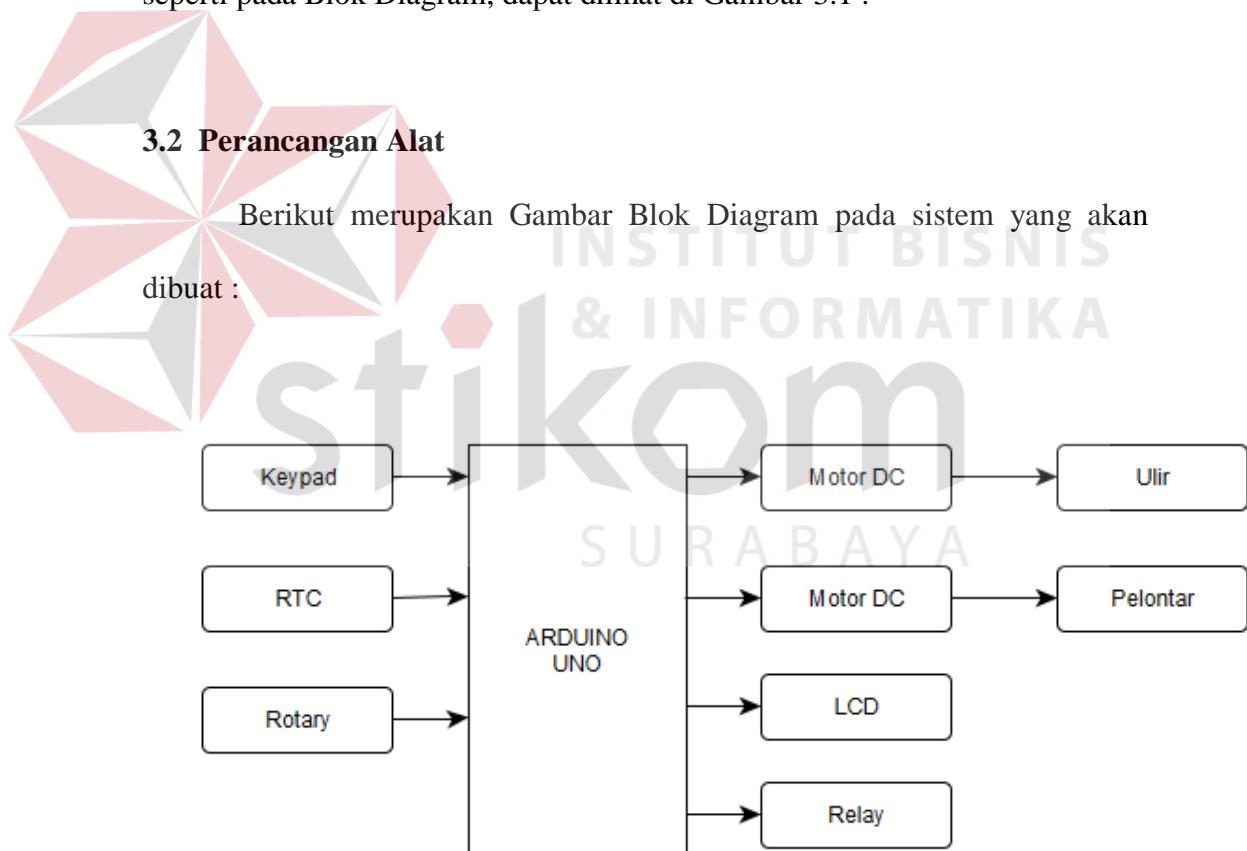


BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Pada Bab ini akan menjelaskan tentang perancangan sistem alat yang dibuat. Alat yang dibuat tersebut adalah sebuah alat pemberi pakan ikan secara otomatis pada kolam ikan berbasis *Arduino*. Penjelasan perancangan alat seperti pada Blok Diagram, dapat dilihat di Gambar 3.1 .



Gambar 3.1 Blok diagram

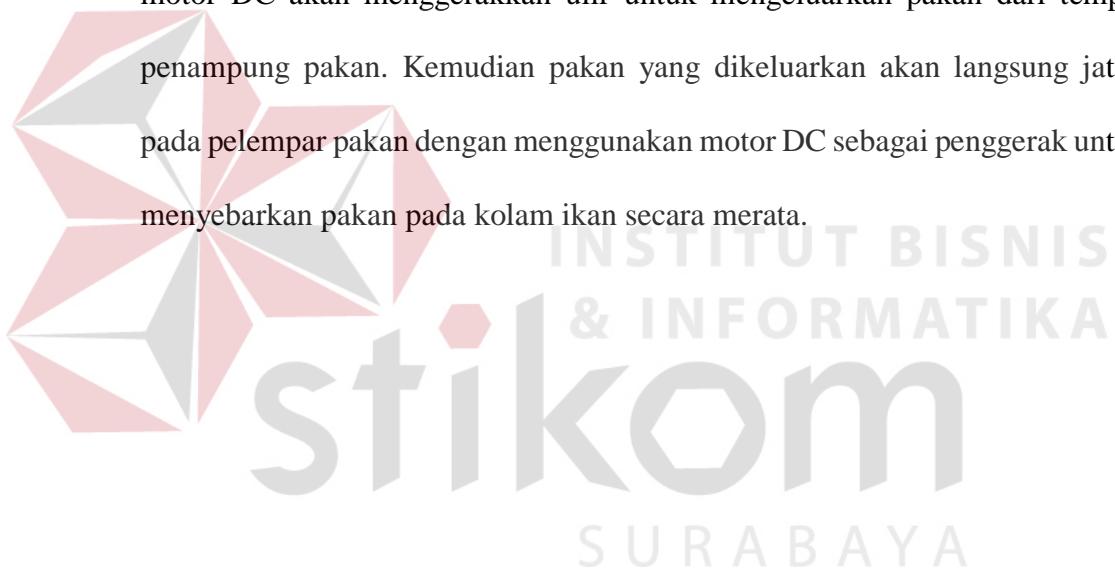
Dari Gambar 3.1 dijelaskan beberapa *input* dan *output* yang digunakan antara lain, *Keypad*, RTC (*Real Time Clock*), *Rotary*, dan Potensiometer sebagai *input*. Alat ini memiliki beberapa *actuator* yaitu dua buah motor DC yang dikendalikan oleh *Arduino*. Data yang diperoleh dari *input* akan diproses oleh *Arduino* lalu ditampilkan di sebuah LCD 20x4 berupa jumlah berat pakan, waktu penjadwalan pakan, dan jumlah takaran pakan, serta motor DC akan bergerak sesuai perintah *Arduino*.

Pada Gambar 3.1 Blok diagram proses dari alat ini berawal mula pada *input* yang dilakukan oleh *user* menggunakan *keypad* yang tersedia. *Inputan* yang dilakukan ada tiga bagian, yaitu *input* penjadwalan pemberian pakan, *input* jumlah takaran ikan dan *input* jumlah berat pakan yang dimasukkan pada tempat penampung pakan. Untuk *input* penjadwalan menggunakan *inputan* berformat (hh : mm) atau jam : menit. Untuk *input* takaran pakan ikan yang ingin dikeluarkan dan berat pakan ikan yang dimasukkan pada tempat penampung pakan menggunakan satuan gram. Setelah *user* menginputkan data pada *keypad*, data akan diolah dan disimpan menggunakan *mikrokontroller* *Arduino Mega 2560* sebagai *controller*.

