BAB III

PERANCANGAN *STAND ALONE* RFID *READER*

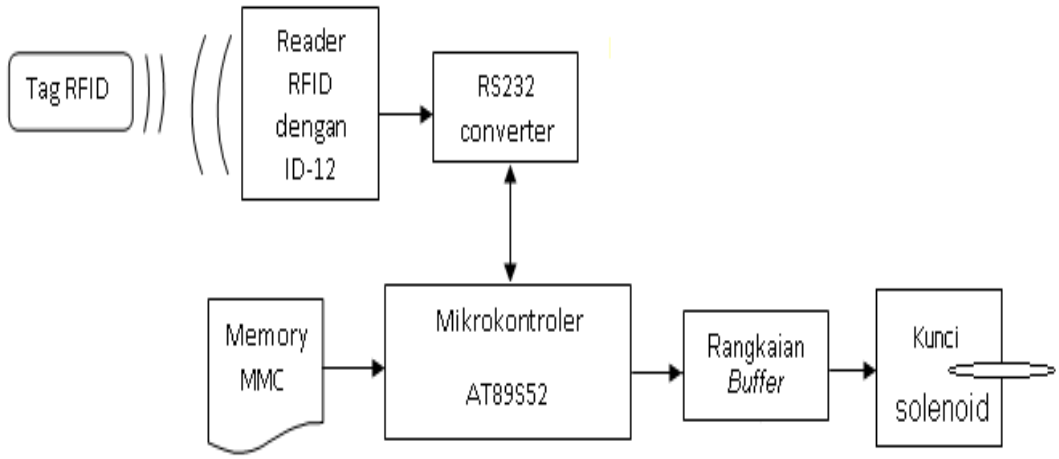
3.1 Perancangan Sistem

Dalam penelitian ini, perancangan sistem meliputi :

a. perancangan perangkat keras (*hardware*) dengan membuat *reader* RFID yang *stand alone* dengan menggunakan ID-12 sebagai IC pembaca, mikrokontroler AT89S8252 sebagai pengontrol dan MMC sebagai memori penyimpan *database*.

b. perancangan perangkat lunak (*software*) dengan membuat program yang akan digunakan sebagai pengontrol dengan menggunakan mikrokontroler AT89S8252.

Secara keseluruhan, blok diagram sistem *stand alone* RFID *reader* ini dapat dilihat seperti Gambar 3.1 di bawah ini :



Gambar 3.1 *Blok Diagram Stand alone RFID Reader*

Reader RFID menggunakan IC ID-12 yang secara otomatis akan memancarkan gelombang elektromagnetik dan kemudian sebuah *tag* dengan frekuensi yang sama akan aktif sehingga memancarkan gelombang elektromagnetik juga. Dalam hubungan seperti itu terjadi pengiriman data dari *tag* ke *reader*. Data tersebut berupa identitas dari *tag*. Kemudian data akan dilanjutkan ke mikrokontroler yang kemudian akan diproses di dalam mikrokontroler tersebut. Di samping itu, antara mikrokontroler dengan MMC juga terjadi pengiriman data secara serial dan terus menerus. Data dari MMC juga akan diproses dan dibandingkan dengan data yang diterima dari *reader*. Apabila data tersebut sama atau cocok maka akan mengaktifkan kunci solenoid yang artinya bahwa pintu dapat dibuka.

