

BAB IV

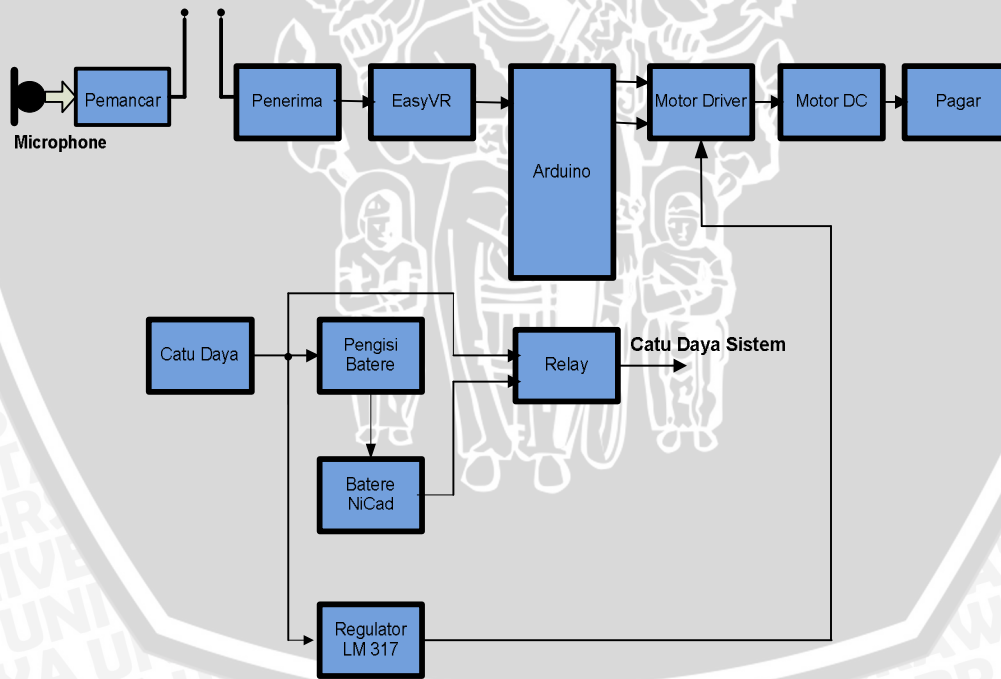
PERANCANGAN DAN PEMBUATAN ALAT

Perancangan prototipe pintu pagar ini dilakukan secara bertahap sehingga akan memudahkan dalam analisis pada setiap bloknnya maupun secara keseluruhan. Perancangan terdiri dari:

- Perancangan sistem.
- Perancangan perangkat keras.
- Perancangan sistem *Voice Recognition*.
- Perancangan perangkat lunak.

2.1 Diagram Blok Sistem

Secara garis besar, diagram blok perancangan *hardware* sistem secara keseluruhan ditunjukkan dalam Gambar 4.1



Gambar 4. 1. Diagram Blok Sistem

Fungsi masing-masing blok dalam diagram blok diatas adalah sebagai berikut :

1).Pemancar dan Penerima

Untuk menunjang kontrol nirkabel (*wireless control*) hubungan *microphone* dengan *EasyVR* direalisasi dengan menggunakan pemancar dan penerima FM.

2).Blok Modul *EasyVR*

Berfungsi sebagai pendeteksian suara dan percakapan. Masukan *password* yang berupa suara akan dideteksi dan dilakukan pencocokan dengan basis data suara dalam modul *EasyVR* , jika *match EasyVR* akan melakukan *encoding* input *password* dan mempresentasikannya dalam kode biner yang akan ditransfer ke Arduino melalui komunikasi serial. Data-data suara disampling pada modul *EasyVR* dengan menggunakan program *EasyVR Commander*. Data-data yang sudah disampling akan dimasukkan kembali ke basis data suara *EasyVR*.

3).Blok Mikrokontroler

Mikrokontroler yang digunakan adalah ATmega 328 yang berfungsi untuk mengolah data dari modul *EasyVR* , menjalankan proses *password decoding*, melakukan kendali motor DC dan memberikan *progress report* dari kinerja *EasyVR* berupa indikator LED .

4).Blok Motor *Driver*

Motor *driver* berfungsi untuk menguatkan arus keluaran dari Mikrokontroler sehingga dapat menggerakkan motor DC. Motor *driver* juga menyediakan konfigurasi *H-Bridge* sehingga Mikrokontroler dapat mengendalikan arah dan kecepatan melalui motor DC

5).Blok Motor DC

Motor DC berfungsi sebagai penggerak prototipe pintu pagar. *Manuver* pintu direncanakan dapat bergerak ke kiri, ke kanan dan berhenti dengan durasi dan kecepatan yang dapat diprogram (*programmable*).