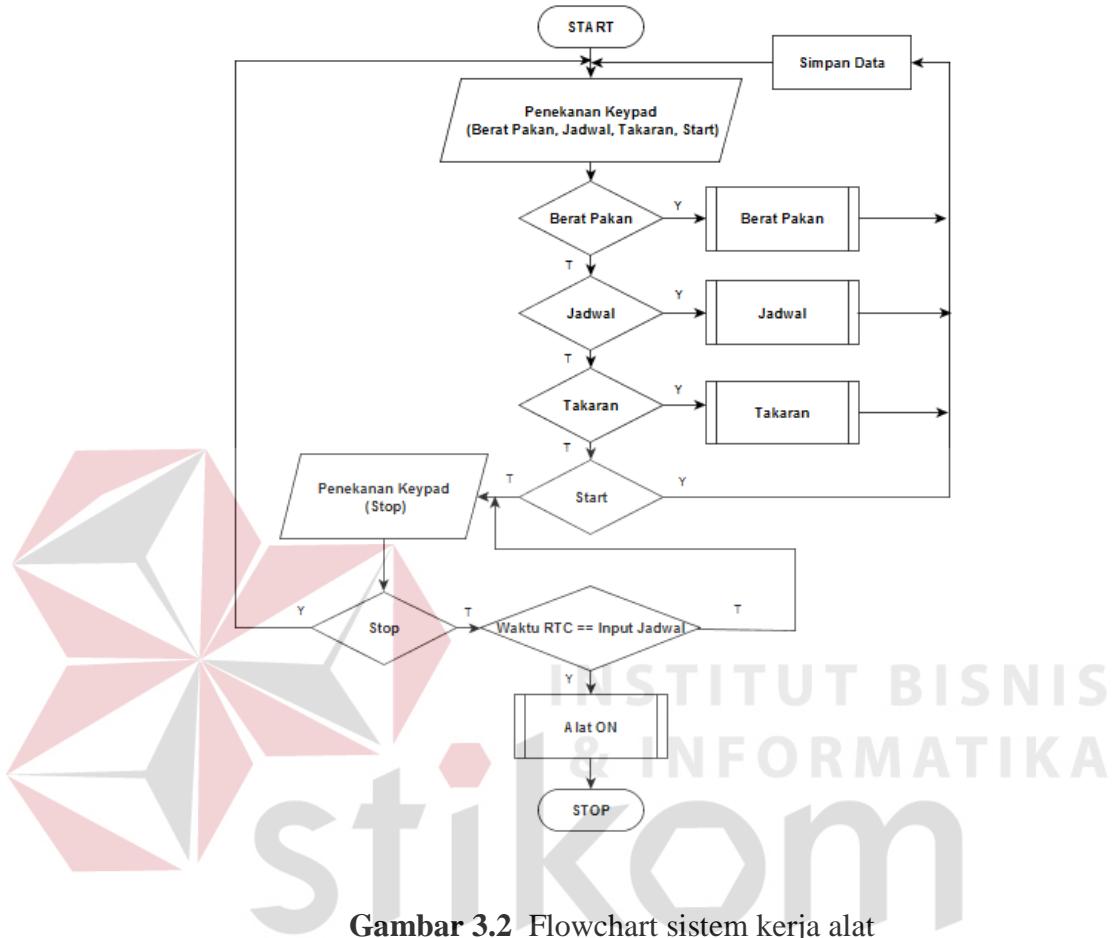
Secara keseluruhan alat dapat dibagi menjadi tiga poin utama, yaitu takaran pakan yang ingin dikeluarkan, penjadwalan pemberian pakan dan penyebaran pakan pada kolam. Pada poin penentu jumlah takaran pakan yang ingin dikeluarkan menggunakan sensor *Rotary* yang akan membaca putaran motor DC pada ulir. Jumlah takaran pakan ikan yang dikeluarkan berdasarkan berapa lama dan kecepatan putaran motor DC pada ulir. Setelah data sensor

Rotary didapatkan, data tersebut akan diolah menggunakan *mikrokontroller* dan dibandingkan dengan *input* jumlah takaran yang telah dilakukan oleh *user*.

Pada poin penjadwalan pemberian pakan dan penyebar pakan pada kolam, *mikrokontroller* akan menjalankan alat sesuai dengan jadwal pemberian pakan yang telah diinputkan oleh *user* yang kemudian akan dibandingkan dengan RTC pada *Arduino*. Ketika sudah memasuki waktu penjadwalan pemberian pakan, maka *mikrokontroller* akan memproses data dan selanjutnya motor DC akan menggerakkan ulir untuk mengeluarkan pakan dari tempat penampung pakan. Kemudian pakan yang dikeluarkan akan langsung jatuh pada pelempar pakan dengan menggunakan motor DC sebagai penggerak untuk menyebarluaskan pakan pada kolam ikan secara merata.



3.3 Perancangan Flowchart

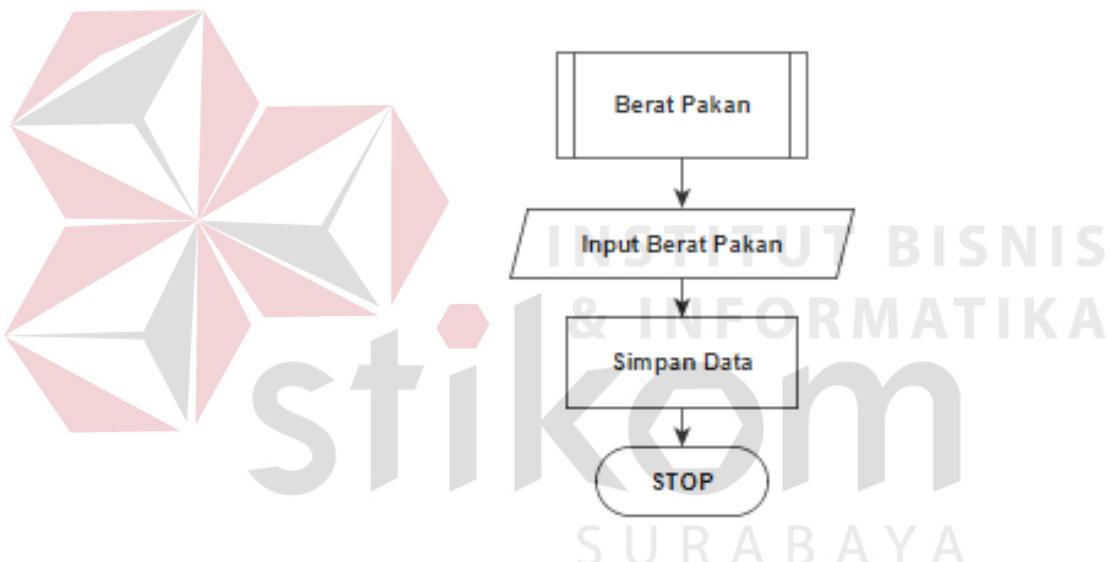


Gambar 3.2 Flowchart sistem kerja alat

Pada Gambar 3.2 *Flowchart* sistem kerja alat adalah alur cara kerja dari keseluruhan alat. Sebelum proses ini berjalan terdapat empat menu yang akan dipilih oleh *user* yaitu Proses berat pakan , jadwal dan takaran .Proses ini bermula proses dari alat ini berawal mula pada *input* yang dilakukan oleh *user* menggunakan *keypad* pada menu yang telah tersedia. Selanjutnya proses ini akan terbagi menjadi beberapa *sub flowchart*.

