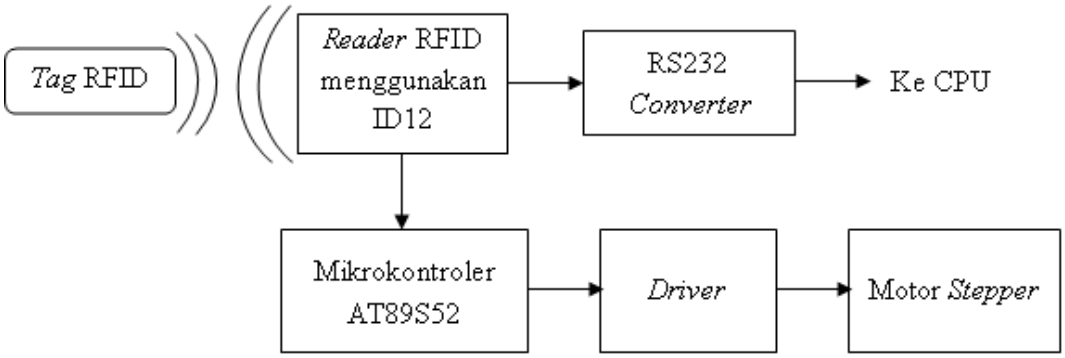
BAB III

PERANCANGAN SISTEM

Pada bab ini akan dibahas tentang fungsi komponen alat yang akan digunakan yang didalamnya memuat prinsip dan cara kerja alat, juga akan membahas tentang perancangan sistem. Perancangan sistem adalah tahap untuk merancang suatu sistem dengan mengkonfigurasikan komponen-komponen perangkat lunak dan perangkat keras sehingga menghasilkan suatu sistem yang baik.

3.1 Blok Diagram Sistem

Secara keseluruhan, blok diagram sistem RFID untuk aplikasi pinti tol otomatis dapat dilihat pada Gambar 3.1 berikut ini.



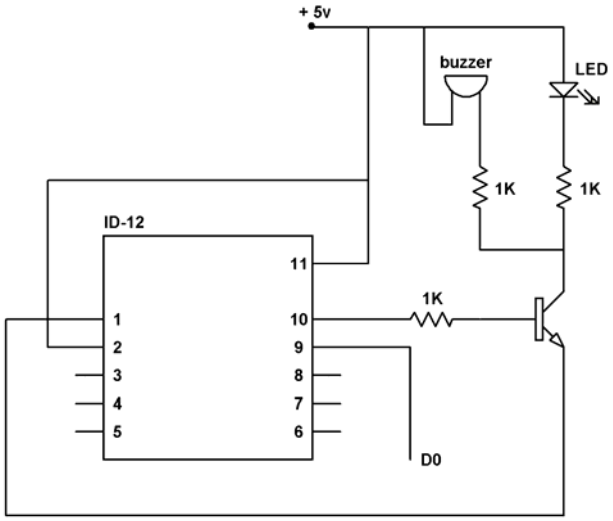
Gambar 3.1 *Blok Diagram Sistem Aplikasi RFID Untuk Pintu Tol Otomatis*

3.2 Perancangan Perangkat Keras (*Hardware*)

Perancangan perangkat keras ini terdiri dari perancangan rangkaian RFID, rangkaian mikrokontroler, rangkaian sumber tegangan, dan rangkaian motor *stepper*.

3.2.1 Rangkaian RFID

Rangkaian RFID berfungsi sebagai pemancar sinyal untuk *tag* RFID dan menerima data yang dikirimkan oleh *tag* RFID tersebut. Rangkaian ini tersusun atas sebuah IC *reader*, tiga buah resistor 1k ohm, sebuah transistor, dan sebuah LED atau beeper sebagai indikator. Untuk lebih jelasnya dapat dilihat pada Gambar 3.2 berikut.