6).Blok Catu Daya

Blok catu daya berfungsi memberikan suplai daya untuk Mikrokontroler dan rangkaian pendukungnya

7).Blok Pengisi Baterai

Berfungsi untuk mengisi baterai NiCad yaitu berupa rangkaian sumber arus tetap dengan arus keluaran konstan sebesar $1/10$ dari kapasitas baterai

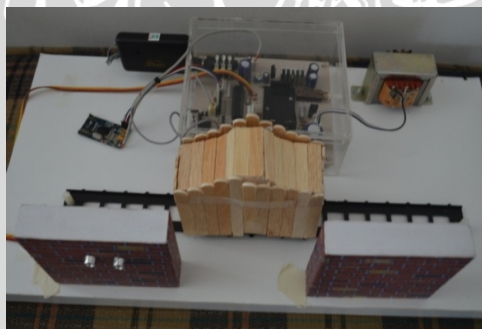
8).Blok Regulator LM 317

Berfungsi memberikan suplai tegangan sebesar tegangan 7,1 V pada motor DC

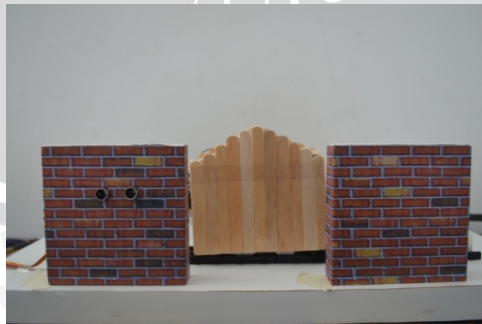
2.2 Perancangan Perangkat Keras

2.2.1 Perancangan Mekanik

Sistem mekanik dirancang sebagai pelindung, penopang komponen elektrik dan sebagai *prototipe* pintu pagar otomatis. Dalam perancangan ini digunakan multiplek dengan ketebalan 4 cm serta sterofoam sebagai pintu pagar geser. Untuk perancangan sistem mekanik ditunjukkan dalam Gambar 4.2 dan dalam Gambar 4.3.



Gambar 4.2. Perspektif Tampak Atas Prototipe Pintu Pagar

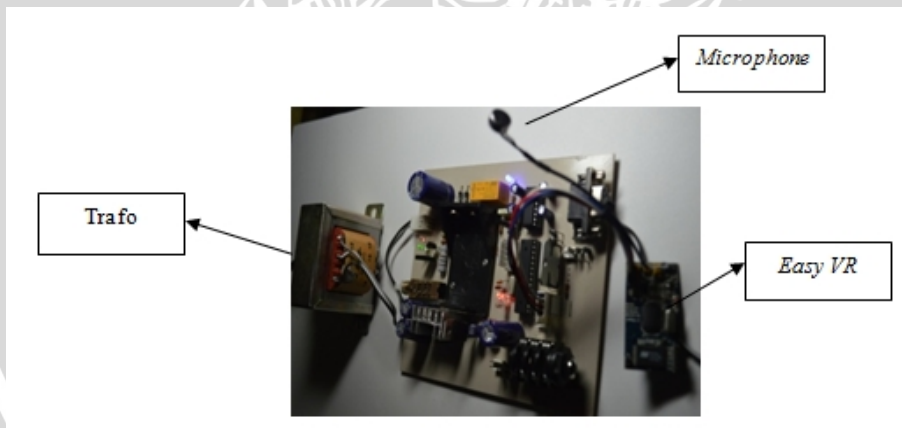


Gambar 4.3. Perspektif Tampak Depan Prototipe Pintu Pagar

Dalam perancangan ini pintu pagar akan bergerak menggeser ke kanan dan ke kiri untuk simulasi membuka dan menutup pintu pagar. Pergerakan pintu pagar akan dikendalikan oleh motor DC dan bergerak di atas rel.

2.2.2 Perancangan Sistem Elektronik

Diagram blok sistem elektronik, terdiri dari bagian catu daya, catu daya alternatif, sistem pengendali, dan juga sistem masukan dan keluaran. Pada bagian masukan berupa sebuah sensor *microphone* yang berfungsi untuk menerima masukan *password* berupa suara manusia dan modul *EasyVR* berfungsi sebagai pendeteksian suara dan percakapan. Untuk menunjang kontrol nirkabel (*wireless control*) hubungan *microphone* dengan *EasyVR* direalisasi dengan menggunakan pemancar dan penerima FM. Pada bagian kendali menggunakan *ArduinoUNO* sebagai Mikrokontroler. Pada bagian keluaran berupa *driver* motor dan motor DC untuk simulasi pintu pagar. Perangkat keras Sistem Keamanan Pintu Pagar Otomatis Menggunakan *Voice Recognition* ditunjukkan dalam Gambar 4.4.