3.3.1 Sub Flowchart Berat Pakan

Pada Gambar 3.3 adalah *Sub flowchart* berat pakan yang digunakan untuk *menginputkan* jumlah berat pakan pada penampung. Pada proses *penginputan* ini menggunakan satuan berat gram. Setelah *user* *menginputkan* data, maka data berat pakan tersebut akan tersimpan dan diproses oleh *Arduino*. Untuk lebih detailnya bisa dilihat pada Gambar 3.3 .

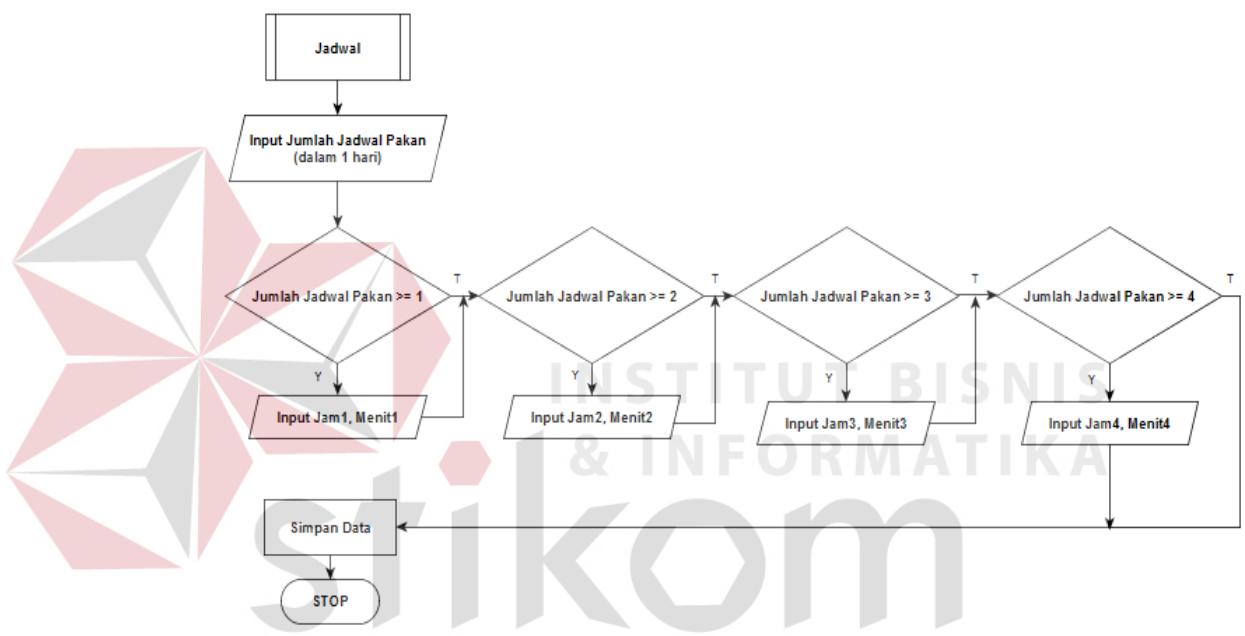


Gambar 3.3 *Sub flowchart* berat pakan

3.3.2 Sub Flowchart Jadwal

Pada Gambar 3.4 *Sub flowchart* jadwal adalah proses untuk menginputkan jumlah jadwal pemberian pakan dalam satu hari dan pada pukul berapa saja penjadwalan itu dilakukan. Pada proses *penginputan* ini menggunakan satuan jam dan menit. Jika *user* *menginputkan* jumlah

jadwal pakan itu sebanyak satu kali dalam sehari, maka *user* hanya menginputkan jam dan menit sebanyak satu kali berlaku untuk seterusnya. Setelah *user* menginputkan data, maka data jadwal tersebut akan tersimpan dan diproses oleh *Arduino*. Untuk lebih detailnya bisa dilihat pada Gambar 3.4.

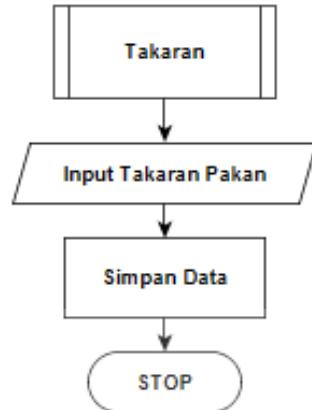


Gambar 3.4 Sub flowchart jadwal

3.3.3 Sub Flowchart Takaran

Pada Gambar 3.5 Sub flowchart takaran menjelaskan proses untuk *menginputkan* jumlah berat takaran pakan yang dikeluarkan setiap satu kali penjadwalan pemberian pakan ikan. Pada proses *penginputan* ini menggunakan satuan berat gram. Setelah *user* menginputkan data, maka data takaran tersebut akan tersimpan dan

diproses oleh *Arduino* untuk lebih detailnya bisa dilihat pada Gambar 3.5.



