuai, perhitungan sudut lontar, dan masih banyak lagi.

4.4.3. Pengujian Keluaran Pakan

Keluaran pakan ini diuji untuk mengetahui apakah alat penampung dan penyalur pakan dapat berfungsi seperti apa yang diharapkan. Pengujian ini untuk melihat apakah pakan dapat tersalurkan menuju pelontar dengan lancar atau

mampet pada satu keadaan. Hal lain yang dilihat dari pengujian ini yaitu untuk mengetahui apakah pakan udang saat keluar dari wadah ada yang berceceran/terjatuh ke tanah yang akan membuat alat ini menjadi tidak produktif.

Pada pengujian keluaran pakan ini, pengujian lain yang dilakukan yaitu penghitungan berapa banyak pakan yang dikeluarkan oleh motor pelontar. Tujuannya yaitu untuk mengetahui garis besar berapa lama waktu yang terpakai untuk menghabiskan pakan sebesar 9 kg (1 kali makan). Pengujian ini dilakukan beberapa kali sampai ditemukannya hasil akhir yang pas untuk keluaran pakan ini. Pakan yang digunakan untuk uji coba ini sebanyak 1 kg, pengujian yang dilakukan ini untuk mencari seberapa lama waktu yang diperlukan *automatic feeder* dalam menghabiskan pakan sebanyak 1 kg. Pengujian ini dilakukan dengan merubah program pada putaran motor DC 12 volt sebagai pemutar penyalur pakan. Sehingga untuk memperkecil banyaknya pakan yang dikeluarkan penyalur pakan harus lebih diperlambat putarannya. Untuk hasil pengujian keluaran pakan per menit dapat dilihat pada tabel 4.3 berikut ini.

Tabel 4. 3 Pengujian Keluaran Pakan

Tegangan	RPM	Pakan	Waktu
12V	Max (100)	1Kg	47,09 detik
12V	Max (100)	2Kg	1,56 menit
12V	Max (100)	5Kg	3,92 menit

4.5. Uji Coba Software

Pengujian *software* ini dilakukan untuk mengetahui apakah alat yang telah dibuat sebelumnya dapat berfungsi jika telah digabungkan menggunakan sistem kontrol. Ada beberapa macam sistem kontrol yang terdapat pada alat ini. Sistem kontrol ini nantinya berfungsi sebagai kontrol untuk menggerakkan *automatic feeder*. Berikut adalah pengujian software pada *automatic feeder* berbasis IoT.

4.5.1. Pengujian antara Arduino, *Real Time Clock* (RTC)

Real Time Clock (RTC) berfungsi sebagai monitoring waktu pemberian pakan dan juga sebagai pemicu Bergeraknya kontrol Arduino dan NodeMCU. Waktu di *setting* sesuai dengan jadwal pemberian pakan. Oleh karena itu diperlukan pengkoneksian antara Arduino dan RTC. Proses pengujian ini yang pertama ketika waktu pemberian pakan telah di *setting* dalam program Arduino, dan sudah mencapai waktu maka motor DC 12V, 24V, dan relay aktif. Sebaliknya ketika proses pemberian pakan telah selesai maka motor DC 12V, 24V, dan relay berhenti. Berikut merupakan program pengujian Arduino dan RTC

```
//PAGI=====
if (now.minute() >= 6 &&now.minute() <= 6 ) {
  kecilcw();
  speedpwm = 255; //kecepatanpelontar
  analogWrite(pwm, speedpwm);
  digitalWrite(52, LOW);
  Serial.println("RELAY AKTIF");
  proses();
  cw();
  delay(500);
  ccw();
  delay(500);
  Stop();
  delay(300);
}
```

Pengaturan waktu pada RTC dan sistem kontrol bekerja, meliputi (Motor 24V, Motor 12V, dan Relay)

```

else if (now.minute() == 7 ) {
    kecilstop();
    digitalWrite(52, HIGH);
    Serial.println("RELAY OFF");
        delay(1000);
    digitalWrite(49, HIGH);
        Stop();
        delay(100);
    }