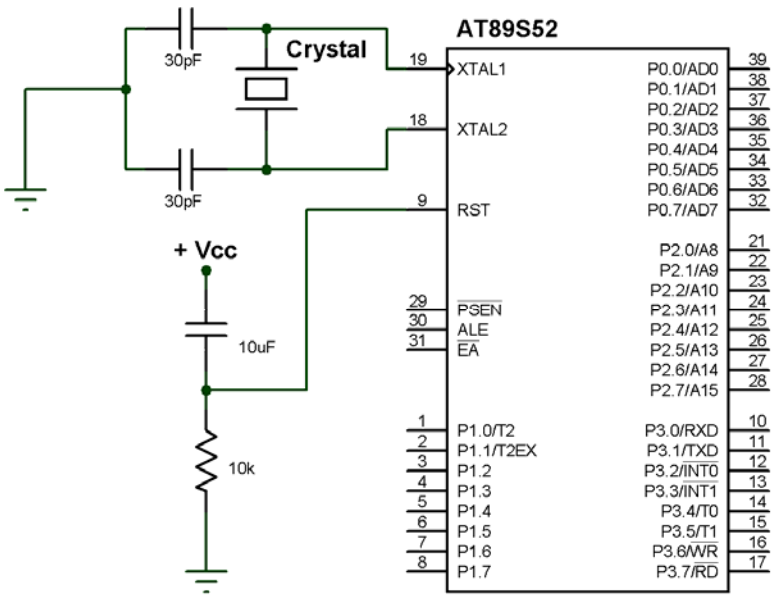
Gambar 3.2 *Rangkaian Reader RFID*

Dalam rangkaian ini IC *reader* yang dipergunakan adalah IC *reader* ID-12 yang diproduksi oleh Innovations. Keluaran dari rangkaian ini akan dihubungkan dengan rangkaian mikrokontroler dengan memanfaatkan pin 9 (D0) sebagai keluaran dari rangkaian RFID.

3.2.2 Rangkaian Mikrokontroler

Rangkaian mikrokontroler merupakan pusat pengolahan data dan pusat pengendali. Didalam rangkaian mikrokontroler terdapat empat buah *port* yang digunakan untuk menampung masukan ataupun keluaran data serta terhubung

langsung dengan rangkaian alat lainnya. Rangkaian ini tersusun atas osilator kristal 11,0592 MHz yang berfungsi untuk membangkitkan pulsa internal dan dua buah kapasitor yang berfungsi untuk menstabilkan frekuensi seperti yang ditunjukkan oleh Gambar 3.3 berikut ini.



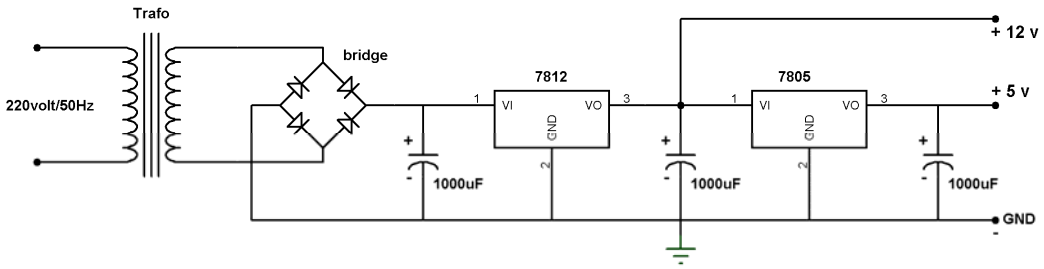
Gambar 3.3 *Rangkaian Minimum Mikrokontroler*

Pada IC mikrokontroler AT89S52 terdapat reset yang berfungsi untuk me-*reset* sistem sehingga proses bisa dijalankan mulai dari awal program lagi (alamat ROM 00H). Rangkaian reset dapat disusun dengan menggunakan sebuah kapasitor dan sebuah resistor yang dihubungkan dengan Vcc. Reset terjadi dengan memberikan logika satu selama minimal dua *machine cycle* pada kaki reset.

3.2.3 Rangkaian Catu Daya

Rangkaian catu daya adalah rangkaian yang memberikan suplai tegangan untuk rangkaian-rangkaian lainnya. Rangkaian ini terdiri dari sebuah transformator

stepdown, dioda *bridge*, IC regulator 7805, dan dua buah kapasitor 1000uF seperti yang terlihat pada Gambar 3.4 berikut.