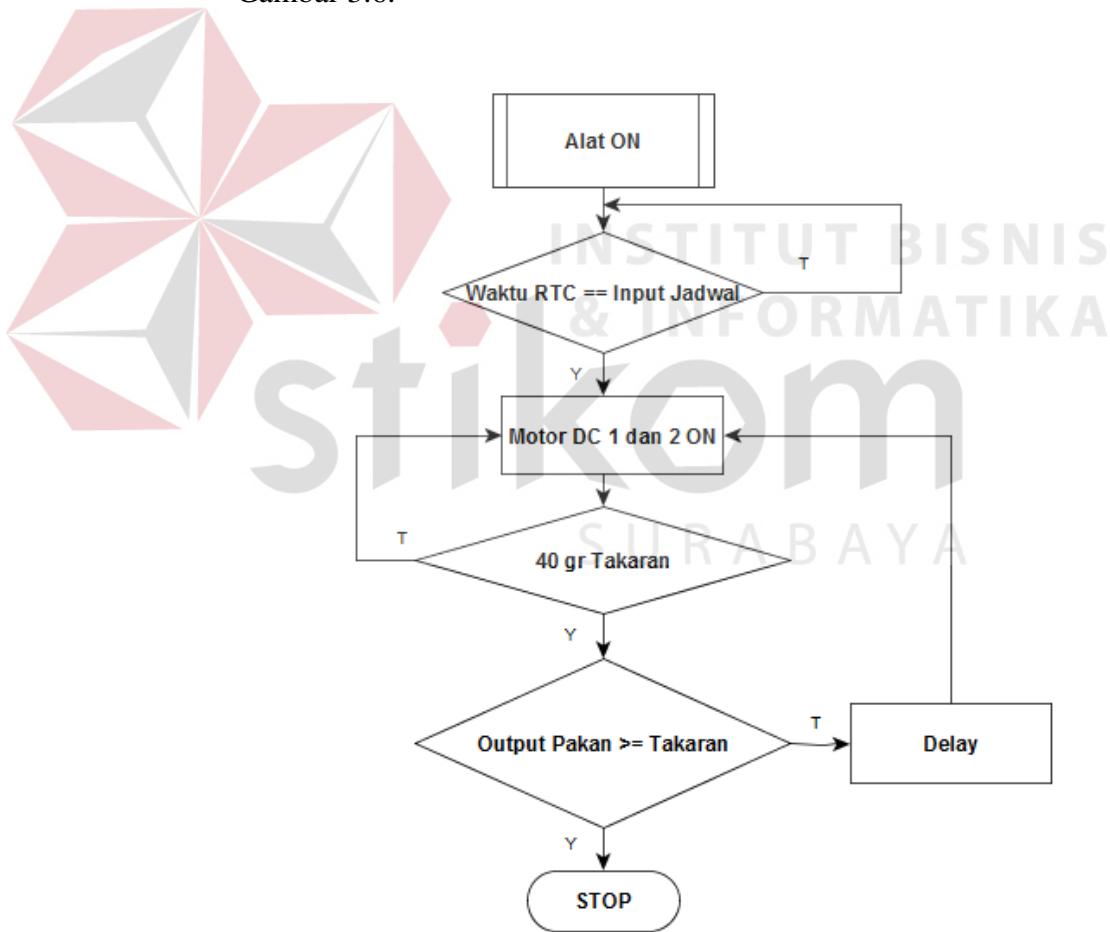
Gambar 3.5 Sub flowchart takaran

3.3.4 Sub Flowchart Alat ON

Pada sub flowchart alat on ini menjelaskan proses bekerjanya alat sesuai jadwal yang telah ditentukan oleh user. Proses ini berawal dari waktu RTC dan waktu input yang menunjukkan jam dan menit yang sama. Setelah itu motor DC 2 otomatis akan menyala terlebih dahulu lalu diikuti oleh menyalanya motor DC 1. Motor DC 1 berfungsi untuk menggerakkan ulir yang berisi pakan kemudian pakan akan jatuh ke bawah tepat ketika motor DC 2 (Pelontar Pakan) berputar, otomatis pakan akan terlontar pada kolam. Ketika motor DC 1 dan motor DC 2 menyala maka akan ditinjau kembali apakah motor DC 1 berputar sebanyak 100 kali, jika benar maka motor DC 1 akan berhenti berputar

dengan delay tertentu, jika tidak, maka motor DC 1 akan berputar kembali hingga memenuhi 100 kali putaran. Setelah memenuhi 100 kali putaran, maka akan ditinjau apakah takaran pakan yang dikeluarkan apakah sudah sesuai dengan apa yang diinputkan oleh user, jika benar maka motor DC 2 akan berhenti yang menandakan pemberian pakan telah selesai, jika tidak maka, motor DC 1 akan kembali berputar hingga jumlah takaran pakan terpenuhi untuk lebih detailnya bisa dilihat pada

Gambar 3.6.



Gambar 3.6 Sub *flowchart* alat on

3.4 Penginputan dan Penyimpanan Data

Alat ini memberikan 5 menu untuk *user*, yang pertama adalah *penginputan* berat pakan yang dimasukkan pada penampung pakan. Pada *inputan* ini menggunakan satuan berat gram. Proses ini berguna untuk memeriksa ketersediaan pakan dalam penampung pakan. Yang kedua adalah *penginputan* jadwal pemberian pakan ikan. Pada *inputan* ini, dibagi menjadi dua bagian yaitu menentukan jumlah jadwal pemberian pakan ikan dalam satu hari dan pada jam berapa saja jadwal pemberian pakan itu dilakukan. Untuk jumlah pemberian pakan per hari, memiliki batasan sebanyak empat kali dalam satu hari. Untuk penjadwalan pemberian pakan menggunakan format jam, menit dan detik. Yang ketiga adalah *penginputan* jumlah takaran pakan ikan pada setiap kali penjadwalan dengan menggunakan satuan berat gram. Yang keempat adalah menu start yang digunakan untuk memulai menjalankan program. Dan yang kelima adalah menu lihat jadwal yang digunakan untuk melihat jadwal yang telah diinputkan. Untuk tampilan menu pada LCD dapat dilihat pada Gambar 3.7.



Gambar 3.7 Tampilan menu pada LCD 20x4

Misal :

User menginputkan jumlah berat pakan pada penampung pakan dengan nilai sebesar 500, maka program akan membacanya sebanyak 500 gram, jika *user menginputkan* jumlah berat pakan pada penampung pakan dengan nilai sebesar 4000, maka program akan membacanya sebanyak 4000 gram.

Setelah *user menginputkan* jumlah berat pakan pada penampung, maka selanjutnya adalah *menginputkan* berapa banyak jadwal pemberian pakan dalam satu hari, jika *user menginputkan* nilai sebesar 2, maka program akan membacanya jadwal pemberian pakan dalam satu hari sebanyak dua kali. Setelah itu *user menginputkan* 07.00.00 untuk penjawalan pakan pertama dan 16.00.00 untuk jadwal pakan kedua, maka program akan membacanya jadwal pertama akan berjalan pada jam 07.00 pagi dan jadwal kedua pada jam 16.00 sore.

Selanjutnya jika *user menginputkan* jumlah takaran pakan sebanyak 1000 pada setiap penjadwalan pemberian pakan, maka jumlah takaran pada setiap penjadwalan pemberian pakan adalah 1000 gram.