```

Sistem berhenti bekerja pada waktu yang telah ditentukan

Waktu diset sesuai dengan waktu real, jika waktu telah terpenuhi maka sistem kontrol akan bekerja, meliputi Motor DC 24V, Motor DC 12V, dan Relay. Untuk hasil pengujian RTC sistem kontrol hidup dapat dilihat pada gambar 4.19 berikut ini.

```

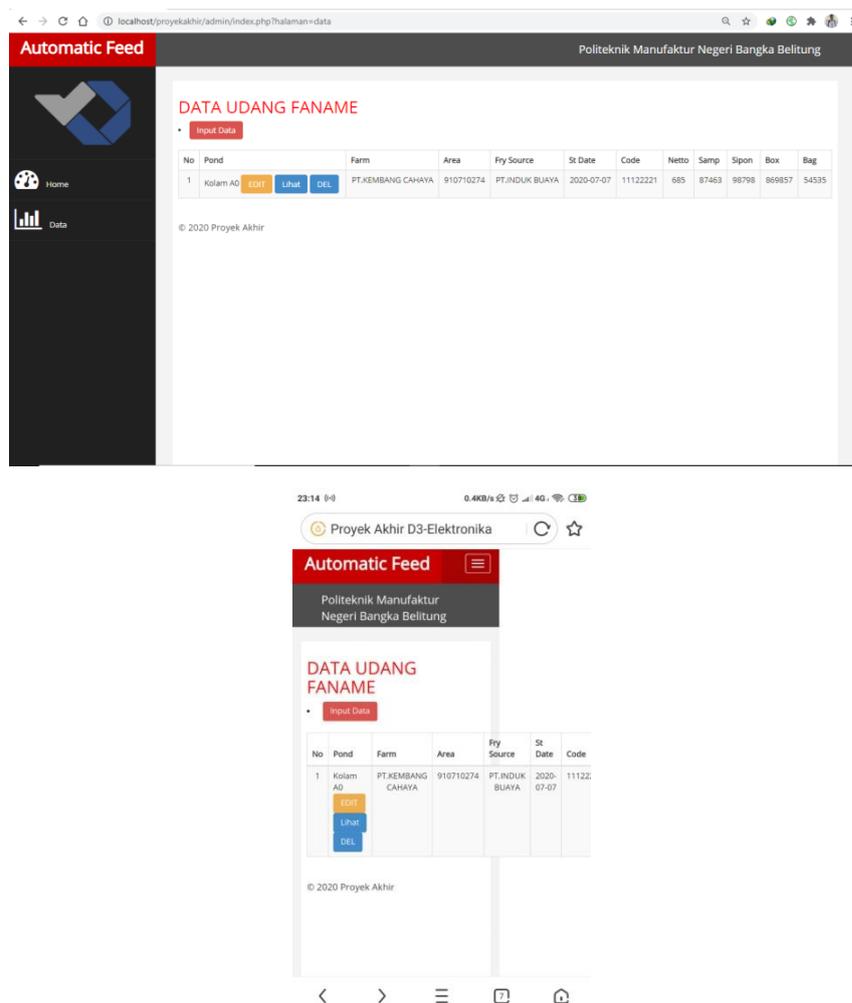
COM6
cwkecilRELAY AKTIF
cwccwstop16:6:5
cwkecilRELAY AKTIF
cwccwstop16:6:7
cwkecilRELAY AKTIF
cwccwstop16:6:10
cwkecilRELAY AKTIF
cwccwstop16:6:13
cwkecilRELAY AKTIF
cwccwstop16:6:15
cwkecilRELAY AKTIF
cwccwstop16:6:18
cwkecilRELAY AKTIF
cwccwstop16:6:20
cwkecilRELAY AKTIF
cwccw

