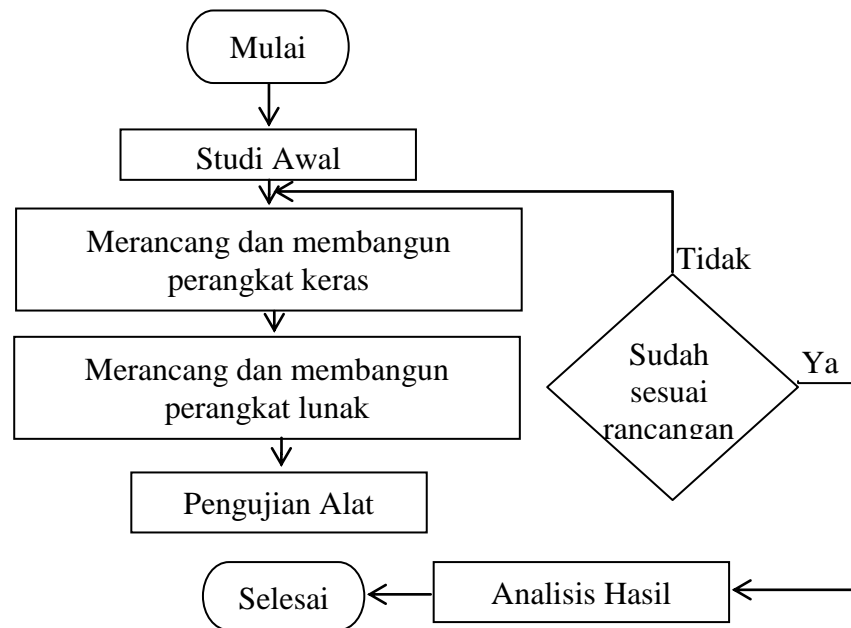


## **BAB 3**

### **PERANCANGAN SISTEM**

#### **3.1.Langkah Penelitian**

Penelitian dimulai dengan studi awal terhadap karya-karya sejenis yang menggunakan NFC untuk mengakses pintu. Setelah itu dilakukan perancangan alat secara keseluruhan. Perancangan alat secara keseluruhan akan dibagi menjadi 2 bagian, yaitu perancangan perangkat keras dan perancangan perangkat lunak. Jika perancangan perangkat keras dan perancangan perangkat lunak telah selesai, dilanjutkan dengan pengujian untuk memeriksa bahwa alat dapat bekerja sesuai dengan rancangan. Apabila terdapat bagian – bagian yang tidak bekerja sesuai dengan rancangan, dilakukan pemeriksaan kembali dari perangkat keras maupun perangkat lunak. Jika setelah diuji kembali, alat dapat bekerja sesuai dengan rancangan, maka dilanjutkan dengan menganalisis hasil penelitian. Gambar 3.1 merupakan *flowchart* dari langkah penelitian yang akan dilakukan dalam penelitian ini.

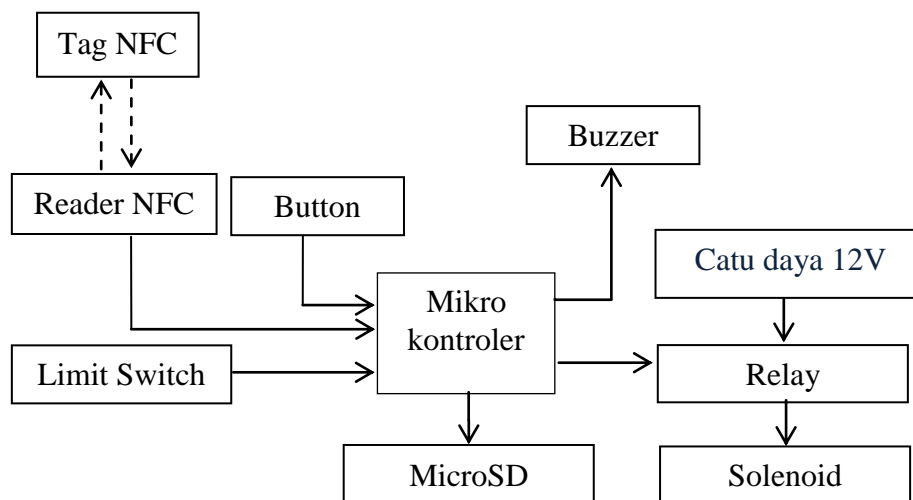


**Gambar 3.1** *Flowchart* langkah penelitian

Pada langkah pengujian alat, terdiri dari pengujian fungsional di masing-masing bagian dan pengujian alat secara keseluruhan. Pengujian ini dilakukan untuk memastikan bahwa setiap bagian bekerja sesuai dengan yang telah dirancang. Pengujian fungsional meliputi pengujian pada NFC, mendaftar dan menghapus ID, pengujian *data logger* pada microSD, pengujian membuka kunci pintu dan pengujian alat secara keseluruhan.