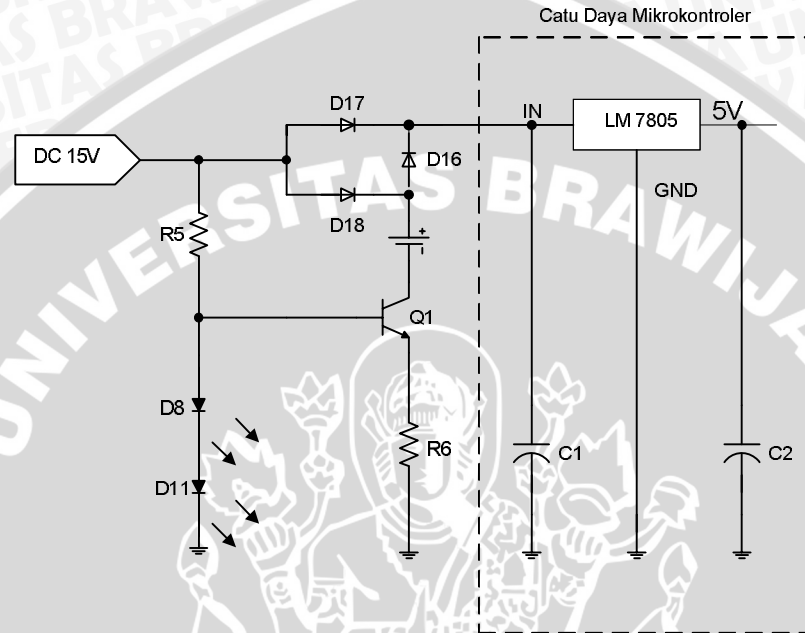
Gambar 4.4. Perangkat Keras Sistem Keamanan Pintu Pagar Otomatis Menggunakan *Voice Recognition*

2.2.2.1 Perancangan Rangkaian Pengisi Baterai

Sistem pagar otomatis ini menggunakan rangkaian catu daya sebesar 5V untuk rangkaian Mikrokontroler *ArduinoUNO*, board *EasyVR* dan *driver* motor L298.

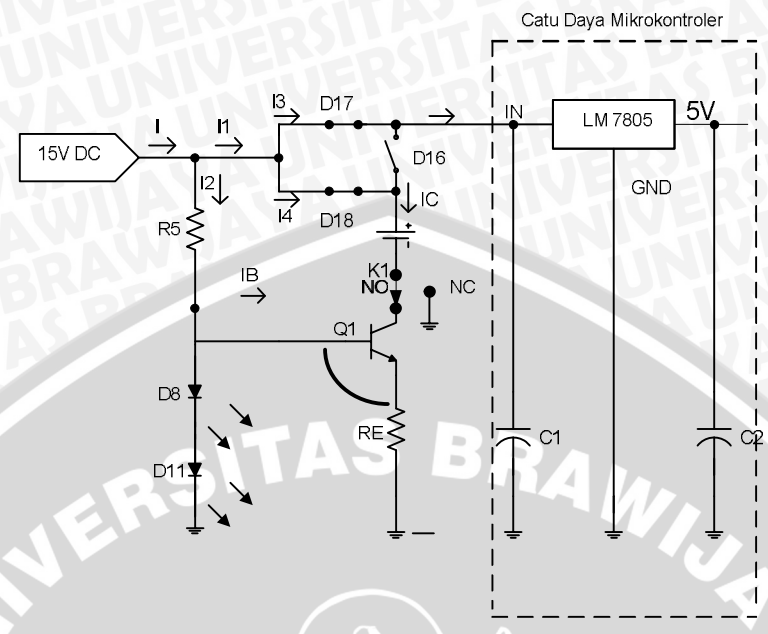
Rangkaian pengisi baterai digunakan untuk mengantisipasi apabila sistem tidak mendapatkan tegangan dari jaringan listrik, maka diharapkan

sistem akan tetap dapat berjalan dengan sumber tegangan dari baterai sebesar 9V. Dimana dengan kapasitas baterai tersebut diharapkan dapat menggantikan tegangan dari jaringan listrik dan sistem tetap dapat berjalan. Skema rangkaian pengisi baterai ditunjukkan dalam Gambar 4.5.