```

Gambar 4. 18 hasil pengujian RTC sistem kontrol hidup

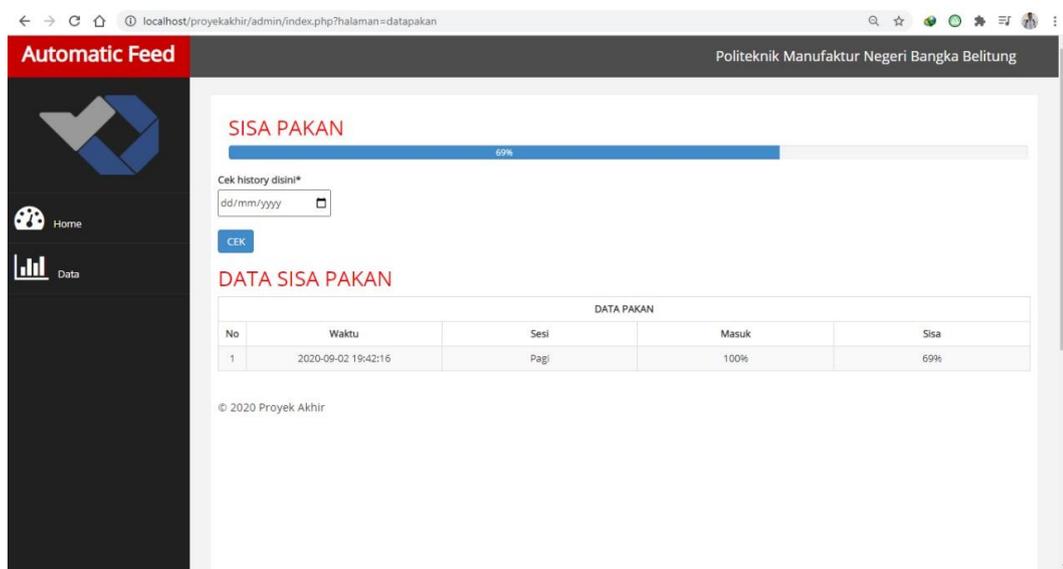
4.5.2. Pengujian antara Modul Wifi NodeMCU, Real Time Clock (RTC), Sensor Ultrasonik dan Database

Sensor ultrasonik berfungsi sebagai monitor sisa pakan dalam tabung, selanjutnya untuk RTC berfungsi sebagai waktu pemberian pakan. Sementara untuk pengkoneksiannya menggunakan Node MCU. Ketika waktu pemberian pakan telah mencapai waktu yang di inginkan maka sensor mendeteksi sisa pakan kemudian data dari sensor akan disimpan di *database*. Didalam *database* juga dapat menampilkan *progress bar* sisa pakan serta dapat melihat *history* tanggal sebelumnya, *database* ini dapat ditampilkan melalui *smartphone*/PC. Berikut adalah gambar tampilan pada *database* *smartphone*/PC. Berikut Gambar 4.20 Tampilan *database* di *smartphone* dan PC.



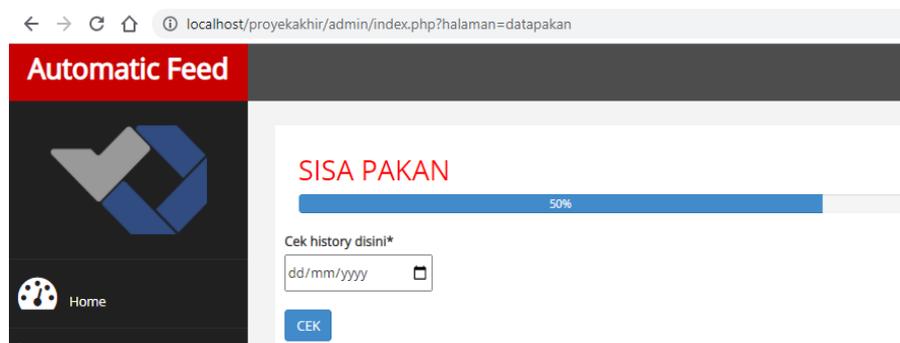
Gambar 4. 19 Tampilan Database di Smartphone dan PC

Pengerjaan alat untuk saat ini masih sebatas jaringan lokal, karena hanya bisa menggunakan satu koneksi jaringan yang sama untuk mengakses ke dalam web. Berdasarkan tuntutan seharusnya kerja alat ini bisa mengontrol waktu dengan PC/smartphone melalui website. Dikarenakan beberapa hal pengerjaan alat ini belum berfungsi secara IoT. Pengerjaan alat ini terlalu lama memakan waktu dibagian konstruksi dan hardware, sehingga kehabisan waktu untuk pembuatan software dan pengecekan *trouble*. Berikut gambar menu yang ada pada *database*.