### 3.2. Gambaran Umum

Pengertian perancangan sistem adalah penggabungan beberapa rangkaian yang sudah ada ataupun membuat rangkaian menjadi satu sistem utuh. Pada Gambar 3.2 menjelaskan blok diagram dari perancangan sistem keseluruhan yang akan dibuat.



**Gambar 3.2 Blok diagram sistem**

Gambar 3.2 terdapat beberapa peralatan yang membentuk suatu sistem. Secara umum sistem akan bekerja saat *tag* NFC dibaca oleh *NFC reader*, kemudian ID yang terbaca oleh Arduino ditampilkan pada layar *serial* dan dicatat secara *real time* pada *microSD*. Selanjutnya Arduino akan mengaktifkan solenoid jika ID tersebut ada pada daftar ID di memori Arduino dan *buzzer* juga akan berbunyi. Untuk melakukan pembacaan ID yang lain, *limit switch* harus aktif terlebih dahulu sebagai tanda bahwa pintu sudah dalam keadaan tertutup dan terkunci kembali. Jika ID yang dibaca *reader NFC* tidak terdaftar pada memori Arduino, maka *buzzer* aktif dengan suara yang berbeda dan *solenoid* tidak aktif.

### 3.3. Perancangan Perangkat Keras

#### 3.3.1. NFC (*Near Field Communication*)

Komponen utama pada sistem ini adalah NFC (*Near Field Communication*). Terdapat 2 komponen utama pada NFC (*Near Field Communication*) yang berperan pada sistem, yaitu *tag* NFC dan *NFC reader*. Tipe NFC (*Near Field Communication*) *reader* yang digunakan adalah NXP PN532 dengan

menggunakan komunikasi I2C (*Inter Integrated Circuit*). Maka pada PCB NFC terdapat saklar konfigurasi untuk memilih komunikasi I2C dengan membuat saklar 1 ON dan saklar 2 OFF. Sedangkan tag NFC yang akan digunakan adalah tipe 2 yaitu Mifare 1K ISO/IEC 14443A yang bekerja pada frekuensi yang sama dengan *reader* yaitu 13,56 MHz. NFC terhubung ke RTC (*Real Time Clock*) dengan konfigurasi VCC ke pin VCC, Gnd ke pin Gnd, SDA ke pin SDA dan SCL ke pin SCL. NFC akan terhubung ke Arduino melalui RTC, maka sambungan RTC ke Arduino adalah VCC ke pin 5V, GND ke pin GND, SDA ke pin A4 dan SCL ke pin A5.

### **3.3.2. Ethernet Shield**

Ethernet shield pada sistem ini digunakan untuk menghubungkan microSD dengan Arduino, karena Ethernet Shield memiliki slot untuk dipasang sebuah microSD. MicroSD akan digunakan sebagai *data logger* ID yang dibaca oleh Arduino. Ethernet shield dihubungkan ke Arduino dengan cara menumpuknya di bagian atas Arduino dengan menyesuaikan pin-pinnya. MicroSD yang akan digunakan termasuk pada class 4 dengan kapasitas memori 4 GB.

### **3.3.3. Solenoid dan Relay**

*Solenoid* pada sistem ini digunakan sebagai kunci pintu elektrik dengan tambahan relay sebagai saklarnya. *Solenoid* yang digunakan adalah seri *door lock* LY-03 dengan 12 Volt. Maka, *solenoid* dihubungkan ke relay modul 5 V yang *compatible* dengan Arduino pada pin gnd untuk gnd, pin 5 V untuk VCC dan pin 7 untuk input. Selanjutnya *solenoid* diberi tegangan 12 Volt dengan 2-3Amp dari sebuah adaptor yang terpisah dengan adaptor Arduino sebagai *power supply*.