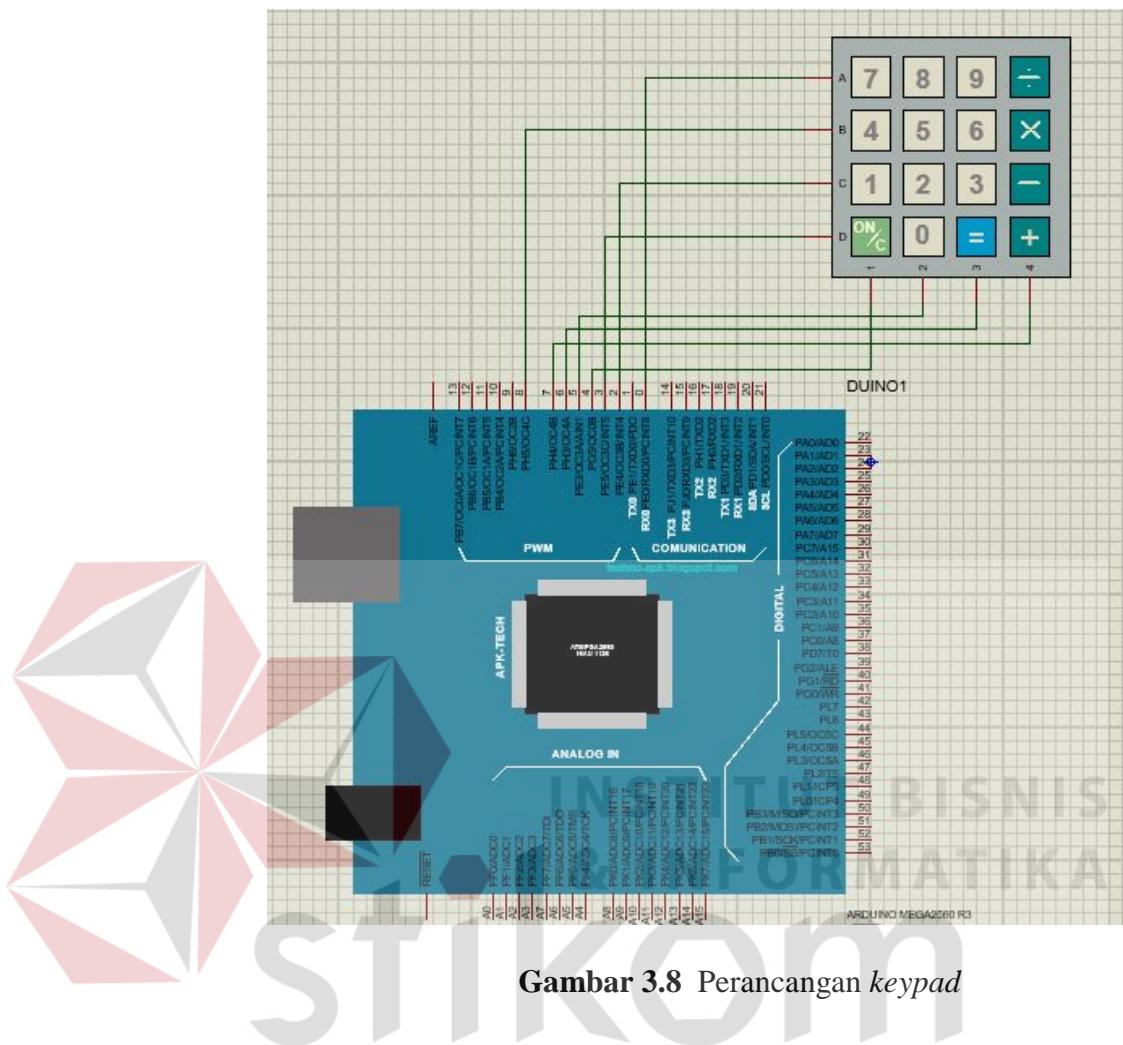
3.5 Perancangan *Hardware*

Setelah melakukan perancangan sistem, penulis akan melakukan perancangan *hardware* sesuai dengan perancangan sistem yang telah dibuat agar sistem dapat terhubung dan terintegrasi dengan baik.

3.5.1 Perancangan *Keypad*

Keypad yang digunakan adalah *keypad* matriks 4 x 4. *Keypad* matriks 4 x 4 terdiri dari sejumlah saklar, yang terhubung sebagai baris dan kolom dengan susunan seperti yang ditunjukkan pada Gambar 3.8. Agar *Arduino* dapat melakukan *scan keypad*, maka *port* mengeluarkan salah satu bit dari 4 bit yang terhubung pada kolom dengan logika *low “0”* dan selanjutnya membaca 4 bit pada baris untuk menguji jika ada tombol yang ditekan pada kolom tersebut.

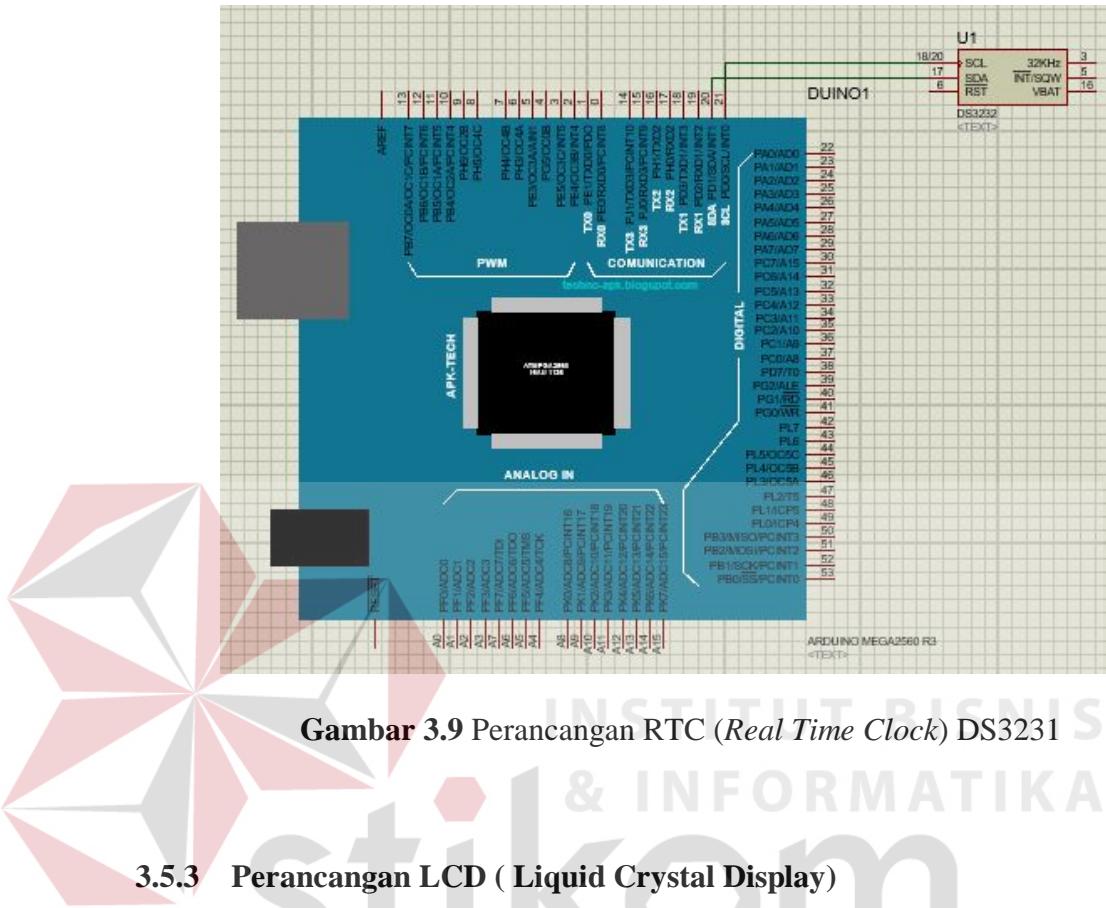
Keypad berguna untuk memilih menu yang ditampilkan oleh LCD dan *menginputkan* sesuai yang diinginkan oleh *user*. *Inputan* itu meliputi jumlah berat pakan ikan pada penampung, penjadwalan pemberian pakan ikan, jumlah takaran pakan ikan serta menu *start* yang digunakan untuk memulai program.