Gambar 4. 20 Tampilan menu sisa pakan

Pada tampilan ini *user* dapat melihat *bar* sisa pakan dalam bentuk persen yang dikirim dari sensor ultrasonik melalui NodeMCU. Berikut Gambar 4.22 Progress bar.



Gambar 4. 21 Progres bar

Tidak hanya progress bar, *user* juga bisa mengecek histori sisa pakan pada hari-hari sebelumnya. Dengan mengklik gambar tanggal, pilih tanggal yang ingin dicek, lalu klik “CEK”.Berikut Gambar 4.23 Menu cek histori.

Cek history disini*

dd/mm/yyyy
📅

CEK

Gambar 4.22 Menu cek histori

Kemudian pada tampilan menu sisa pakan juga menampilkan data sisa pakan dan sesi waktu pemberian makan yang terdiri dari sesi pagi, siang, dan sore. Sesuai dengan waktu yang telah ditentukan.Pada tampilan data sisa pakan selain progress bar, data sisa pakan juga ditampilkan pada saat pemberian makan berlangsung.Misal, pada sesi pagi pakan yang masuk 100% dan sisa pakan yang telah dikeluarkan 53%. Maka, sisa pakan 53% tersimpan ke database dan data tersebut menjadi data masuk pada sesi siang.Begitu juga untuk sesi sore. Berikut Gambar 4.24 Data sisa pakan.

DATA SISA PAKAN

DATA PAKAN				
No	Waktu	Sesi	Masuk	Sisa
1	2020-08-28 04:51:00	Pagi	100%	53%
2	2020-08-28 04:52:02	Siang	53%	43%
3	2020-08-28 04:53:04	Sore	43%	0%

Gambar 4.23 Menu cek histori.

BAB V

PENUTUP

Setelah dilakukannya pembuatan dan pengujian sampai selesainya alat ini dibuat, kesimpulan dan saran yang dapat diambil yaitu :

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari pembuatan Proyek Akhir ini adalah :

1. *Automatic feeder* belum bisa dikontrol dari jarak jauh menggunakan PC dan *Android*.
2. *Automatic feeder* dapat melontarkan pakan dengan jarak sampai batas terjauh yaitu 1,5 meter, karena kurangnya perhitungan dipembuatan konstruksi.
3. Dengan menggunakan *smartphone* dan PC *user* dapat melihat data sisa pakan yang tersedia serta *database* penggunaan pakan.

5.2. Saran

Jika Proyek Akhir ini akan dikembangkan saran yang diberikan yaitu :

1. Alat ini juga sudah ada indikator berupa lampu pada saat proses pemberian pakan. Jika alat ini akan dikembangkan diharapkan alat dapat mengukur PH air pada tambak serta menampilkan alarm saat pakan telah habis.
2. Tampilan pada *smartphone* saat ini hanya bersifat memonitor data sisa pakan dari *database* penggunaan pakan. Jika alat ini akan dikembangkan diharapkan alat dapat mengatur dari PC dan *smartphone*, karena saat ini alat hanya dapat dimonitoring melalui PC dan *smartphone*.
3. *Automatic feeder* yang telah dibuat saat ini sudah bisa melontarkan pakan yang bersifat basah, tetapi konstruksinya belum sempurna, diharapkan kedepannya dapat disempurnakan lagi dengan jarak

lontar yang cukup jauh.

4. *Database* saat ini hanya dapat mengetahui jumlah pakan yang habis perhari, kedepannya diharapkan dapat mengetahui jumlah pakan perminggu, bulan dan tahun.

DAFTAR PUSTAKA

R. S. S. M. Anzullah, “Rancang Bangun Alat Pemberi Pakan Udang Vaname (Semi automatic feeder) di atas 30 Hari Berbasis IoT(Internet of Things),” Bangka Belitung, Politeknik Manufaktur Negeri Bangka Belitung, 2019.

Sarah R. Megumi, “Udang Vaname, Primadona Budidaya Perikanan,” [Online]. Available: <https://www.greeners.co/flora-fauna/udang-vaname-primadona-budidaya-perikanan/>. [Diakses 20 April 2020].

“Mari Mengenal Apa itu Internet of Thing (IoT),” 17 Juli 2016. [Online]. Available: <https://idcloudhost.com/mari-mengenal-apa-itu-internet-thing-iot/>. [Diakses 20 April 2020].