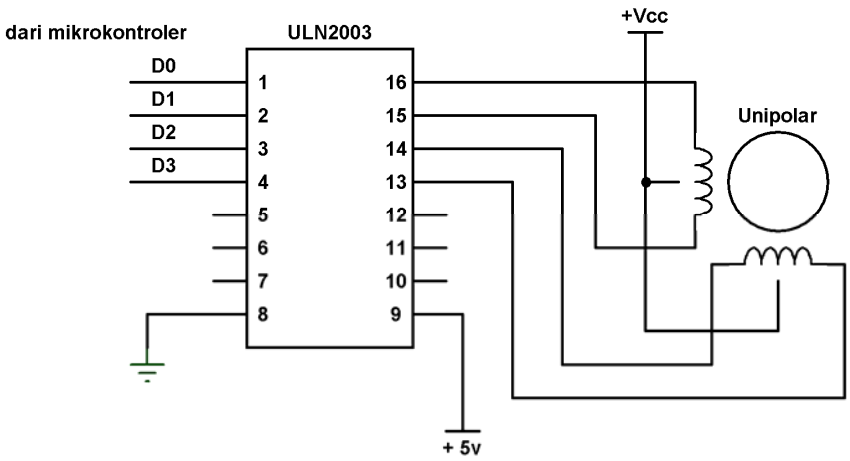
Gambar 3.4 *Rangkaian Catu Daya*

Transformator *stepdown* berfungsi sebagai penurun tegangan AC dari jala-jala listrik PLN (220v/50Hz) menjadi tegangan yang lebih kecil seperti 5 volt, 9 volt, 12 volt atau 15 volt. Lilitan sekunder transformator dihubungkan dengan dioda *bridge* yang berfungsi sebagai penyearah gelombang penuh yang mengubah tegangan AC menjadi tegangan DC. Keluaran dari dioda *bridge* kemudian dihubungkan dengan IC regulator 7812 yang berfungsi sebagai penurun tegangan DC yang merubah tegangan keluarannya menjadi 12 volt DC. Setelah itu keluaran dari IC regulator 7812 dihubungkan lagi dengan IC regulator 7805 yang berfungsi untuk menurunkan tegangan menjadi 5 volt DC. Sedangkan kapasitor yang terletak diantara dioda *bridge* dengan IC regulator 7812 dan IC regulator 7805 berfungsi untuk mengurangi *ripple* agar tegangan keluaran yang dihasilkan nilainya lebih stabil. Semakin besar nilai kapasitor yang digunakan, maka akan semakin stabil tegangan yang dihasilkan.

3.2.4 Rangkaian Motor *Stepper*

Rangkaian motor *stepper* merupakan rangkaian penggerak yang akan menggerakkan pintu tol otomatis. Rangkaian ini tersusun atas sebuah motor

stepper unipolar dan sebuah *driver* seperti yang terlihat pada Gambar 3.5 berikut ini.