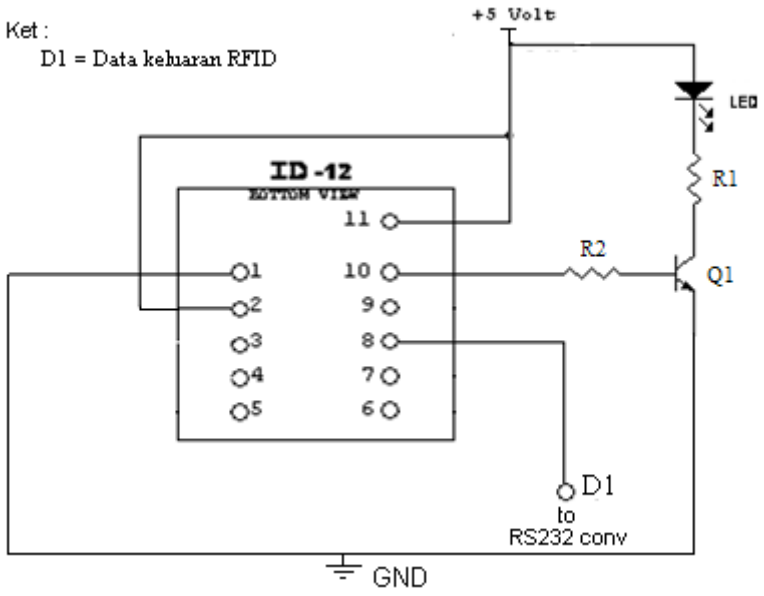
3.2 Perancangan Perangkat Keras (*Hardware*)

Dalam penelitian ini terdapat beberapa perangkat keras yang akan digunakan sebagai sistem *stand alone* RFID *reader* antara lain *reader* RFID, mikrokontroler AT89S8252, catu daya, RS232 *Converter*, relay.

3.2.1 *Reader* RFID

Pembuatan *reader* RFID memanfaatkan IC ID-12. Rangkaian RFID dapat dilihat seperti Gambar 3.2 di bawah ini. Keluaran pada rangkaian tersebut akan dihubungkan dengan sistem mikrokontroler AT89S8252 dengan memanfaatkan pin 8 (D1) sebagai keluaran dari rangkaian RFID. Pin ini akan digunakan sebagai jalur pengiriman data yang terbaca dari *tag*. Cara komunikasi rangkaian RFID

dengan mikrokontroler adalah secara serial, artinya pengiriman bit demi bit data dilakukan secara bergantian mengikuti siklus *clock* tertentu.



Gambar 3.2 *Rangkaian Sederhana RFID Reader dengan ID-12*

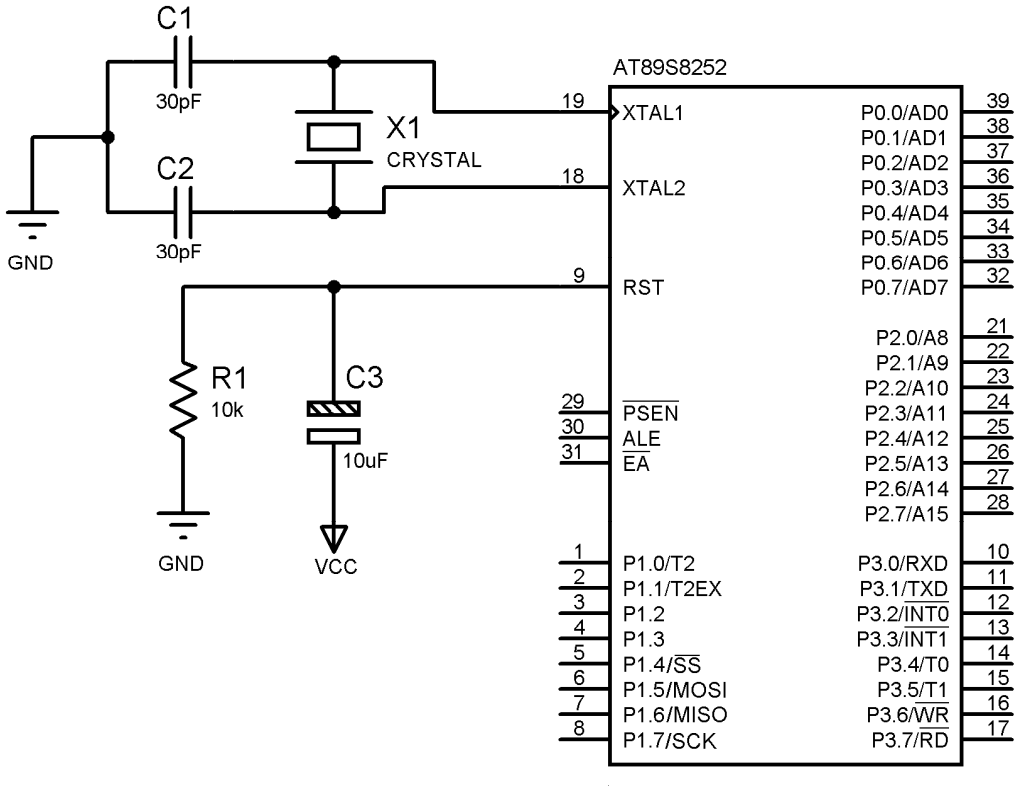
Sesuai rekomendasi dari *datasheet* ID *Innovations* ID12, harga komponen-komponen pada rangkaian RFID tersebut adalah :