“PENGUNAAN automatic feeder TINGKATKAN EFISIENSI BUDIDAYA UDANG DI TAMBAK,” 12 April 2017. [Online]. Available: https://www.djpb.kkp.go.id/index.php/mobile/arsip/c/501/PENGUNAAN-AUTOMATIC-FEEDER-TINGKATKAN-EFISIENSI-BUDIDAYA-UDANG-DI-TAMBAK/?category_id=14. [Diakses 20 April 2020].

LAMPIRAN 1 : Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP



1. Data Pribadi

Nama Lengkap : Hanifah Thohiroh
Tempat & Tanggal Lahir : Sungailiat, 08 April 1999
Alamat Rumah : Jln. Cendrawasih 1 No. 19A
Hp : 082179545417
Email : thohirohanifah@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam

2. Riwayat Pendidikan

SD Negeri 8 Sungailiat	Lulus 2011
SMP Negeri 1 Sungailiat	Lulus 2014
SMK Negeri 1 Sungailiat	Lulus 2017

Sungailiat, 19 Agustus 2020

Hanifah Thohiroh

DAFTAR RIWAYAT HIDUP



1. Data Pribadi

Nama Lengkap : Harindra Bayu Aji Susanto
Tempat & Tanggal Lahir : Sungailiat, 21 Januari 2000
Alamat Rumah : Jln. Duyung Raya N0.50, BTN – Air Ruay
Hp : 082279992020
Email : santo3360@gmail.com
Jenis Kelamin : laki-laki
Agama : Islam