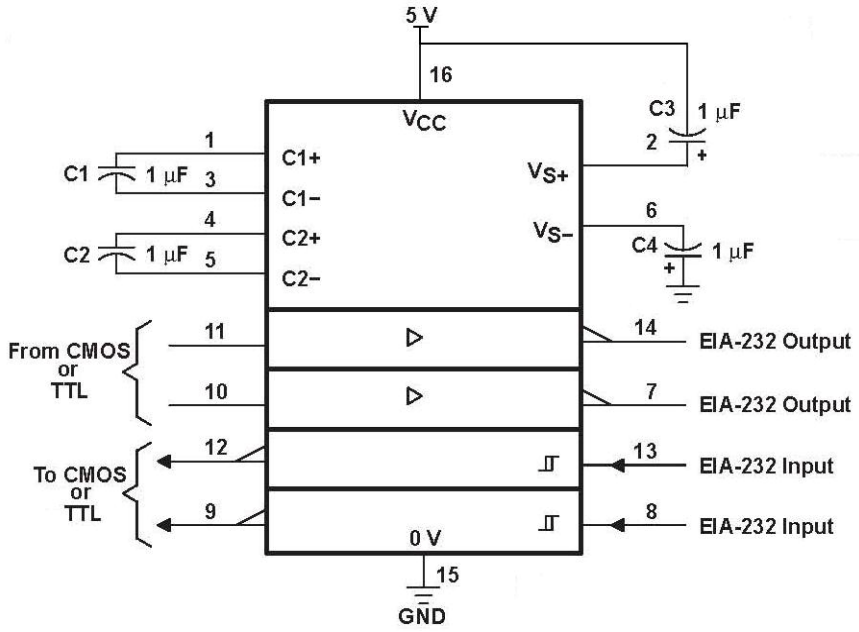


Gambar 3.5 *Rangkaian Motor Stepper dan Driver*

Driver yang digunakan dalam rangkaian ini adalah IC ULN2003A yang didalamnya terdapat tujuh buah rangkaian Darlington. Rangkaian Darlington ini tersusun atas dua buah transistor dan berfungsi sebagai penguat arus. Arus maksimal yang dapat dihasilkan oleh IC ULN2003A adalah sebesar 500mA. Pin-pin masukan *driver* ini dihubungkan dengan rangkaian mikrokontroler.

3.2.5 Rangkaian RS232 *Converter*

Rangkaian RS232 *converter* ini digunakan sebagai penghubung antara rangkaian mikrokontroler dengan komputer. Rangkaian ini berfungsi untuk merubah level tegangan dari level TTL menjadi level RS232. Untuk lebih jelasnya rangkain RS232 *converter* dapat dilihat pada Gambar 3.6 berikut ini.



Gambar 3.6 *Rangkaian RS232 Converter*

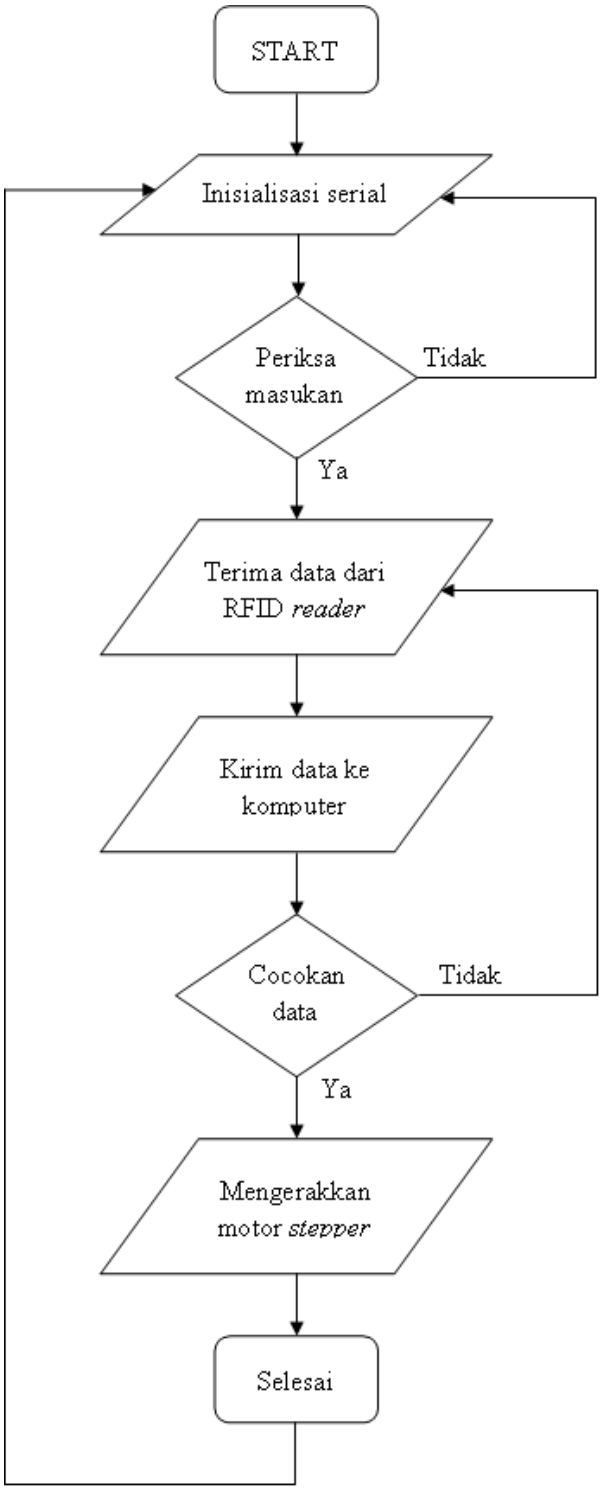
Rangkaian RS232 *converter* ini memanfaatkan sebuah IC MAX232. Komunikasi MAX232 dengan mikrokontroler dilakukan dengan menghubungkan kaki RxD mikrokontroler (pin 10) dengan kaki R1out MAX232 (pin 12) dan dengan menghubungkan kaki TxD mikrokontroler (pin 11) dengan kaki T1in MAX232 (pin 11). Sedangkan komunikasi antara komputer dengan MAX232 dilakukan dengan menghubungkan pin 14 MAX232 dengan pin TxD dari DB9 (pin 2) dan menghubungkan *ground* DB9 (pin 5) dengan *ground* rangkaian.