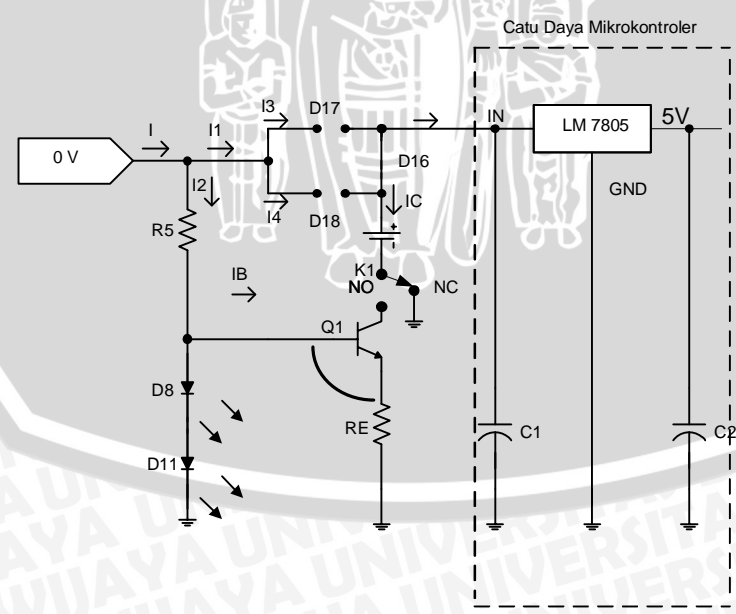
Gambar 4. 5. Rangkaian Pengisi Baterai

Analisa dari rangkaian baterai *charger* tersebut adalah sebagai berikut, pada saat rangkaian listrik aktif relay K_1 aktif, polaritas negatif naterai NiCad terhubung dengan kolektor Q1 dan dioda D17 dan D18 menghantar dan D16 tidak menghantar sehingga baterai Nicad tidak dapat memberikan suplai daya ke unit catu daya Mikrokontroller. Pada kondisi ini rangkaian catu daya Mikrokontroller (LM7805) mendapat suplai daya dari *input* DC 15V melalui dioda D17, melalui dioda D18 rangkaian rangkaian sumber arus konstan yang terdiri atas komponen R5, R6, D8, D11 dan Q1 mengisi baterai dengan arus tetap sebesar 1/10 dari kapasitas beteraai. Seperti ditunjukkan dalam Gambar 4.6



Gambar 4. 6. Rangkaian Ekuivalen Pengisi Baterai pada Saat Jaringan Listrik Aktif

Pada saat jaringan listrik padam ,rangkaian catu daya tidak aktif, relay K_1 tidak aktif, polaritas negatif naterai NiCad terhubung dengan *ground* dan dioda D_{17} dan D_{18} *off* dan D_{16} *ON*. Baterai NiCad akam memberikan suplai daya ke Mikrokontroler melalui D_{16} seperti ditunjukkan dalam Gambar 4.7



Gambar 4. 7. Rangkaian Ekuivalen Pengisi Baterai pada Saat Jaringan Listrik Padam

Langkah pertama perancangan rangkaian pengisi baterai adalah menentukan besarnya I_{LED} yang mengalir pada D_8 dan D_{11} dalam perencanaan ini ditentukan $I_{LED} = 30\text{mA}$ dari I_{LED} maksimum yang diijinkan 100mA

Dari DataSheet diketahui $V_{\text{led merah}} = 1,9\text{ V}$ dan $V_{\text{led hijau}} = 3,2\text{ V}$

Maka :

$$V_{R5} + V_{\text{led merah}} + V_{\text{led hijau}} - 15,6\text{V} = 0$$

$$V_{R5} = 15,6\text{V} - V_{\text{led merah}} - V_{\text{led hijau}}$$

$$V_{R5} = 10,5\text{V}$$

$$R_5 = V_{R5} / I_2$$

$$\text{Dimana } I_2 = I_{LED}$$

$$\text{Sehingga } R_5 = 10,5\text{V} / 30\text{mA}$$

$$R_5 = 350\Omega, \text{ dipasaran yang tersedia } 330\Omega$$

Jika I_E adalah arus pengisian yang besarnya ditentukan $1/10$ kapasitas baterai dalam perencanaan ini digunakan baterai NiCad dengan kapasitas 1200 mA/H

Jadi :