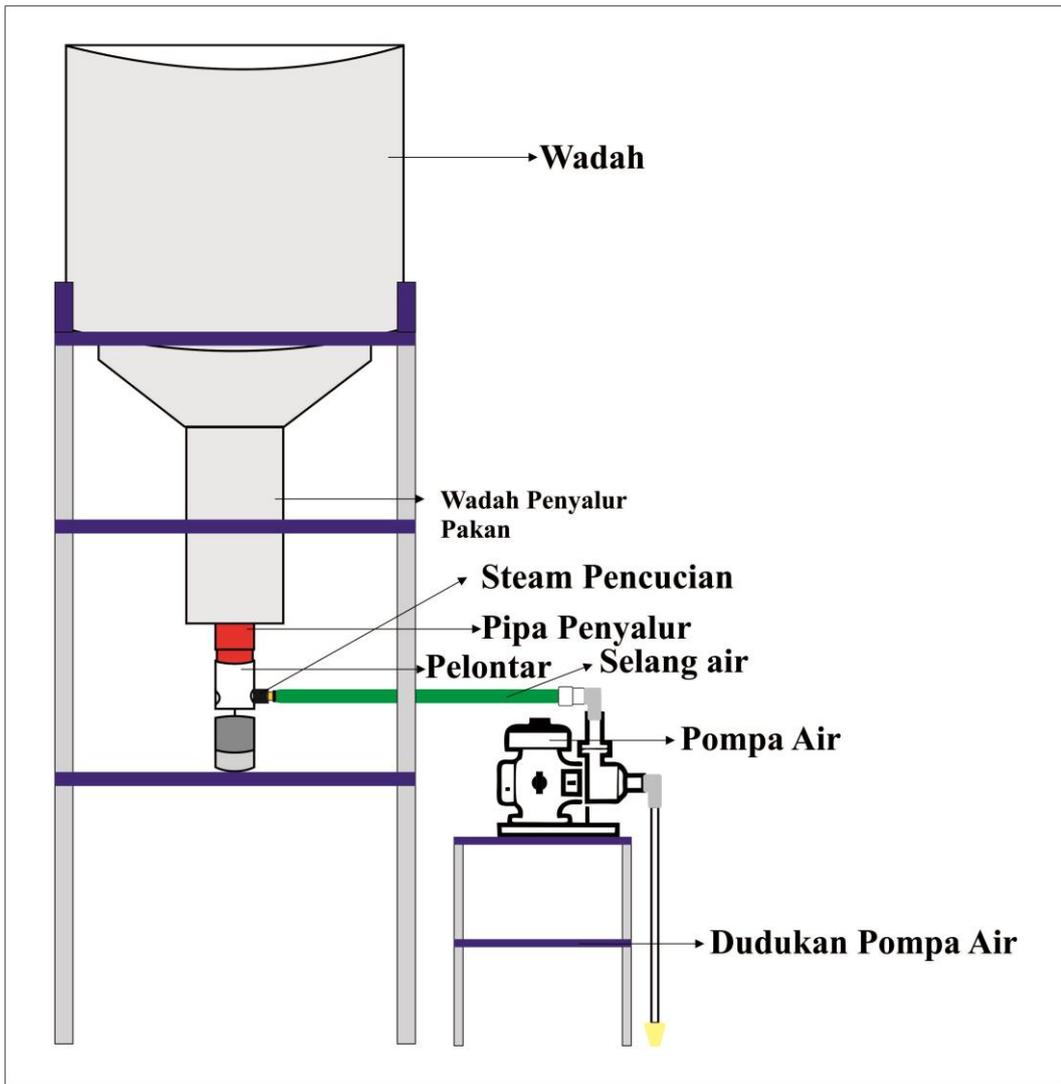
2. Riwayat Pendidikan

SD Muhammadiyah Sungailiat	Lulus 2011
SMP Negeri 1 Sungailiat	Lulus 2014
SMA Negeri 1 Sungailiat	Lulus 2017

Sungailiat, 19 Agustus 2020

Harindra Bayu Aji. S

LAMPIRAN 2 : Desain *AUTOMATIC FEEDER*



LAMPIRAN 3 : Program Arduino Mega 2560

```

//RTC
#include <Wire.h>
#include "RTCLib.h"
RTC_DS3231 rtc;
char daysOfTheWeek[7][12] = {"Sunda2y", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"};

//=====

//MOTOR
#define pwm 10
#define m1 23
#define m2 25
#define men1 27
#define men2 29

#define in1 8
#define in2 9
int speedpwm = 50;

//=====

void setup() {
  //LAMP
  pinMode(53, OUTPUT);
  pinMode(51, OUTPUT);
  pinMode(49, OUTPUT);
  pinMode(47, OUTPUT);

  digitalWrite(49, LOW);
  digitalWrite(53, LOW);
  digitalWrite(51, LOW);
  digitalWrite(47, LOW);

  //=====

  // MOTOR

```

```

pinMode(m1, OUTPUT);
pinMode(m2, OUTPUT);
pinMode(men1, OUTPUT);
pinMode(men2, OUTPUT);
pinMode(pwm, OUTPUT);

pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
Serial.begin(9600);
//=====

//RELAY
pinMode(52, OUTPUT);
digitalWrite(52, HIGH);
//=====

//RTC
Serial.begin(9600);
delay(3000);
if (! rtc.begin()) {
  Serial.println("Couldn't find RTC");
  while (1);
}
// if (rtc.lostPower()) {
//   Serial.println("RTC lost power, lets set the time!");
//   // following line sets the RTC to the date & time this sketch was compiled
//   rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
//   rtc.adjust(DateTime(2020, 8, 27, 20, 20, 0)); //SET WAKTU RTC
// }
//=====
}

void loop() {

```

```

//RTC=====
DateTime now = rtc.now();
Serial.print(now.hour(), DEC);
Serial.print(':');
Serial.print(now.minute(), DEC);
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.println();
delay(1000);
//=====

//PAGI=====
if (now.minute() >= 6 && now.minute() <= 6 ) {
  kecilcw();
  speedpwm = 255; //kecepatan pelontar
  analogWrite(pwm, speedpwm);
  digitalWrite(52, LOW);
  Serial.println("RELAY AKTIF");
  proses();
  cw();
  delay(500);
  ccw();
  delay(500);
  Stop();
  delay(300);
}

else if (now.minute() == 12 ) {
  kecilstop();
  digitalWrite(52, HIGH);
  Serial.println("RELAY OFF");
  delay(1000);
}