### 3.3.4. *Limit Switch, Buzzer dan Push Button*

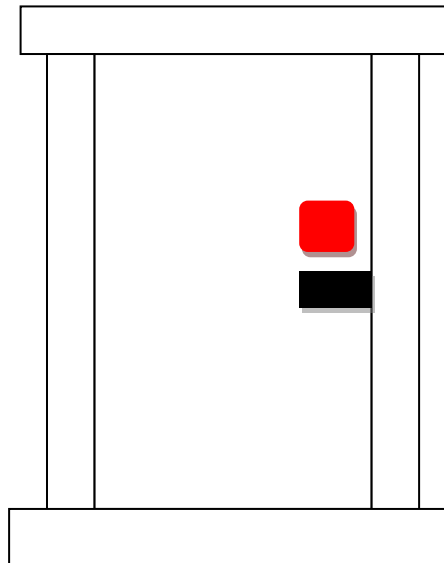
Limit switch berfungsi sebagai indikator saat pintu sedang dalam keadaan tertutup. Jika limit switch belum diaktifkan, maka NFC *reader* tidak dapat membaca *tag* NFC yang lain karena pintu belum dalam keadaan tertutup dan terkunci kembali. Limit switch dihubungkan ke Arduino dengan menyambungkan ke pin GND dan pin 6 sebagai masukan.

Buzzer berfungsi sebagai indikator suara bahwa ID dari *tag* NFC yang dibaca sudah terdaftar di memori Arduino. Buzzer dihubungkan ke pin GND dan pin 5 sebagai keluaran dari Arduino. Saat ID tersebut belum terdaftar di memori Arduino, maka buzzer juga akan berbunyi namun dengan suara yang berbeda.

Selain limit switch dan buzzer, pada sistem ini terdapat 4 buah tombol dengan fungsi yang berbeda-beda. Tombol S1 berfungsi sebagai pengatur waktu untuk kebutuhan *data logger*. Tombol S2 berfungsi sebagai mode penginputan ID baru yang akan dimasukkan ke memori Arduino. Tombol S3 berfungsi sebagai mode untuk menghapus ID yang sudah terdaftar di memori Arduino. Tombol S4 berfungsi untuk memilih ID mana yang akan dihapus dari memori Arduino. Semua tombol tersebut terhubung ke pin analog Arduino dengan konfigurasi S1 ke A0, S2 ke A1, S3 ke A2 dan S4 ke A3 dan *ground* ke pin GND.

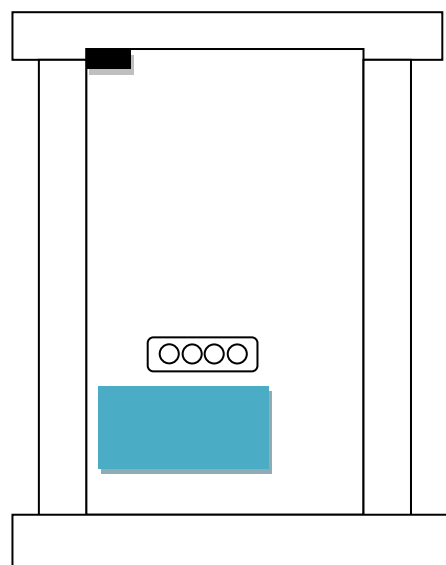
### 3.3.5. *Rancang Bangun Alat*

Rancang bangun alat merupakan perencanaan untuk pemasangan perangkat-perangkat keras yang digunakan pada pintu *prototype*. Perangkat keras akan dipasang dibagian kotak pintu, kecuali 4 buah tombol dan *limit switch*. Gambar 3.3 merupakan rancangan alat tampak dari depan.