3.3 Perancangan Perangkat Lunak (*Software*)

Perancangan *software* dilakukan dengan membuat suatu bahasa pemrograman yang akan disimpan kedalam mikrokontroler. Bahasa pemrograman ini berisi instruksi-instruksi yang digunakan untuk mengolah data dan memberikan keluaran untuk rangkaian lainnya.

3.3.1 *Flowchart* Program

Flowchart program menjelaskan tentang program utama yang akan dijalankan oleh mikrokontroler. *Flowchart* program ditunjukkan oleh Gambar 3.7 berikut ini.



Gambar 3.7 *Flowchart Program*

Ketika program pertama kali dijalankan, mikrokontroler akan menginisialisai *port* serial yang akan digunakan untuk menerima data yang terkirim oleh RFID *reader*. Kemudian apabila *port* tersebut mendapat masukan dari *reader*, maka program akan mengambil data yang masuk dan menyimpannya pada SBUF (*Serial Buffer*). Setelah itu progam akan mengirim data ke komputer dan mencocokkan data tersebut. Apabila data cocok, maka program akan memberikan masukan untuk motor *stepper* sehingga motor *stepper* tersebut aktif dalam beberapa saat.

3.3.2 Program Inisialisasi Serial

Sebelum mikrokontroler dapat menerima data dari RFID *reader*, maka harus dilakukan penginisialisasian port terlebih dahulu. Berikut ini adalah program utuk melakukan penginisialisasian port mikrokontroler.

```
void InisialisasiSerial() {
        SCON = 0x50;            //serial control
        TMOD = 0x20;            //timer mode
        TH1  = 0xFD;
        TR1  = 1;
        TI   = 1;               //eksternal timer
        RI   = 0;               //pointer
}
```

3.3.3 Program Terima Data

Setelah dilakukan inisialisasi serial, maka komunikasi antara RFID *reader* dengan mikrokontroler dapat dilakukan. Berikut ini adalah program untuk menerima data serial dari RFID *reader*.

```
char TerimaData() {
        unsigned char x;
```

```
        while (RI !=1) {;}//!= ==>not equal to
        RI = 0;
        if(isalnum(SBUF))
        x = SBUF;
        return x;
}
```

3.3.4 Program Kirim Data

Setelah dilakukan penginisialisasian port mikrokontroler dan data telah diterima, maka instruksi selanjutnya adalah mengirimkan data tersebut ke komputer untuk dicocokkan. Berikut adalah perintah untuk mengirimkan data ke komputer.

```
void KirimData(unsigned char x) {
        while (TI !=1) {;}
        TI = 0;
        SBUF = x;       //serial data buffer
}
```

3.3.5 Program Untuk Menggerakkan Motor *Stepper*

menggerakkan motor *stepper*. Program untuk menggerakkan motor *stepper* adalah sebagai berikut.

```
int stepper(int loop){
int a,b,dt;

for (b=0;b<loop;b++){
        dt = 0x10;
                for (a=0;a<4;a++){
                P1 = dt;
                dt<<=1;
                delay(2);
                }
}
}
```