```

```

digitalWrite(49, HIGH);
Stop();
delay(100);
}

//SIANG=====
if (now.minute() >= 13 && now.minute() <= 13 ) {

    kecilcw();
    speedpwm = 255; //kecepatan pelontar
    analogWrite(pwm, speedpwm);
    digitalWrite(52, LOW);
    Serial.println("RELAY AKTIF");
    proses();
    cw();
    delay(500);
    ccw();
    delay(500);
    Stop();
    delay(300);

}
else if (now.minute() == 14) {

    kecilstop();
    digitalWrite(52, HIGH);
    Serial.println("RELAY OFF");
    delay(1000);
    digitalWrite(51, HIGH);
    Stop();
}

```

```

    delay(100);
}

//SORE=====
if (now.minute() >= 15 && now.minute() <= 15 ) {

    kecilcw();
    speedpwm = 255; //kecepatan pelontar
    analogWrite(pwm, speedpwm);
    digitalWrite(52, LOW);
    Serial.println("RELAY AKTIF");
    proses();
    cw();
    delay(500);
    ccw();
    delay(500);
    Stop();
    delay(300);

}

else if (now.minute() == 16) {

    kecilstop();
    digitalWrite(52, HIGH);
    Serial.println("RELAY OFF");
    delay(1000);
    digitalWrite(53, HIGH);
    Stop();
    delay(100);
}

```

```
if (now.minute() == 17) {
```

```
    kecilstop();
```

```
    digitalWrite(51, LOW);
```

```
    digitalWrite(53, LOW);
```

```
    digitalWrite(49, LOW);
```

```
    digitalWrite(47, LOW);
```

```
    digitalWrite(52, HIGH);
```

```
    Serial.println("RELAY OFF");
```

```
    delay(1000);
```

```
    Stop();
```

```
    delay(100);
```

```
}
```

```
}
```

```
//=====
```

```
//PUTARAN MOTOR
```

```
void cw() {
```

```
    // speedpwm =70; //kecepatan pelontar
```

```
    // analogWrite(pwm, speedpwm);
```

```
    digitalWrite(m1, HIGH);
```

```
    digitalWrite(m2, LOW);
```

```
    digitalWrite(men1, HIGH);
```

```
    digitalWrite(men2, HIGH);
```

```
    Serial.print("cw");
```

```
}
```

```
void ccw() {
```

```
    // speedpwm =70; //kecepatan pelontar
```

```

// analogWrite(pwm, speedpwm);
digitalWrite(m1, LOW);
digitalWrite(m2, HIGH);
digitalWrite(men1, HIGH);
digitalWrite(men2, HIGH);
Serial.print("ccw");
}
void Stop() {
// speedpwm =0; //kecepatan pelontar
// analogWrite(pwm, speedpwm);
digitalWrite(m1, LOW);
digitalWrite(m2, LOW);
digitalWrite(men1, LOW);
digitalWrite(men2, LOW);
Serial.print("stop");
}
//=====
//LAMPU PROSES
void proses() {
digitalWrite(47, HIGH);
delay(100);
digitalWrite(47, LOW);
delay(100);
}

void kecilcw()
{
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
analogWrite(in1, 100);
Serial.print("cwkecil");
}

```

```
}
```

```
void kecilstop()
```

```
{
```

```
  analogWrite(in1, 0);
```

```
  analogWrite(in2, 0);
```

```
}
```

LAMPIRAN 4 : Program NodeMCU

```
#include <ESP8266WiFi.h>
#define TRIGGER_PIN  D7
#define ECHO_PIN     D8
#include <Wire.h>
#include "RTClib.h"

RTC_DS3231 rtc;

char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"};

const char* ssid    = "WIFI PRIBADI";
const char* password = "niatkanduluaja2";

//const char* ssid    = "JENGKOL";
//const char* password = "ntahkudaktau";

const char* host = "192.168.43.140";
//const char* host = "192.168.1.7";

WiFiClient client;
const int httpPort = 80;
String url;
long duration, p;
float distance;
unsigned long timeout;

void setup() {
  Serial.begin(9600);
  delay(10);
```

```
if (! rtc.begin()) {
  Serial.println("Couldn't find RTC");
  while (1);
}

if (rtc.lostPower()) {
  Serial.println("RTC lost power, lets set the time!");
  // following line sets the RTC to the date & time this sketch was compiled
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  // This line sets the RTC with an explicit date & time, for example to set
  // January 21, 2014 at 3am you would call:
  rtc.adjust(DateTime(2020, 17, 8, 3, 15, 0));
}

pinMode(TRIGGER_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);

// We start by connecting to a WiFi network
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);

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