**Gambar 3.3 Rancangan alat tampak dari depan**

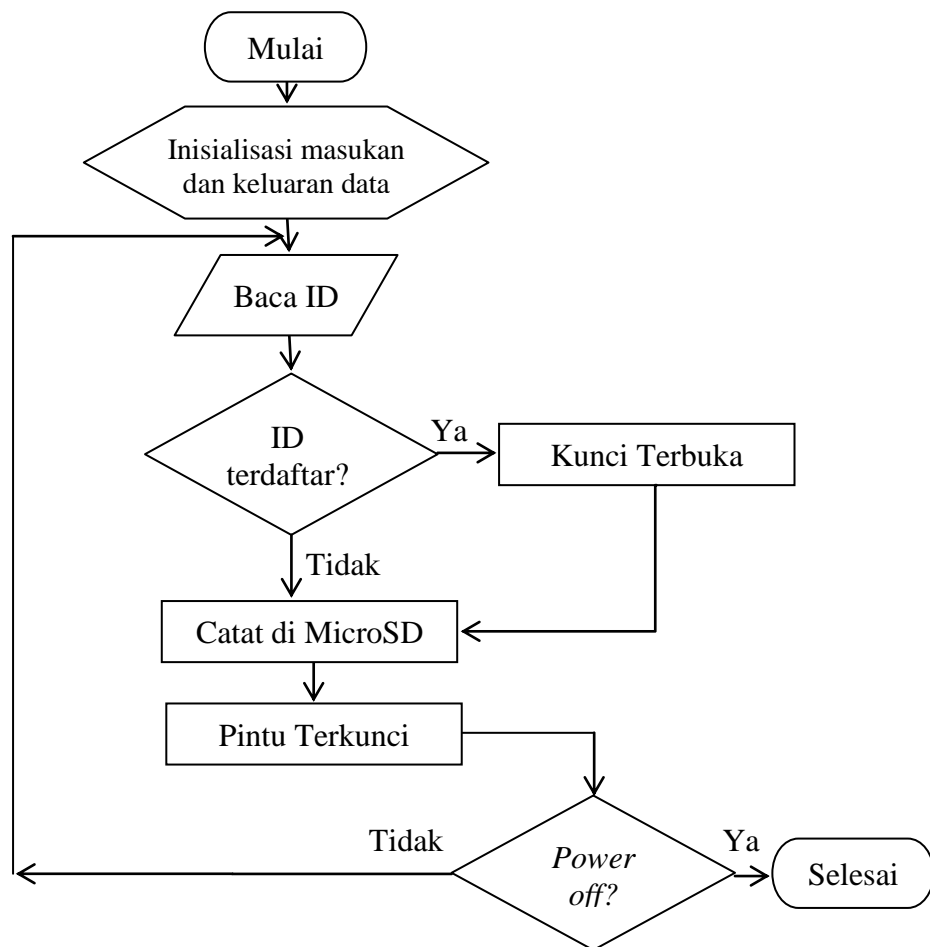
Pada bagian depan alat, kotak berwarna merah adalah letak NFC *reader*, dan kotak berwarna hitam merupakan *solenoid* sebagai kunci pintu. Untuk rancangan alat tampak dari belakang ditunjukkan oleh Gambar 3.4. dengan *limit switch* di bagian atas pintu, Arduino Uno dan *Ethernet Shield* pada kotak biru, serta 4 tombol.



**Gambar 3.4 Rancangan alat tampak belakang**

### 3.4. Perancangan Perangkat Lunak

Perancangan perangkat lunak digunakan sebagai dasar acuan untuk membuat program yang akan ditanamkan pada Arduino Uno. Arduino diprogram dengan menggunakan *software* IDE Arduino yang telah disediakan oleh web Arduino sendiri. IDE Arduino dapat melakukan *compile* dan unggah program sekaligus ke *board* Arduino. Pemrograman Arduino memiliki kemiripan dengan bahasa pemrograman C namun lebih sederhana. File program yang disimpan akan berbentuk \*.ino. Gambar 3.5 merupakan *flowchart* atau diagram alir dari rancangan program keseluruhan yang akan dibuat.