- R1 = 1K Ω
- R2 = 1K Ω
- Q1 = BC337 (NPN)
- ID-12 = ID Innovations ID12

3.2.2 Mikrokontroler AT89S8252

Dalam penelitian ini, yang digunakan sebagai pengontrol adalah mikrokontroler AT89S8252 yang merupakan mikrokontroler terbaru dan relatif lebih murah. Selain itu memiliki beberapa kelebihan dibandingkan dengan mikrokontroler seri

sebelumnya. Gambar 3.3 merupakan gambar rangkaian sistem minimum AT89S8252 dengan frekuensi kristal yang digunakan sebesar 11,0592 MHz dan dua buah kapasitor masing-masing sebesar 30pF. Fungsi kapasitor di sini adalah untuk menstabilkan osilasi yang dihasilkan oleh kristal. Penempatan antara kapasitor dengan kristal diusahakan sedekat mungkin untuk menghindari terjadinya *noise*. Rangkaian yang tersusun atas kristal dan dua kapasitor tersebut disebut rangkaian osilator yang merupakan subsistem dari mikrokontroler yang berfungsi untuk membangkitkan *clock* pada mikrokontroler. *Clock* tersebut diperlukan oleh mikrokoktroler untuk mensinkronkan proses yang sedang berlangsung dalam mikrokontroler tersebut.



Gambar 3.3 S*istem Minimum AT89S8252*

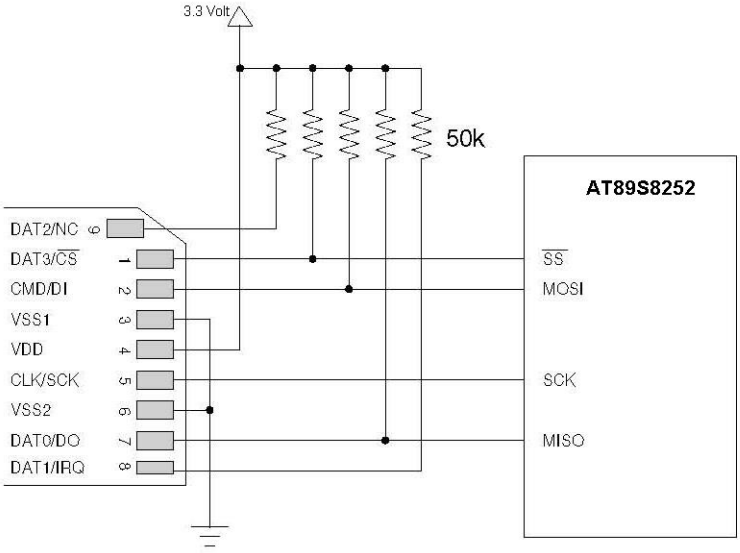
Selain osilator, dalam sistem minimum tersebut juga terdapat rangkaian *reset*. Rangkaian ini dibuat untuk me-*reset* sistem sehingga proses dapat dijalankan mulai dari awal lagi. Kemudian agar dapat berfungsi maka pin 40 (Vcc) dihubungkan pada *supply* sebesar 5 Volt. Sedangkan untuk *ground* dihubungkan pada pin 20 (GND). Selanjutnya hubungan antara mikrokontroler AT89S8252 dengan rangkaian lain dapat dilihat pada Tabel 3.1 di bawah ini.