Gambar 3.8 Perancangan keypad

3.5.2 Perancangan RTC (*Real Time Clock*)

RTC (*Real Time Clock*) yang digunakan adalah RTC (*Real Time Clock*) tipe DS3231. RTC (*Real Time Clock*) DS3231 menyediakan sebuah *clock* referensi yang stabil dan akurat. RTC (*Real Time Clock*) DS3231 menyediakan waktu dan kalender dengan waktu alarm yang telah di atur dalam satu hari . RTC (*Real Time Clock*) DS3231 berguna sebagai acuan waktu *real* dengan waktu yang telah *diinputkan* oleh *user*. Untuk model perancangan RTC (*Real Time Clock*) DS3231 dapat dilihat pada Gambar 3.9.

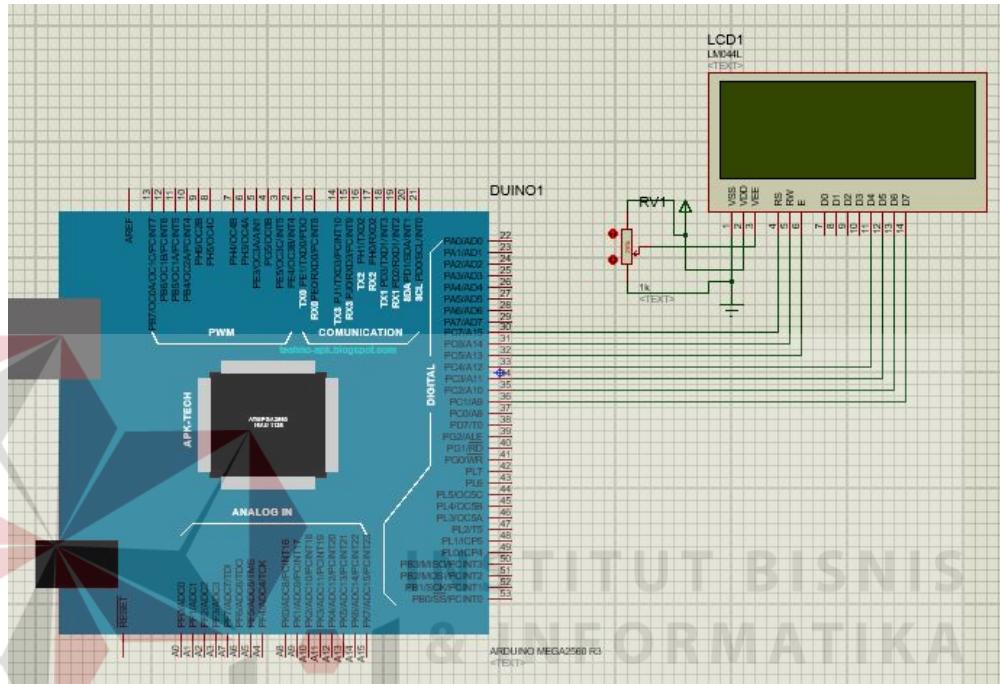


3.5.3 Perancangan LCD (Liquid Crystal Display)

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. LCD (*Liquid Crystal Display*) bisa memunculkan Gambar atau tulisan dikarenakan terdapat banyak sekali titik cahaya (piksel) yang terdiri dari satu buah kristal cair sebagai sebuah titik cahaya.

LCD (*Liquid Crystal Display*) yang akan digunakan adalah LCD (*Liquid Crystal Display*) berukuran 20 x 4. Salah satu alasan mengapa memakai LCD (*Liquid Crystal Display*) 20x4 ini adalah untuk memudahkan

dalam penampilan menu yang tersedia. Untuk model perancangan RTC (*Real Time Clock*) DS3231 dapat dilihat pada Gambar 3.10.

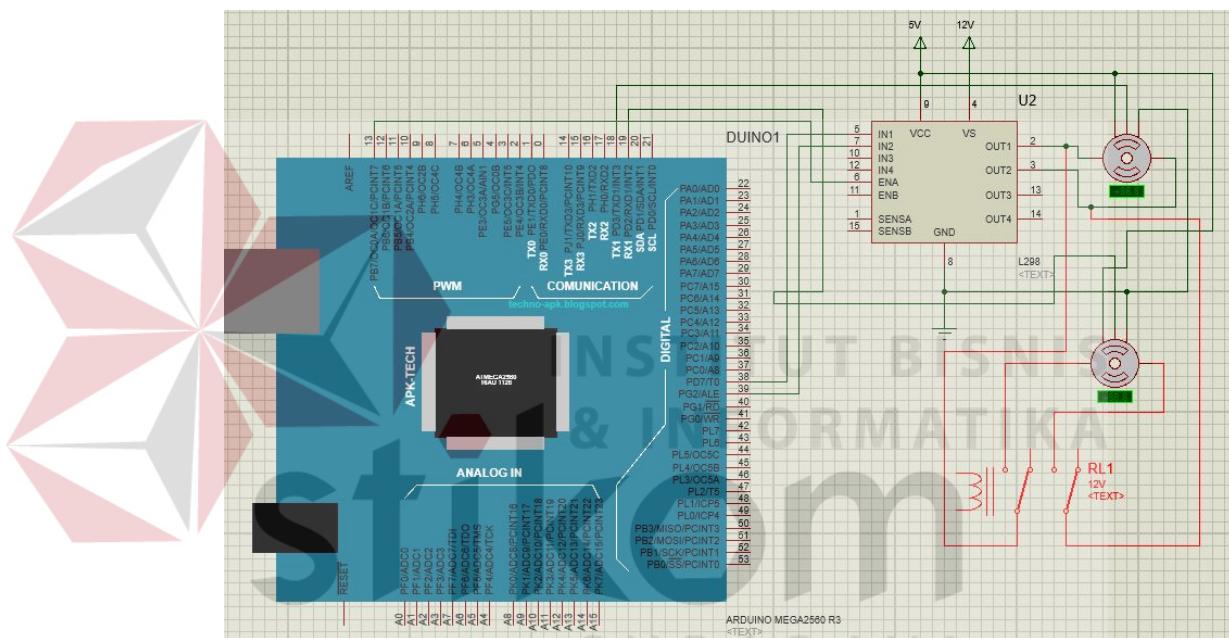


Gambar 3.10 Perancangan LCD (*Liquid Crystal Display*)

3.5.4 Perancangan Motor Driver

Pada Projek ini menggunakan motor *driver* tipe EMS (*Embedded Module Series*) 2 A Dual H-Bridge merupakan *driver* H-Bridge yang didesain untuk menghasilkan *drive* dua arah dengan arus kontinu hingga 2 A pada tegangan 4,8 Volt sampai 46 Volt. Modul ini dapat men-*drive* beban-beban induktif seperti misalnya motor DC, motor *stepper*, koil relay, selenoida, dan beban-beban lainnya.

Pada EMS Dual H-Bridge terdapat tiga *Interface Header*, Yang pertama adalah *Interface Header 1* berfungsi untuk mengatur motor DC pada ulir, pada *Interface Header 2* berfungsi untuk mengatur motor DC pada pelontar pakan dan pada *Interface Header 3* berfungsi sebagai *Power* dan konektor pada motor DC. Untuk model perancangan Motor *Driver* dapat dilihat pada Gambar 3.11.