3.3.6 Program Waktu Tunda (*Delay*)

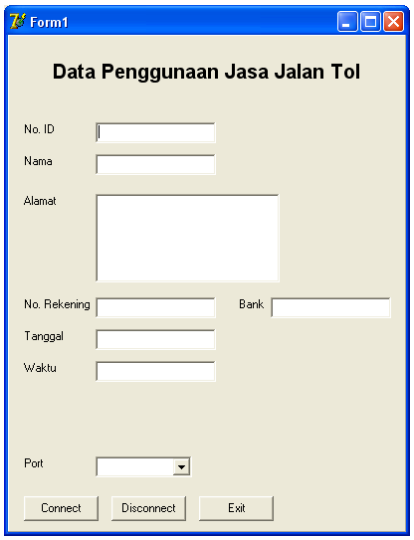
Program waktu tunda atau *delay* digunakan untuk melakukan penundaan waktu sebelum instruksi selanjutnya dijalankan. Berikut ini adalah program untuk melakukan penundaan waktu selama 1 detik.

```
void delay(int m){
        int i,j;
        for (i=1;i<m;i++){
            for (j=1;j<1000;j++){;}
        }

}
```

3.4 Perancangan Tampilan Pada Delphi

Setelah data berhasil diterima oleh mikrokontroler, mikrokontroler kemudian mengirimkan data tersebut ke komputer untuk dicocokan dan ditampilkan. Berikut ini adalah gambar tampilan program pada komputer dengan menggunakan *software* Delphi 7.



Gambar 3.8 *Tampilan Utama Program*

Gambar 3.8 di atas merupakan tampilan utama program pada komputer. Tampilan tersebut digunakan untuk menampilkan informasi dari pemilik kendaraan. Penjelasan dari tampilan di atas dapat dilihat pada tabel 3.1 berikut ini.

Tabel 3.1 *Penjelasan Dari Tampilan Program*

Nama	Keterangan
Nomor Id	Menampilkan nomor seri dari *tag* RFID
Nama	Menampilkan nama pemilik kendaraan
Alamat	Menampilkan alamat pemilik kendaraan
No. Rekening	Menampilkan nomor rekening pemilik kendaraan yang akan dikirimkan tagihan pembayaran jasa penggunaan jalan tol
Bank	Menampilkan nama bank yang digunakan pemilik kendaraan
Tanggal	Menampilkan tanggal ketika pemilik kendaraan melewati *reader* RFID
Waktu	Menampilkan waktu ketika pemilik kendaraan meleati *reader* RFID
Port	Memilih port mana yang akan digunakan dalam komunikasi serial dengan mikrokontroler
Tombol *Connect*	Mengkoneksikan komputer dengan mikrokontroler dengan memilih port terlebih dahulu
Tombol *disconnect*	Memutus koneksi antara komputer dengan mikrokontroler
Tombol *exit*	Keluar dari aplikasi

Agar program dapat berjalan dengan baik dan benar, maka harus dimasukkan suatu perintah-perintah program pada Delphi. Berikut ini adalah perintah-perintah program yang digunakan.

a. Perintah untuk melakukan koneksi dengan mikrokontroler dan menampilkan datanya pada *textbox* Nomor Id.

```
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var
  str : string;
begin
```

```
 comport1.ReadStr(str, count);
 no_id.Text := no_id.Text + str;
end;
```

b. Perintah untuk menampilkan data pemilik kendaraan

```
procedure TForm1.no_idChange(Sender: TObject);
begin
if no_id.Text = '15005B61C5EA' then
 begin
  nama.Text := 'Ongko Widjoyo';
  alamat.Text := 'Taman Tridaya Indah Jl. Anyelir Raya blok E1 no. 2';
  no_rek.Text := '12345678';
  bank.text := 'Bank Sejahtera';
```

dan seterusnya.

c. Perintah untuk menampilkan tanggal

```
tgl.Text:=formatdatetime ('dd mmmm yyyy', now);
```

d. Perintah untuk menampilkan waktu

```
waktu.Text:=formatdatetime ('hh:nn:ss', now);
```

e. Perintah untuk mengkoneksikan komputer dengan rangkaian mikrokontroler

```
comport1.Port:=pil_port.Text;
comport1.Connected:=true;
```

f. Perintah untuk memutuskan koneksi komputer dengan mikrokontroler

```
comport1.Connected:=false;
```

g. Perintah untuk keluar dari aplikasi

```
application.Terminate;
```