Tabel 3.1 *Fungsi Pin Mikrokontroler dengan Rangkaian Lain*

Pin	Nama	Keterangan
5	P1.4/SS	*Slave Select* terhubung dengan pin *Chip Select* (CS) pada MMC
6	P1.5/MOSI	*Master Out Slave In* (pengiriman data dari mikrokontroler ke MMC)
7	P1.6/MISO	*Master In Slave Out* (menerima data dari MMC ke mikrokontroler)
8	P1.7/SCK	*Clock*
10	P3.0/RxD	Menerima data yang dikirim oleh rangkaian RFID

3.2.3 Rangkaian Komunikasi SPI pada MMC

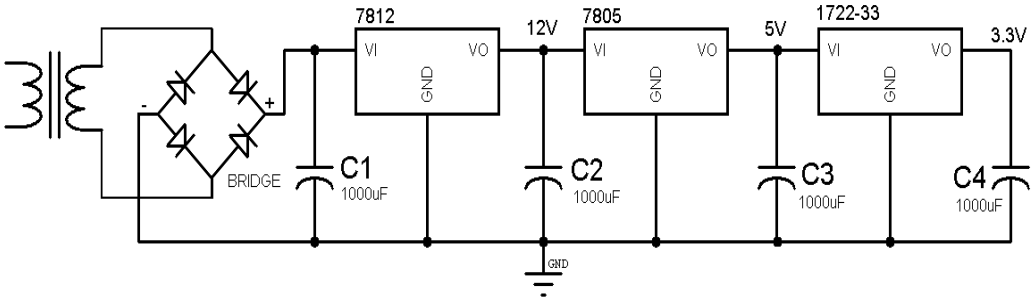
MMC digunakan sebagai media penyimpanan database. MMC ini dihubungkan dengan mikrokontroler AT89S8252 secara *Serial Peripheral Interface* (SPI). MMC akan mengirimkan data yang ada ke dalam mikrokontroler yang kemudian akan diproses oleh mikrokontroler tersebut. *Supply* tegangan yang dibutuhkan adalah 3,3 volt DC. Gambar 3.4 di bawah ini merupakan gambar rangkaian MMC dengan mikrokontroler AT89S8252.



Gambar 3.4 *Rangkaian MMC/ SD Card*

3.2.4 Catu Daya

Agar semua rangkaian dapat bekerja, maka diperlukan sebuah rangkaian sebagai sumber tegangan. Rangkaian ini digunakan untuk rangkaian DC (*Direct Current*) atau rangkaian searah. LM7805 merupakan sebuah IC regulator yang digunakan untuk rangkaian yang membutuhkan tegangan sebesar 5 Volt seperti mikrokontroler dan RFID. Sedangkan LM7812 digunakan untuk rangkaian yang membutuhkan tegangan sebesar 12 Volt. Selain itu digunakan pula 1722-33 yang merupakan IC regulator 3.3 volt dan dimanfaatkan sebagai sumber tegangan pada MMC. Untuk lebih jelasnya rangkaian catu daya dapat dilihat seperti Gambar 3.5 di bawah ini.