Gambar 3.11 Perancangan motor *driver*

3.6 Perancangan Software

Perancangan *software* sangatlah diperlukan untuk menentukan jalannya program sesuai perancangan *hardware* yang telah dilakukan. Perancangan *software* yang akan dilakukan meliputi perancangan *keypad*, perancangan

LCD (*Liquid Crystal Display*) dan perancangan RTC pada *Arduino* menggunakan aplikasi *Arduino IDE*.

3.6.1 Perancangan Program *Keypad*

Perancangan ini digunakan sebagai tombol dalam proses *input* yang dilakukan oleh *user*. Hal ini dilakukan agar memudahkan *user* dalam memilih menu dan *menginputkan* nilai yang telah ditampilkan pada LCD. Untuk tampilan program *keypad* dapat dilihat pada Gambar 3.12. .



```

sketch_jul17a:sketch_jul17a.ino

#include <LiquidCrystal.h>

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
//define the symbols on the buttons of the keypads
char Keys[ROWS][COLS] = {
{'1','4','7','*'},
{'2','5','8','0'},
{'3','6','9','#'},
{'A','B','C','D'}
};
byte rowPins[ROWS] = {3, 2, 8, 0}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {7, 6, 5, 4}; //connect to the column pinouts of the keypad
Keypad keypad = Keypad(makeKeymap(Keys),rowPins,colPins,ROWS,COLS);

void setup()
{
    char key = keypad();
}

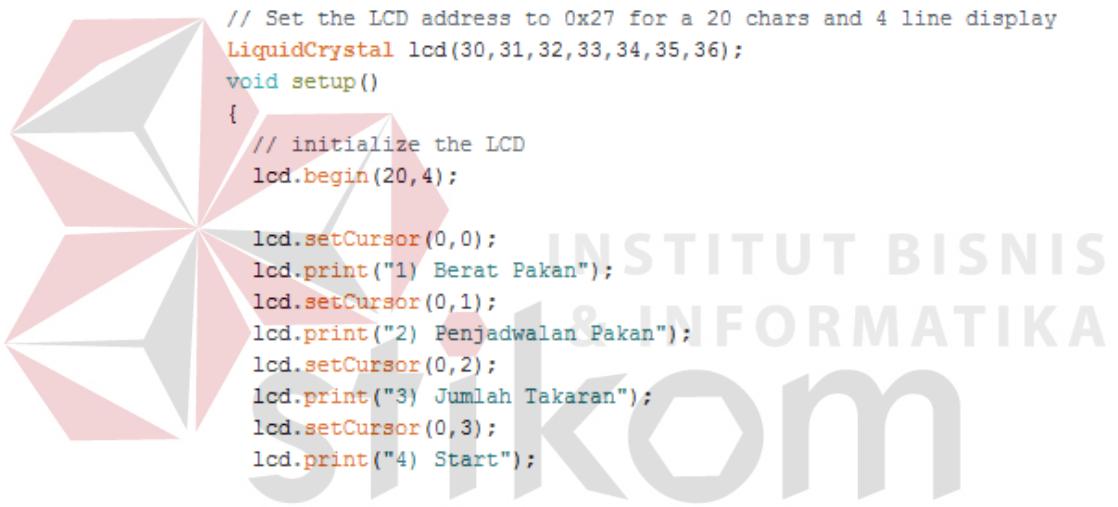
void loop()
{
}

```

Gambar 3.12 Tampilan program *keypad*

3.6.2 Perancangan Program LCD (*Liquid Crystal Display*)

Perancangan ini digunakan sebagai tampilan menu dalam proses *input* yang dilakukan oleh *user*. Hal ini dilakukan agar memudahkan *user* dalam memilih pilihan menu yang telah ditampilkan. Untuk tampilan program LCD (*Liquid Crystal Display*) dapat dilihat pada Gambar 3.13.



```
sketch_jul17a §
#include <LiquidCrystal.h>

// Set the LCD address to 0x27 for a 20 chars and 4 line display
LiquidCrystal lcd(30,31,32,33,34,35,36);
void setup()
{
    // initialize the LCD
    lcd.begin(20,4);

    lcd.setCursor(0,0);
    lcd.print("1) Berat Pakan");
    lcd.setCursor(0,1);
    lcd.print("2) Penjadwalan Pakan");
    lcd.setCursor(0,2);
    lcd.print("3) Jumlah Takaran");
    lcd.setCursor(0,3);
    lcd.print("4) Start");

}
void loop()
{
}
```

Gambar 3.13 Tampilan program LCD (*Liquid Crystal Display*)

3.6.3 Perancangan Program RTC (*Real Time Clock*)

Perancangan ini digunakan sebagai sumber data waktu yang akurat berupa data jam, hari, bulan maupun tahun. Hal ini dilakukan agar memudahkan alat untuk melakukan penjadwalan pemberian pakan secara tepat. Untuk tampilan program RTC (*Real Time Clock*) dapat dilihat pada

Gambar 3.14..

The logo of Institut Bisnis & Informatika Stikom Surabaya is visible in the background of the code editor window. It features a stylized geometric design composed of overlapping triangles in shades of pink, grey, and white. The text "INSTITUT BISNIS & INFORMATIKA" is stacked above the word "Stikom", which is written in a large, bold, sans-serif font. Below "Stikom" is the word "SURABAYA".

```
sketch_jul17a§
#include <DS3231.h>

// Init the DS3231 using the hardware interface
DS3231 rtc(SDA, SCL);
Time t;

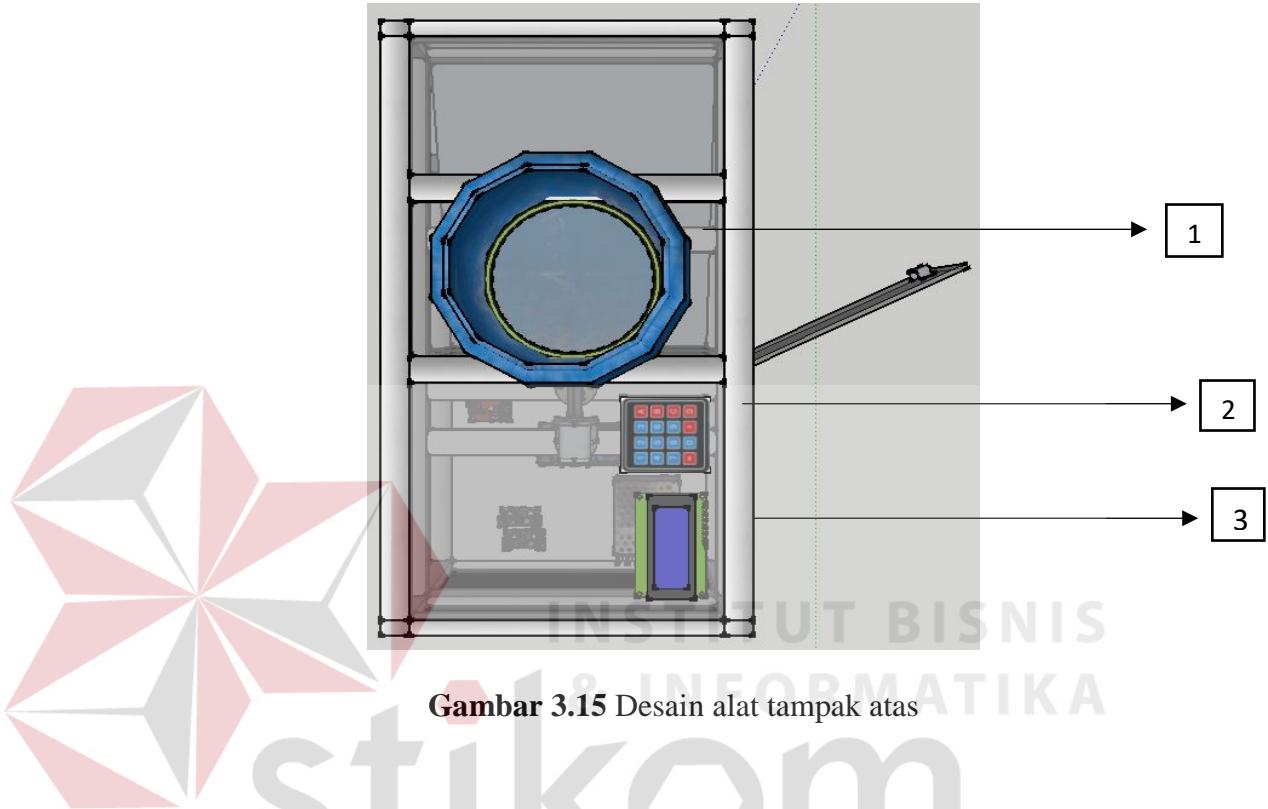
void setup()
{
    // Initialize the rtc object
    rtc.begin();

    t = rtc.getTime();
    delay(1000);
}

void loop()
{
```