```

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

}

void baca_jarak() {
  digitalWrite(TRIGGER_PIN, LOW); // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(TRIGGER_PIN, LOW);
  duration = pulseIn(ECHO_PIN, HIGH);
  distance = (duration / 2) / 29.1;
  if (distance <= 7) {
    p = 100;
  }
  else if (distance >= 45) {
    p = 0;
  }
  else {
    p = (45 - distance) / (45 - 7) * 100;
  }
  Serial.print(p);
  Serial.println(" %");
}

void loop() {
  DateTime now = rtc.now();

```

```

Serial.print(" ");
Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
Serial.println(" ");

Serial.print(now.hour(), DEC);
Serial.print(':');
Serial.print(now.minute(), DEC);
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.println();
delay(2000);

if (now.minute() >= 23 && now.minute() <= 23 ) {
  makanpagi();
  delay(2000);
}
else if (now.minute() >= 24 && now.minute() <=24 ) {
  makansiang();
  delay(2000);
}

else if (now.minute() >= 25 && now.minute() <= 25) {
  makansore();
  delay(2000);
}

}

//MAKAN PAGI
void makanpagi() {
  Serial.print("baca jarak ");

```

```

baca_jarak();

Serial.print("connecting to ");
Serial.println(host);

if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    //return;
}

// We now create a URI for the request
url = "/proyekakhir/admin/insertdatapakan.php/datapakan?sisas=";
url += p;

Serial.print("Requesting URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
timeout = millis();
while (client.available() == 0) {
    if (millis() - timeout > 5000) {
        Serial.println(">>> Client Timeout !");
        client.stop();
        return;
    }
}

// Read all the lines of the reply from server and print them to Serial

```

```
while (client.available()) {  
    String line = client.readStringUntil('\r');  
    Serial.print(line);  
}
```

```
Serial.println();  
Serial.println("closing connection");  
Serial.println();  
delay(1000);  
}
```

```
//MAKAN SIANG
```

```
void makansiang() {  
    Serial.print("baca jarak ");  
    baca_jarak();  
  
    Serial.print("connecting to ");  
    Serial.println(host);  
  
    if (!client.connect(host, httpPort)) {  
        Serial.println("connection failed");  
        //return;  
    }  
}
```

```
// We now create a URI for the request
```

```
url = "/proyekakhir/admin/insertdatapakan1.php/datapakan?sis=";  
url += p;
```

```
Serial.print("Requesting URL: ");  
Serial.println(url);
```

```

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
    "Host: " + host + "\r\n" +
    "Connection: close\r\n\r\n");
timeout = millis();
while (client.available() == 0) {
    if (millis() - timeout > 5000) {
        Serial.println(">>> Client Timeout !");
        client.stop();
        return;
    }
}

// Read all the lines of the reply from server and print them to Serial
while (client.available()) {
    String line = client.readStringUntil('\r');
    Serial.print(line);
}

Serial.println();
Serial.println("closing connection");
Serial.println();
delay(1000);
}
//MAKAN SORE
void makansore() {
    Serial.print("baca jarak ");
    baca_jarak();

    Serial.print("connecting to ");

```

```

Serial.println(host);

if (!client.connect(host, httpPort)) {
  Serial.println("connection failed");
  //return;
}

// We now create a URI for the request
url = "/proyekakhir/admin/insertdatapakan2.php/datapakan?sis=";
url += p;

Serial.print("Requesting URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
  "Host: " + host + "\r\n" +
  "Connection: close\r\n\r\n");
timeout = millis();
while (client.available() == 0) {
  if (millis() - timeout > 5000) {
    Serial.println(">>> Client Timeout !");
    client.stop();
    return;
  }
}

// Read all the lines of the reply from server and print them to Serial
while (client.available()) {
  String line = client.readStringUntil('\r');
  Serial.print(line);
}

```

```
}
```

```
Serial.println();
```

```
Serial.println("closing connection");
```

```
Serial.println();
```

```
delay(1000);
```

```
}
```