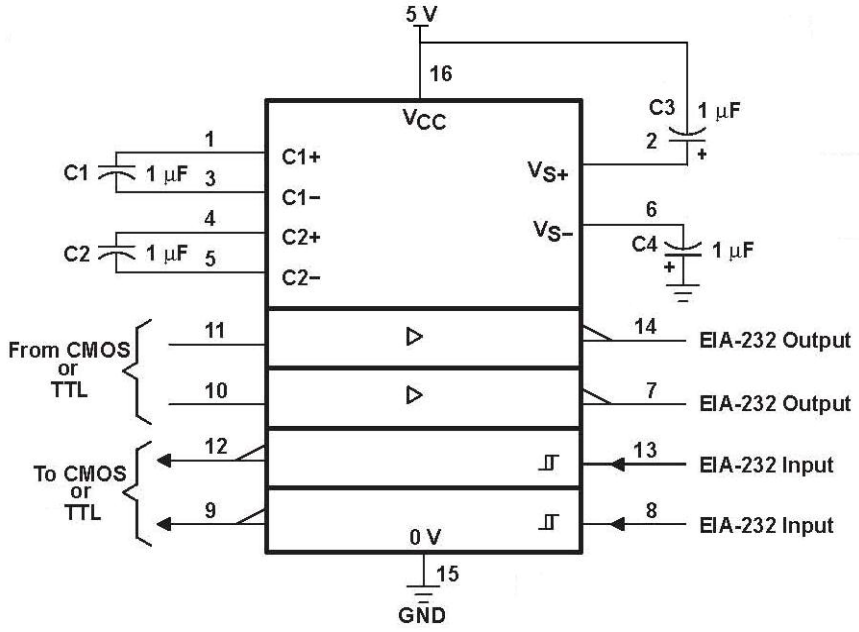


Gambar 3.5 *Rangkaian Catu Daya*

Rangkaian dioda digunakan untuk menyearahkan gelombang AC ke DC. Gelombang tersebut disaring melalui kapasitor sebesar 1000µF yang akan membuat *reeple* gelombang makin kecil. Hal tersebut dilakukan agar mendapatkan tegangan *supply* yang stabil.

3.2.5 RS232 *Converter*

Komunikasi data pada keluaran RFID dilakukan secara serial. Keluaran ini akan masuk ke dalam rangkaian mikrokontroler. Untuk dapat terhubung perlu ada suatu *converter* yang dapat menyesuaikan level tegangan TTL antara RFID dengan mikrokontroler, yaitu dengan menggunakan rangkaian RS232. Rangkaian ini memanfaatkan sebuah IC MAX232 dengan konfigurasi seperti Gambar 3.6 di bawah ini.



Gambar 3.6 *Rangkaian RS232 Converter*

Komunikasi MAX232 dengan mikrokontroler yaitu dengan menghubungkan kaki RxD mikrokontroler (pin 10) dengan R1out MAX232 (pin 12), dan TxD mikrokontroler (pin 11) dihubungkan dengan T1in MAX232 (pin 11). Hubungan antara pin MAX232 dengan rangkaian lain dapat dilihat pada Tabel 3.2 berikut.

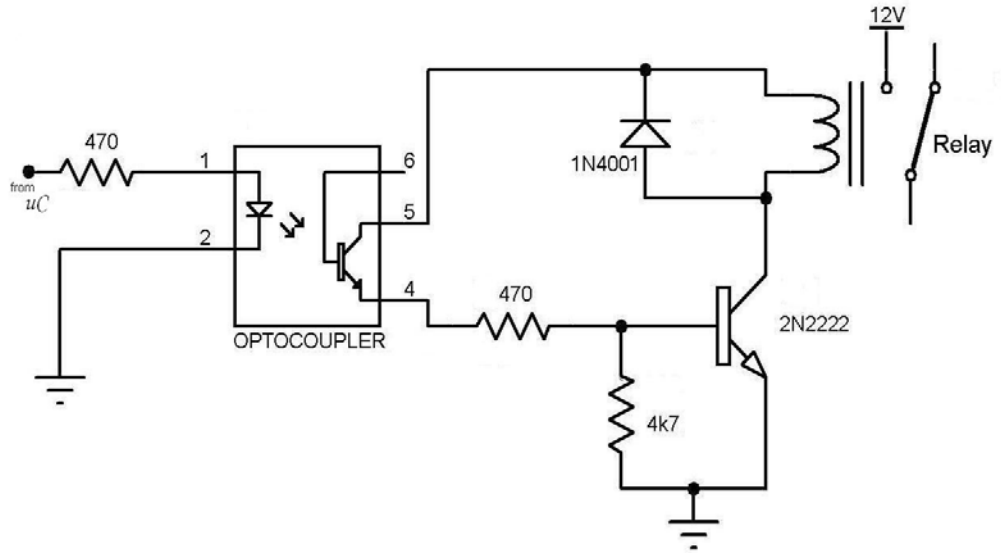
Tabel 3.2 *Hubungan Pin MAX232 dengan Rangkaian Lain*