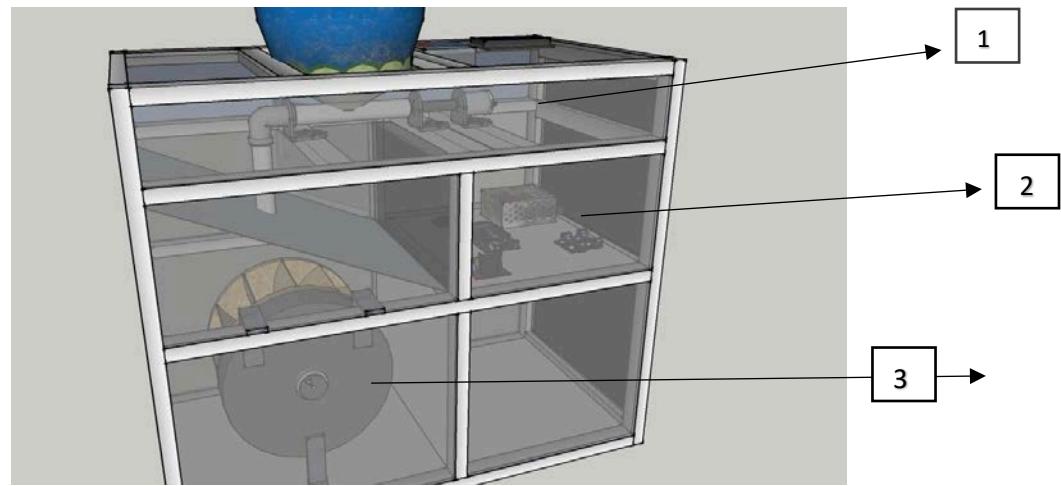
Gambar 3.14 Tampilan program RTC (*Real Time Clock*)

3.7 Desain Alat Pemberi Pakan Ikan Otomatis



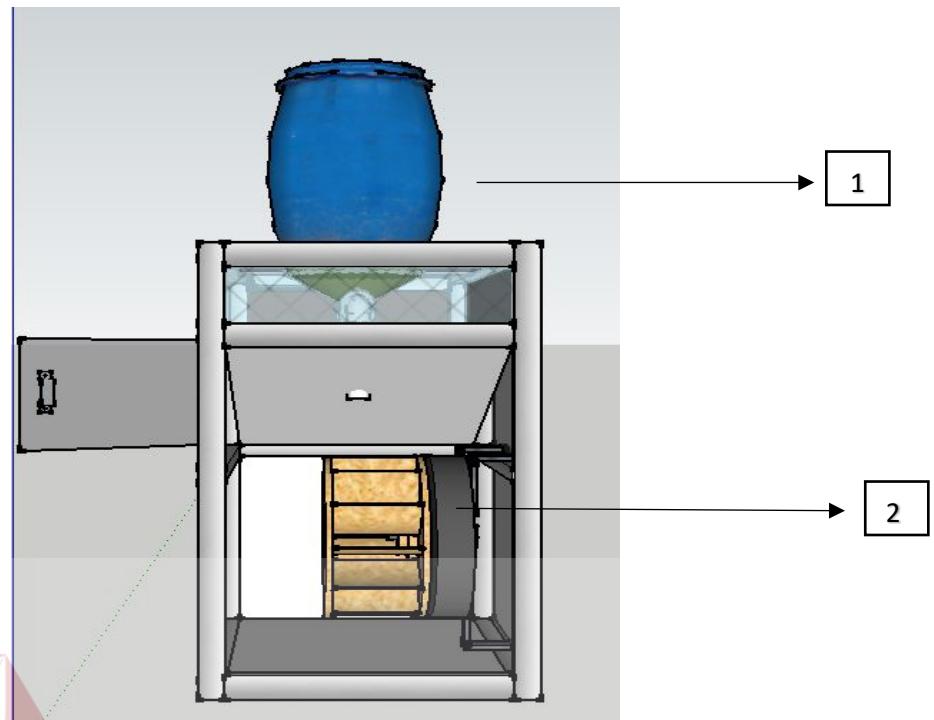
Gambar 3.15 Desain alat tampak atas

Pada Gambar 3.15 adalah desain alat yang dilihat dari atas. Pada kotak ke-1 adalah penampung pakan yang digunakan untuk menampung pakan ikan. Pada kotak ke-2 adalah *keypad* yang digunakan untuk *menginputkan* nilai. Pada kotak ke-3 adalah LCD yang digunakan sebagai *user interface*.



Gambar 3.16 Desain alat tampak samping

Pada Gambar 3.16 adalah desain alat tampak samping. Pada kotak ke-1 adalah motor DC yang telah terhubung dengan ulir (mata bor kayu jenis auger) yang digunakan untuk menyalurkan pakan dari penampung ke pelontar pakan. Pada kotak ke-2 adalah komponen-komponen elektronika yang digunakan pada alat ini. Pada kotak ke-3 adalah pelontar pakan berupa blower ac mobil yang digunakan untuk melontarkan pakan ke kolam.



Gambar 3.17 Desain alat tampak depan

Pada Gambar 3.17 adalah desain alat tampak depan. Pada kotak ke-1 adalah penampung pakan yang digunakan untuk menampung pakan ikan. Pada kotak ke-2 adalah pelontar pakan (yang dilihat dari depan) berupa blower ac mobil yang digunakan untuk melontarkan pakan ke kolam.