$$I_E = 1200\text{ mA} / 10$$

$$I_E = 120\text{ mA}$$

$$I_E = V_E / R_E$$

$$\text{Dimana } V_E = V_B - 0,6$$

$$= 5,1\text{V} - 0,6$$

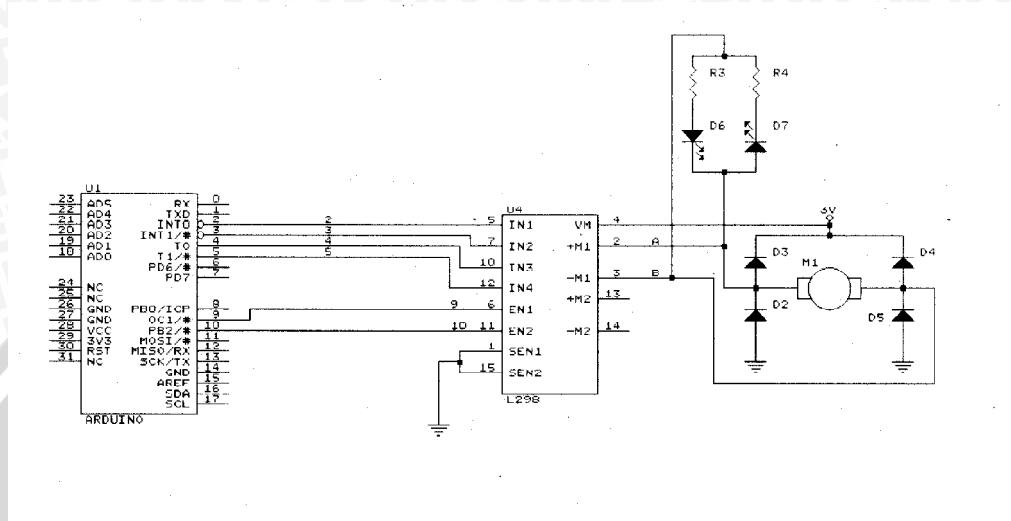
$$= 4,5\text{V}$$

$$\text{Maka } R_E = V_E / I_E = 37,5\Omega$$

2.2.2.2 Perancangan Rangkaian *Driver* Pengendali Motor DC L298N

Modul pengendali motor DC ini digunakan untuk mengendalikan arah dan putaran motor DC yang menjadi penggerak pada pintu pagar. Rangkaian ini dihubungkan dengan Mikrokontroler *ArduinoUNO*. *Driver* pengendali menggunakan *driver* L298N yaitu sebuah rangkaian terintegrasi (IC) yang berfungsi untuk menguatkan arus keluaran dari mikrokontroler dan menyediakan konfigurasi jembatan H (*H Bridge*) sehingga mikrokontroler dapat mengatur arah putaran motor melalui pin masukan IN_1 dan IN_2 . L298N juga menyediakan pin *Enable* (EN_1 dan EN_2) dengan memasukkan

sinyal PWM pada pin ini mikrokontroler dapat mengatur kecepatan Motor DC tanpa merubah tegangan input nominal dari Motor DC. Skema rangkaian *driver* motor L298N ditunjukkan dalam Gambar 4.8.



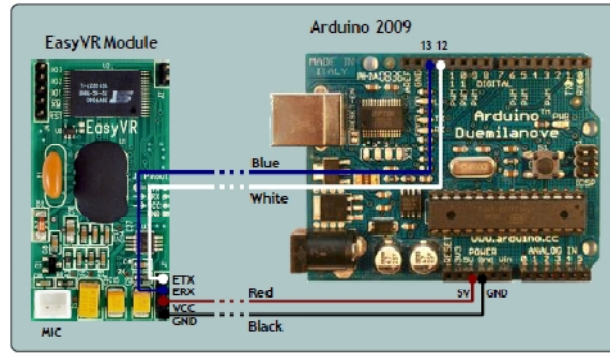
Gambar 4.8. Rangkaian *Driver* Motor DC

Dalam perancangan konfigurasi *driver* L298N ini untuk menjalankan motor, pin *enable* A dan *enable* B dihubungkan pada Pin 9 (PWM) dan pin 10 (PWM) pada *ArduinoUNO* dan diberikan sinyal PWM. Pada V_{CC} menggunakan tegangan sebesar 7,1 V dan V_{DD} sebesar 5 V.

Output dari IC L298N ini tidak memiliki dioda pengaman oleh karena itu pada perancangan *driver* motor ini ditambahkan 2 buah dioda pada setiap titik *output*. Menggunakan 2 buah LED sebagai indikator pergerakan pagar. LED berwarna hijau untuk motor bergerak ke kanan dan ke kiri untuk menutup dan membuka pagar dan LED berwarna merah untuk kode *password* yang ditolak.

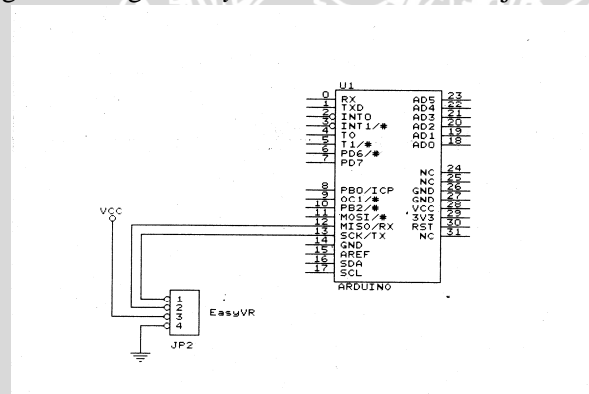
2.2.2.3 Perancangan Komunikasi *EasyVR*

Pada perancangan rangkaian *EasyVR* menggunakan modul *ArduinoUNO* sebagai pengolah utama dalam melakukan proses pengolahan data. Konfigurasi perkawatan (*Wiring Configuration*) *EasyVR* dengan *ArduinoUNO* ditunjukkan dalam Gambar 4.9



Gambar 4.9. Konfigurasi perkawatan (*Wiring Configuration*) EasyVR dengan *ArduinoUNO*
 Sumber :*Datasheet EasyVR*

Komunikasi antara *EasyVR* dengan *ArduinoUNO* menggunakan *Bridge Mode* atau secara *virtuil serial software*. Pin 13 pada *ArduinoUNO* difungsikan sebagai Tx yang akan dihubungkan dengan Rx dari *EasyVR*. Sedangkan pin 12 pada *ArduinoUNO* difungsikan sebagai Rx yang akan dihubungkan dengan Tx pada *EasyVR*. Dalam perancangan alat ini perancangan hubungan *EasyVR* dan *Arduino* ditunjukkan dalam Gambar 4.10