Pin	Nama	Keterangan
11	T1in	Dihubungkan dengan kaki TxD mikrokontroler (pin 11)
12	R1out	Dihubungkan dengan kaki RxD mikrokontroler (pin 10)
13	R1in	Dihubungkan dengan D1 RFID (pin 8)
7,8		*Ground*
15	GND	*Ground*
16	VCC	Vcc 5 Volt

3.2.6 Relay

Relay merupakan alat yang secara mekanis mengontrol penghubungan rangkaian listrik. Bekerja berdasarkan pembentukan elektromagnet yang menggerakkan

elektromekanis penghubung dar dua atau lebih titik penghubung (konektor) rangkaian sehingga dapat menghasilkan kondisi kontak ON atau OFF atau kondisi keduanya. Sebuah relay biassanya terdiri dari suatu solenoid, sehingga untuk mengontrol suatu relay diperlukan suatu rangkaian *buffer*. Salah satu alternatif rangkaian *buffer* dapat menggunakan rangkaian seperti Gambar 3.7 berikut.



Gambar 3.7 *Rangkaian Buffer Sebagai Pengontrol Relay*

Rangkaian di atas akan bekerja setelah ada masukan dari mikrokontroler yang kemudian akan mengaktifkan relay sehingga saklar pada relay terhubung dengan tegangan 12 Volt. Setelah saklar tersebut terhubung maka kunci solenoid akan bekerja sehingga pintu dapat dibuka.

3.3 Perancangan Perangkat Lunak (*Software*)

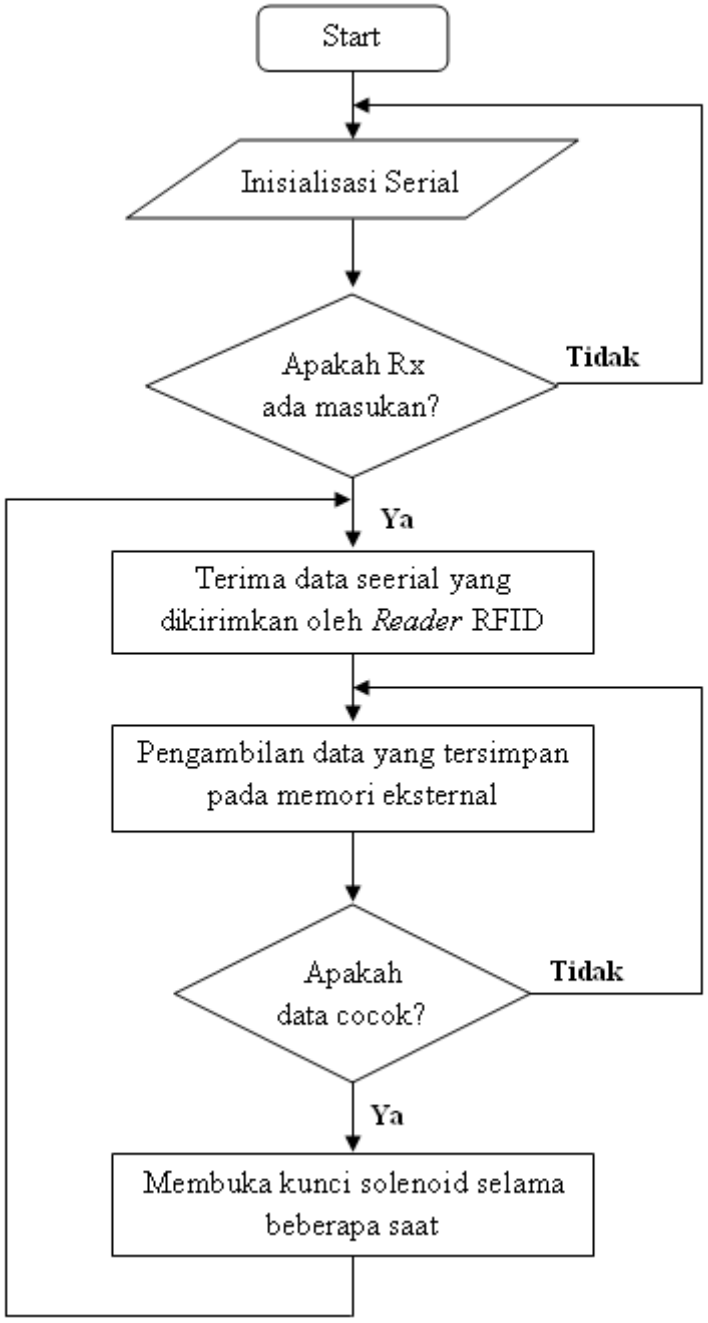
Komunikasi mikrokontroler dengan rangkaian lain dilakukan secara *hardware* maupun *software*. Untuk dapat beroperasi maka mikrokontroler harus diprogram terlebih dahulu sesuai dengan sistem yang akan digunakan. Pembuatan perangkat