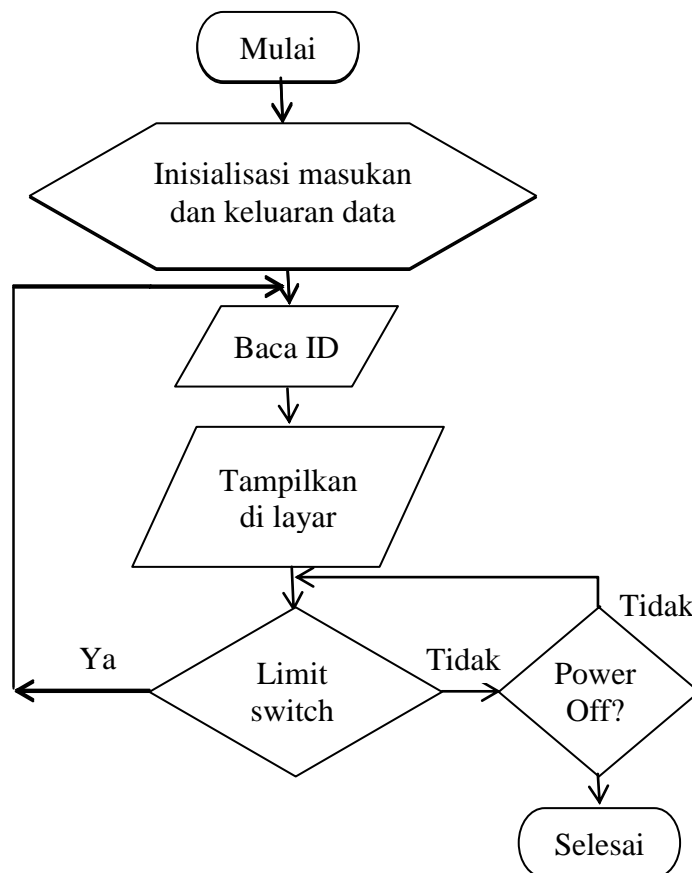


**Gambar 3.5** *Flowchart* program keseluruhan

Berdasarkan Gambar 3.5 *flowchart* diatas, maka perancangan perangkat lunak dibagi menjadi beberapa bagian seperti berikut ini:

### 3.4.1. Program Baca Tag NFC

Program ini berfungsi untuk menjalankan perintah membaca ID *tag* NFC yang kemudian ditampilkan di layar serial. Gambar 3.6 merupakan *flowchart* untuk program membaca ID *tag* NFC sebagai berikut:



**Gambar 3.6** *Flowchart* program baca tag NFC

Program ini dirancang untuk menjalankan perintah kepada NFC *reader* untuk membaca sebuah *tag* NFC yang berada pada jangkauan frekuensi yang sama. Setelah itu akan ditampilkan ID *tag* yang telah dibaca ke layar serial monitor.



Apabila ID sudah ditampilkan ke layar serial, maka *reader* dapat membaca *tag* lagi setelah limit switch diaktifkan. Program akan berakhir jika tidak ada *power supply* yang terhubung ke rangkaian. Berikut ini adalah potongan program baca *tag* NFC:

```
Success=nfc.readPassiveTargetID(PN532_MIFARE_ISO1444
3A, &uid[0], &uidLength);
    delay(10);
```

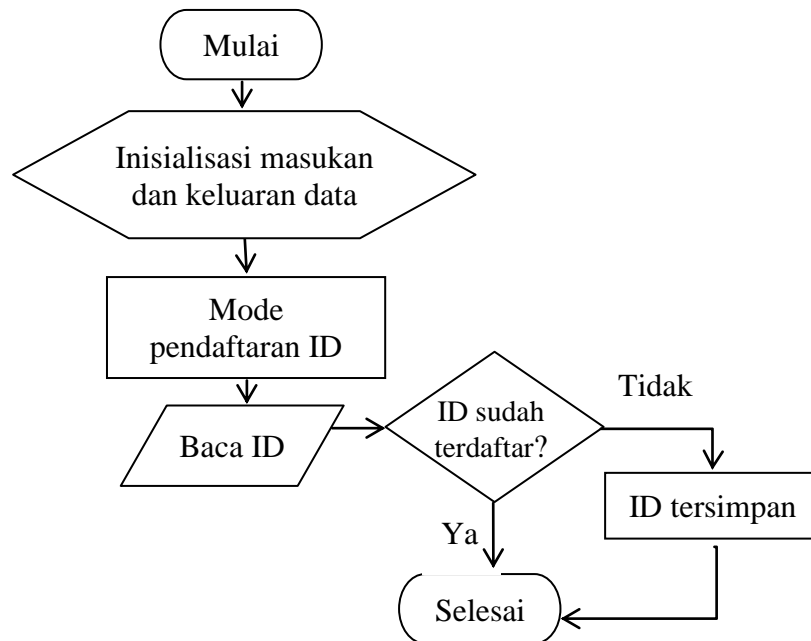
Potongan program diatas merupakan fungsi untuk membaca ID *tag* NFC yang disimpan pada variable *success*.

```
    if(success) {
data_nfc=bytetolong(uid[3],uid[2],uid[1],uid[0]);
Serial.print("ID: "); Serial.println(data_nfc,HEX);
```

Apabila ID telah dibaca oleh *reader* dan disimpan pada variable *success*, maka ID akan ditampilkan di layar serial dalam format *hexadecimal* dengan menggunakan fungsi *serial.print* seperti pada potongan program di atas.