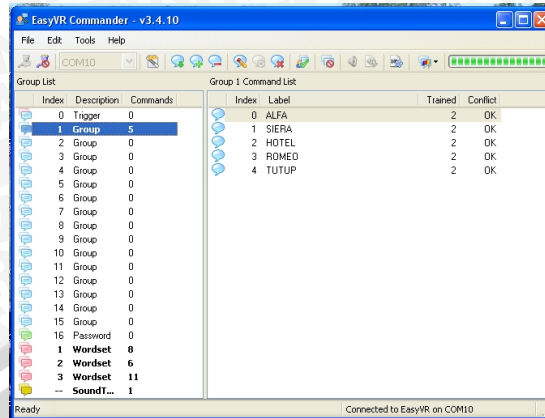


Gambar 4.10. Minimum Sistem Rangkaian *EasyVR*

Dalam perancangan ini Pin 2 (TX) dan pin 3 (RX) pada *EasyVR* dihubungkan dengan pin 12 (RX) dan pin 13 (TX) pada *Arduino*.

2.3 Perancangan Sistem Voice Recognition Menggunakan EasyVR

Perancangan ini bertujuan untuk mengambil *sample* suara yang akan disimpan ke dalam modul *EasyVR*. Pengambilan sampling suara dilakukan melalui PC dengan *software* bawaan dari *EasyVR* yaitu *EasyVR Commander*. *Sample* suara yang akan digunakan adalah sebanyak tujuh kata.



Gambar 4.11. Sampling Suara Pada *EasyVR Commander*

Gambar 4.11 menunjukkan sampling data yang diambil dan terdapat dalam *group 1*. Berikut adalah keterangan dari masing-masing hasil sampling:

- ALFA : Berisi sampling suara dengan kata “alfa” yang berfungsi untuk kata pertama atau salah satu kata dalam pemasukan *password* membuka pintu. Karakter ASCII yang digunakan adalah 0.
- SIERA :Berisi sampling suara dengan kata “siera” yang berfungsi untuk kata kedua dari *password* yang dimasukkan. Karakter ASCII yang digunakan adalah 1.
- HOTEL:Berisi sampling suara dengan kata “hotel” yang berfungsi sebagai kata ketiga untuk *password* yang dimasukkan. Karakter ASCII yang digunakan adalah 2.
- ROMEO:Berisi sampling suara dengan kata “romeo” yang berfungsi sebagai salah satu kata dari *password* yang dimasukkan untuk membuka pagar dan sekaligus pintu akan terbuka jika semua kata *password* telah benar. Karakter ASCII yang digunakan adalah 3.
- TUTUP: Berisi sampling suara dengan kata “tutup” yang berfungsi untuk menggerakkan pintu pagar untuk menutup. Karakter ASCII yang digunakan adalah 4.

2.4 Perancangan dan Pembuatan Perangkat Lunak

Setelah melalui proses perancangan dan pembuatan perangkat keras, selanjutnya akan dilakukan perancangan dan pembuatan perangkat lunak. Perangkat lunak berfungsi untuk memberikan instruksi kerja kepada perangkat keras tersebut.

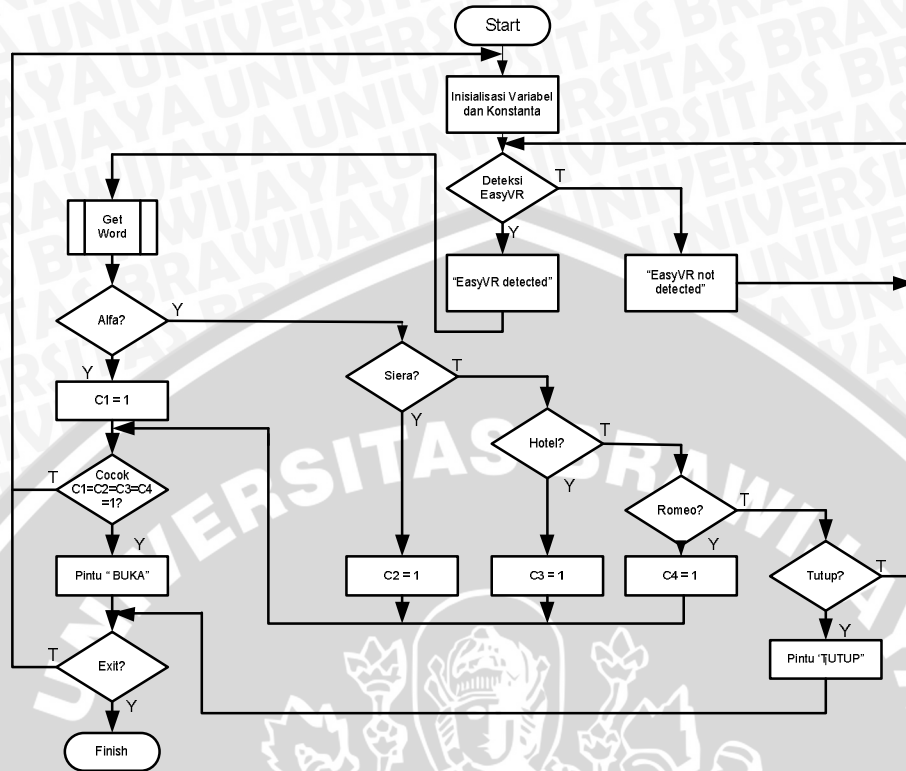
2.4.1 Perancangan Susunan Perangkat Lunak