lunak dalam sistem ini digunakan sebagai protokol antara mikrokontroler dengan MMC. Selain itu, perangkat lunak juga digunakan untuk mengolah data dan mengendalikan keluaran untuk rangkaian lainnya.

3.3.1 Program Utama

Program utama menunjukkan proses dari mikrokontroler secara global. Proses mikrokontroler secara global tersebut ditunjukkan pada Gambar 3.8 di bawah.

Ketika program dijalankan, maka 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 mengambil data yang tersimpan pada MMC secara berulang dan kemudian mencocokkan data tersebut dengan data masukan serial tadi. Apabila data serial dengan data MMC tidak cocok, maka program akan mengulang langkah tersebut hingga data serial cocok dengan data yang tersimpan pada MMC. Selanjutnya jika data serial sama dengan data yang ada pada MMC, maka rangkaian relay akan aktif sehingga kunci solenoid akan terbuka untuk beberapa saat.



Gambar 3.8 *Flowchart Utama*

3.3.2 Program *Delay*

Program *delay*/ waktu tunda digunakan untuk menunda perintah selanjutnya. Dengan kata lain, memperpanjang waktu dalam mengeksekusi perintah sebelumnya sebelum perintah eksekusi selanjutnya dikerjakan. Prinsip dasarnya adalah pengulangan (*looping*) selama yang diinginkan. Berikut ini merupakan contah program untuk waktu tunda.

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

Variabel m merupakan berapa lama waktu yang akan diperlukan. Kemudian dalam program utama, fungsi tersebut dipanggil dengan format sebagai berikut.

```
void main(){
     delay(10); //watu tunda selama 10 ms
}
```

3.3.3 Program Penerimaan Data Serial

Program penerimaan data digunakan untuk menerima data serial yang melalui port RxD dan TxD pada mikrokontroler. Program ini ditujukan untuk menerima data yang dikirimkan oleh RFID *reader* untuk dapat diproses sebagaimana mestinya di dalam mikrokontroler. Sebelumnya, terlebih dahulu harus dilakukan

penginisialisasian port serial supaya port serial tersebutlah yang digunakan. Berikut ini merupakan program untuk inisialisasi serial.

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

Setelah diinisialisasi, maka kominikasi secara serial dapat dilakukan. Berikut adalah fungsi untuk menerima data serial.