### 3.4.2. Program Mendaftar dan Menghapus ID

Program ini dirancang untuk menjalankan perintah mendaftarkan ID *tag* NFC yang belum terdaftar di memori EEPROM dan menghapus ID yang telah terdaftar di memori. Gambar 3.7 merupakan *flowchart* program mendaftarkan ID di memori Arduino.



**Gambar 3.7 Flowchart program daftar ID**

Potongan program berikut ini adalah proses mendaftarkan ID baru di memori

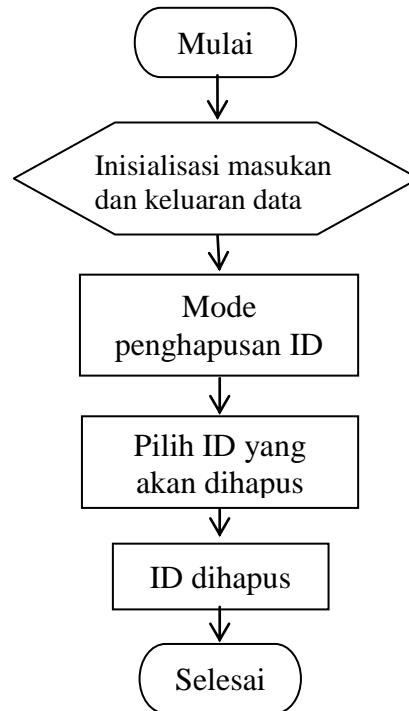
Arduino:

```

if (memory_nfc[i]==data_nfc) {Serial.println("Kartu
sudah terdaftar");break;}
if (memory_nfc[i]==0) {memory_nfc[i]=data_nfc;
EEPROMwritelong (i*4,memory_nfc[i]);
Serial.println("Pendaftaran Berhasil");
Serial.println();baca_eeprom();
  
```

Ketika program masuk ke mode pendaftaran ID, maka ID *tag* yang baru akan dibaca. Untuk kemudian ID diperiksa sudah terdaftar di memori Arduino atau belum. Jika belum terdaftar, maka ID akan disimpan pada memori Arduino dan akan ditampilkan pada layar serial. Jika ID sudah terdaftar, maka program mendaftarkan ID selesai.

Begitu juga halnya dengan mode penghapusan ID di memori, ketika mode penghapusan aktif maka dapat dipilih ID yang ingin dihapus. Berikut ini adalah *flowchart* program untuk menghapus ID di memori Arduino:



**Gambar 3.8 Flowchart program hapus ID**

Potongan program berikut ini merupakan proses memilih ID yang akan dihapus.

```

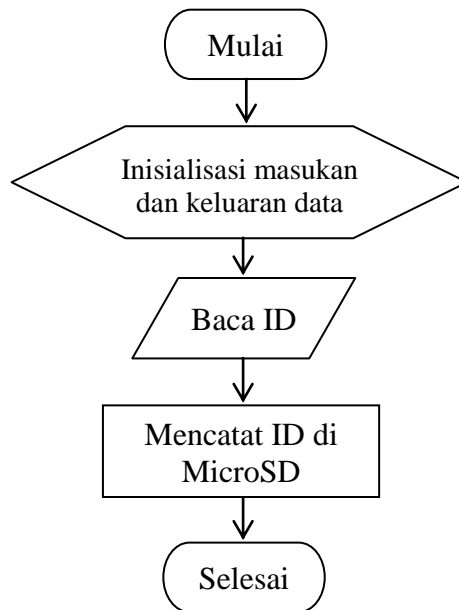
if(digitalRead(s2)==0) i++;
  if(digitalRead(s3)==0) i--;
  if(i<0) i=29;
  if(i>29) i=0;
  if(digitalRead(s2)==0 || digitalRead(s3)==0) {
  Serial.print("ID ");
  Serial.print(i);
  Serial.print(" : ");
  Serial.println(memory_nfc[i], HEX);
  }
  