Tahapan proses yang terdapat pada sistem ini meliputi proses pengolahan data dari modul *EasyVR* ke *Arduino* dan proses pengontrolan motor DC. Semua proses tersebut dilakukan oleh perangkat lunak yang terdapat dalam Mikrokontroler. Perangkat lunak ini tersusun dari instruksi-instruksi yang membentuk sebuah *listing* program atau *source code*.

Semua intruksi program disusun secara terstruktur dalam beberapa subrutin yang secara khusus menangani fungsi tertentu. *Software* Mikrokontroler dibuat menggunakan program *Arduino*.

2.4.2 Diagram Alir Keseluruhan Sistem

Diagram alir keseluruhan sistem ditunjukkan dalam Gambar 4.12. Dimana saat *EasyVR* sudah terdeteksi maka akan dilakukan memasukkan *password*. Setelah itu, data *input* berupa suara “ALFA” dimasukkan dan dibandingkan dengan data sampling yang ada pada basis data suara pada *EasyVR*. Jika sesuai, board *ArduinoUNO* akan menerima kode ASCII dari *password* melalui komunikasi serial kemudian board *ArduinoUNO* melakukan validasi *password* jika tidak sesuai maka akan dilakukan memasukkan *password* berikutnya “SIERA” dan dilakukan proses yang sama. Apabila semua kata *password* yang dimasukkan telah divalidasi oleh *ArduinoUNO* maka akan dikirimkan perintah ke motor *driver* untuk membuka pintu pagar. Dan untuk memasukkan perintah tutup maka hanya dilakukan sekali pencocokan dan jika sesuai maka *ArduinoUNO* akan mengirimkan perintah ke motor *driver* untuk menutup pintu pagar. Dan untuk memasukkan perintah tutup maka hanya dilakukan sekali pencocokan dan jika sesuai *ArduinoUNO* akan mengirimkan perintah ke motor *driver* untuk menutup pintu pagar.



Gambar 4.12 Diagram Alir Keseluruhan Sistem

2.4.3 Diagram Alir Inialisasi Variabel dan Konstanta

Berfungsi untuk menginisialisasi variabel dan konstanta sebagai bagian dari perancangan *software* sistem ini. Berikut adalah cuplikan dari *listing* program inialisasi variabel dan konstanta.

```

int M1=2;
int M2=3;
int E1=9;
.
.
.
#include "EasyVR.h"
#include "Arduino.h"
#include "SoftwareSerial.h"
SoftwareSerial port(12,13);
.
.
//Groups dan Commands
  
```

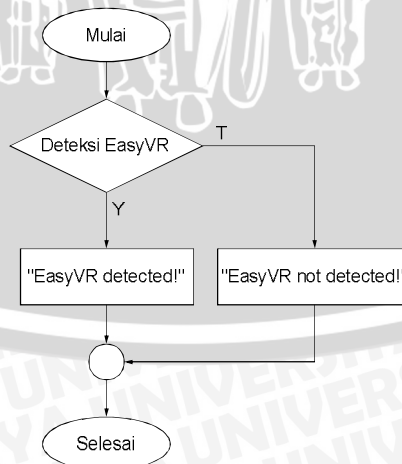
```
enum Groups
{
  //GROUP_0 = 0,
  GROUP_1 = 1,
};

enum Group0
{
  G0_ARDUINO = 0,
};

enum Group1
{
  G1_ALFA = 0,
  G1_SIERA = 1,
  G1_HOTEL = 2,
  G1_ROMEO = 3,
  G1_TUTUP = 4,
};
```

2.4.4 Diagram Alir Program Pendeteksi *EasyVR*

Pendeteksi *EasyVR* berfungsi untuk mengetahui apakah modul *EasyVR* telah terhubung dengan *ArduinoUNO*. Keluaran dari program ini akan muncul pada serial monitor yang terdapat dalam *software Arduino*. Jika *EasyVR* terdeteksi, akan muncul kalimat “*EasyVR Detected*” pada tampilan awal serial monitor. Sebaliknya jika tidak terdeteksi, akan muncul kalimat “*EasyVR not Detected*”. Serial monitor selain berfungsi untuk mengetahui keberadaan *EasyVR* dan untuk mengetahui pengiriman data suara apakah telah sesuai. Diagram alir program pendeteksi *EasyVR* ditunjukkan dalam Gambar 4.13



Gambar 4.13. Diagram Alir Program Pendeteksi *EasyVR*

Berikut adalah cuplikan *listing* program dari inisialisai variabel dan konstanta.