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

     while(RI !=1){;} //!= (tidak sama dengan)
     RI = 0;
     if(isalnum(SBUF))//menghilangkan karakter
     x = SBUF;
     return x;      //mengembalikan nilai x

}
```

Kemudian kedua fungsi tersebut dipanggil dalam program utama.

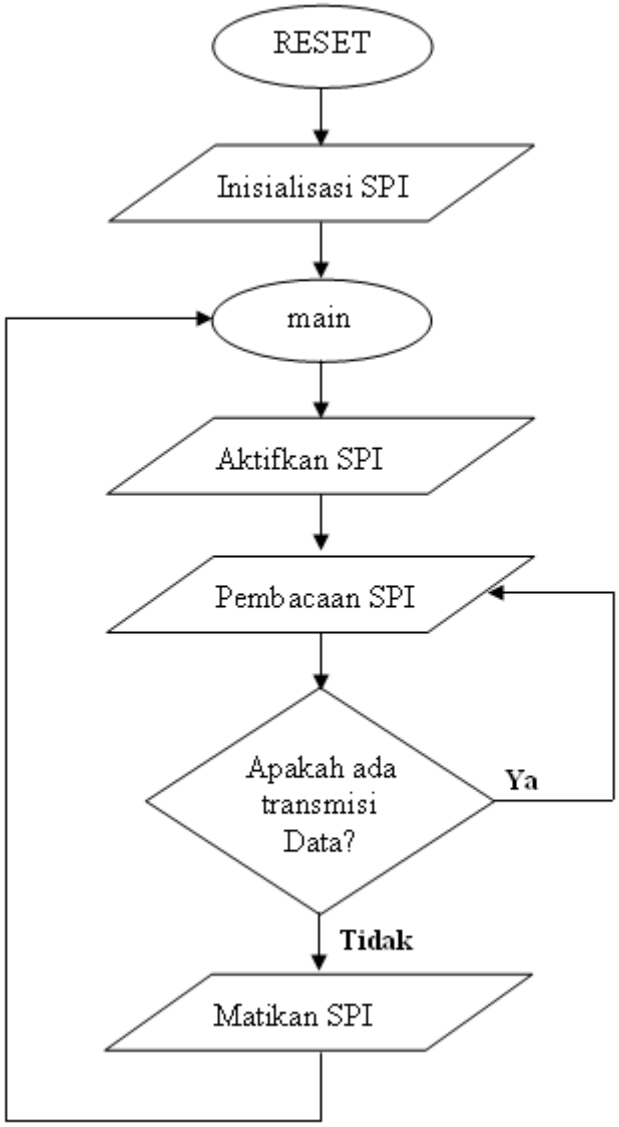
```
void main(){
     unsigned char data_reader[1];

     InisialisasiSerial();
     while(1==1){        //kondisi saat ada masukan
```

```
            data_reader[0]=TerimaData();
            delay(1);
      }
}
```

3.3.4 Program SPI

Program ini digunakan untuk mengambil data yang tersimpan pada memori. Sebelum menginisialisasi port SPI terlebih dahulu program melakukan reset seluruh port. Kemudian dilakukan inisialisasi port SPI yang akan digunakan yaitu SS, MOSI, MISO dan SCK. Selanjutnya program akan masuk ke dalam program utama sampai ada instruksi untuk mengaktifkan SPI. Setelah port SPI diaktifkan, maka dilakukan pembacaan data SPI. Pembacaan akan terus dilakukan sampai batas tertentu sesuai dengan yang diinginkan. Apabila banyaknya pembacaan telah mencapai batas yang ditentukan biasanya delapan kali pembacaan atau 8-bit data, maka program SPI dimatikan dan selanjutnya akan kembali ke program utama. Program SPI akan aktif kembali sampai program tersebut diaktifkan. Program SPI ini mempunyai urutan seperti terlihat pada Gambar 3.9 dibawah ini.



Gambar 3.9 *Flowchart Program SPI*