```

Proses penghapusan ID hanya dapat dilakukan satu persatu dengan memilih ID yang akan dihapus. Setelah selesai dipilih ID yang akan dihapus, selanjutnya adalah proses menghapus ID dari memori Arduino dengan potongan program sebagai berikut:

```
if(digitalRead(s4)==0){
  Serial.print("ID ");
  Serial.print(i);
  Serial.print(" : ");
  Serial.println(memory_nfc[i],HEX);
  Serial.println("Terhapus !");
  memory_nfc[i]=0;
  EEPROMWritelong(i*4,memory_nfc[i]);
}
```

### 3.4.3. Program Mencatat Data di MicroSD

Program ini merupakan perintah untuk mencatat atau merekam ID yang mengakses pintu baik yang sudah terdaftar ataupun belum pada microSD. Pencatatan akses pintu ini menampilkan *real time* pada saat ID membuka kunci dan saat pintu terkunci kembali. Pencatatan akses pintu disimpan pada file \*.txt yang akan disimpan pada MicroSD. Gambar 3.9 adalah *flowchart* dari program mencatat data di MicroSD.



**Gambar 3.9 Flowchart program mencatat ID di MicroSD**

Setelah membaca ID, ID akan dicatat pada MicroSD dengan potongan program sebagai berikut:

```

dataFile.print("ID: ");
dataFile.println(data_nfc, HEX);
  
```

Selain ID yang dicatat pada MicroSD, dicatat juga waktu detail saat ID tersebut mengakses pintu dengan program sebagai berikut:

```

dataFile.println("Kunci Terbuka");
}
dataFile.print(jam);
dataFile.print(":");
dataFile.print(menit);
dataFile.print(":");
dataFile.print(detik);
dataFile.print(" ");
dataFile.print(font[hari]);
  
```

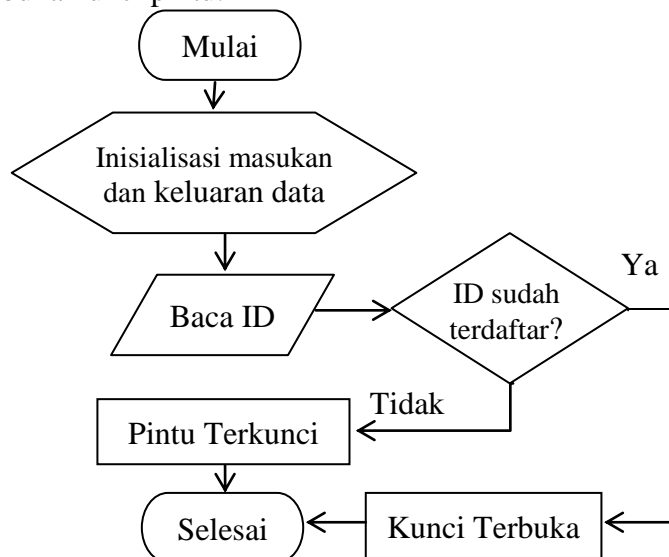
```

dataFile.print("/");
dataFile.print(tanggal);
dataFile.print("/");
dataFile.print(bulan);
dataFile.print("/");
dataFile.println(tahun+2000);
dataFile.println();
dataFile.close();

```

#### 3.4.4. Program Membuka Kunci Pintu

Program ini dirancang untuk membuka kunci pintu dengan mengaktifkan sebuah *solenoid* yang digunakan sebagai kunci pintu elektrik. Program ini berjalan setelah ada ID yang dibaca, kemudian diperiksa pada memori Arduino. Bila ID sudah terdaftar, maka *solenoid* aktif dan membuka kunci pintu. Namun jika belum terdaftar, *solenoid* dalam keadaan tidak aktif. Gambar 3.10 berikut ini adalah *flowchart* program membuka kunci pintu.



**Gambar 3.10** *Flowchart* program membuka kunci pintu

Potongan program berikut merupakan proses mengaktifkan *solenoid*:

```
if(data_nfc==memory_nfc[i]){  
flag_terbuka=1;digitalWrite(relay,LOW);Serial.println("  
Kunci Terbuka")
```

Namun jika ID tidak dapat membuka kunci pintu, maka potongan programnya adalah:

```
while(flag_terbuka==1){  
if(flag_terbuka==1&&flag_terbukal==0&&digitalRead(s_pin  
tu)==1)flag_terbukal=1;  
if(flag_terbuka==1&&flag_terbukal==1&&digitalRead(s_pin  
tu)==0){  
digitalWrite(relay,HIGH);  
Serial.println("Pintu Terkunci");
```