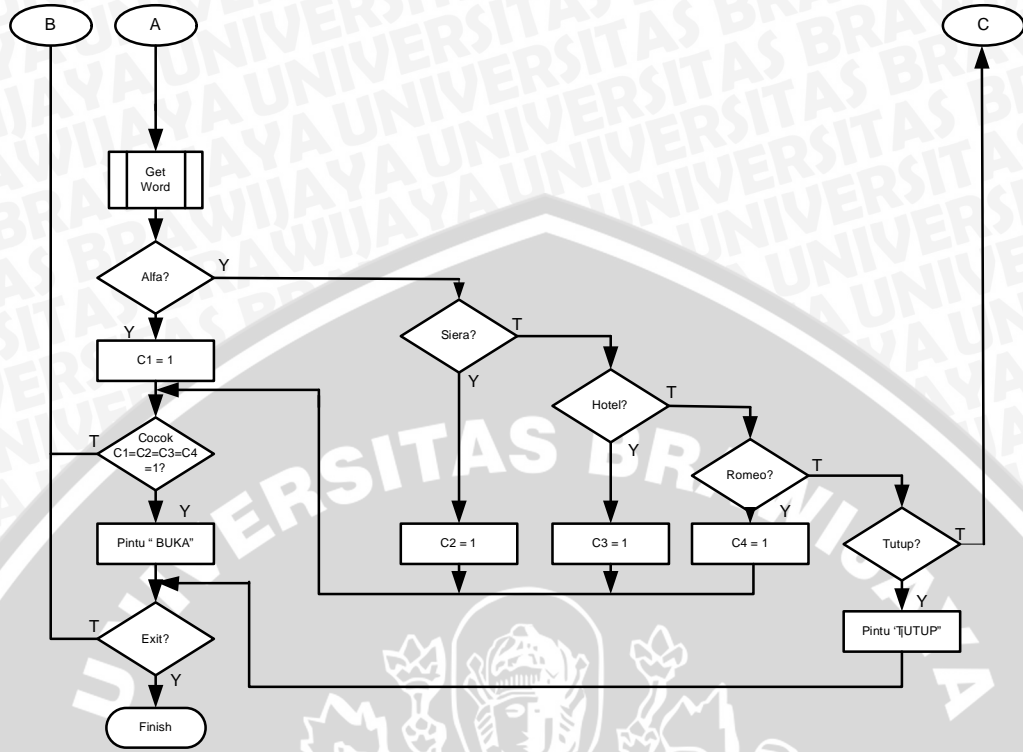
```
if (bridge.check())
{
    .
    .
}
// run normally
Serial.begin(9600);
port.begin(9600);

if (!easyvr.detect())
{
    Serial.println("EasyVR not detected!");
    digitalWrite(fail, LOW);
    digitalWrite(right, LOW);
    delay(1000);
    digitalWrite(fail, HIGH);
    digitalWrite(right, HIGH);
    delay(1000);
    for (;;)
    .
    .
    group = EasyVR::TRIGGER; //&lt;-- start group (customize
    digitalWrite(fail, LOW);
    digitalWrite(right, HIGH);
    delay(1000);
}
```

2.4.5 Diagram Alir Program Action

Dalam perancangan ini berisi rutin program *action* yaitu program untuk mendeteksi dan mencocokkan *password* ,*assign code1*, menyalakan LED Merah dan Hijau sebagai indikator *valid* atau tidaknya *password* yang dimasukkan.

Diagram Alir ditunjukkan dalam Gambar 4.14



Gambar 4.14. Diagram Alir Program Action

Berikut adalah cuplikan dari listing program dari program action :

```

void action()
{
    switch (group)
    case GROUP_1:
    switch (idx)
    {
        case G1_ALFA:
        Serial.print(" ");
        if (code1==0)
        {
            code1=0;
            counter=counter+1;
            digitalWrite(fail, HIGH);
            digitalWrite(right, LOW);
            delay(1000);
        }
        break;
    }
}
  
```

```
case G1_SIERA:
Serial.print(" ");
if (code2==0)
{
code2=1;
counter=counter+1;
digitalWrite(fail, HIGH);
digitalWrite(right, LOW);
delay(1000);
}

break;
.
.
.
case G1_TUTUP:
Serial.print(" ");

if (code5==0)
{
code5=1;
digitalWrite(fail, HIGH);
digitalWrite(right, LOW);
code1=0;
code2=0;
code3=0;
code4=0;
code5=0;
counter=0;

digitalWrite(M1, LOW);
digitalWrite(M2, HIGH);
analogWrite(E1,200);
delay(230);

digitalWrite(M1, LOW);
digitalWrite(M2, HIGH);
analogWrite(E1,0);
delay(300);
}

break;
}

break;
```

