

BAB III

METODE PENELITIAN DAN PERANCANGAN SISTEM

3.1 Model Penelitian

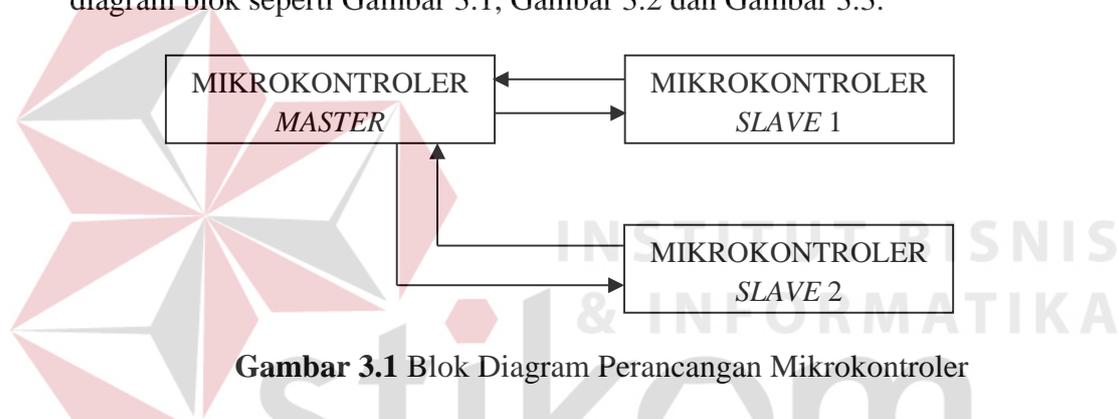
Pada perancangan tugas akhir ini menggunakan metode pemilihan *locker* secara otomatis. Sistem ini dibuat untuk mempermudah *user* dalam memilih *locker* yang belum terpakai oleh *user* lain. Perancangan sistem ini menggunakan 3 buah mikrokontroler, Arduino Uno sebagai *master* dan 2 buah mikrokontroler ATmega32 sebagai *slave*. Digunakan komunikasi I2C sebagai penghubung mikrokontroler Arduino Uno dengan masing-masing mikrokontroler ATmega32. Masing-masing mikrokontroler akan menjadi pusat kendali komponen *input* dan *output* yang terhubung pada mikrokontroler tersebut.

Sistem yang seluruh komponennya telah terintegrasi dengan baik dengan diawali dengan pendeteksian *tag* RFID oleh RFID *reader* untuk mendaftarkan *tag* RFID untuk menentukan *locker* mana yang bisa *user* tersebut gunakan. Setelah itu Arduino Uno akan memilih secara acak *locker* mana yang belum terpakai. Kemudian *user* akan diberikan informasi *locker* mana yang bisa digunakan setelah tombol *locker* ditekan kunci *locker* tersebut secara otomatis terbuka. Jika *user* ingin membuka *locker* tempat *user* menyimpan barang, maka *user* harus melakukan *scan tag* RFID lagi dan menekan tombol pada *locker*. Hal ini bertujuan agar *locker* tempat *user* menyimpan barang dapat terbuka. Jika *user* telah selesai menggunakan *locker* maka *user* harus melakukan *scan tag* RFID jika *user* telah selesai menggunakan *locker*. Pada perancangan terdapat 2 *locker* pada masing-masing mikrokontroler *slave*. Diawali dengan melakukan pengacakan *locker*

kosong yang dapat digunakan *user* oleh mikrokontroler *master*, kemudian *user* diberikan informasi *locker* mana yang bisa digunakan dari hasil pengacakan tersebut. Kemudian ketika seluruh *locker* telah terisi maka akan di laporkan kepada *user* bahwa seluruh *locker* penuh.

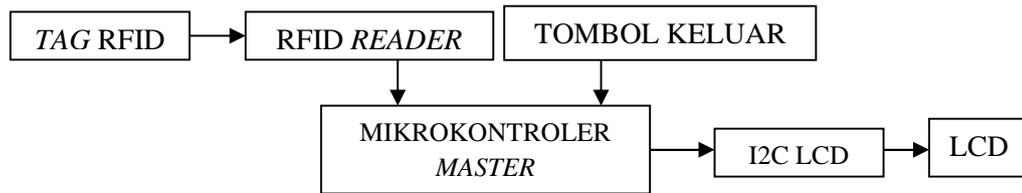
3.2 Model Perancangan

Perancangan sistem pada rancang bangun *locker* otomatis ini akan disajikan 3 blok diagram perancangan untuk mempermudah pemahaman dari perancangan *locker* otomatis. Blok diagram perancangan dapat dilihat dan dijelaskan pada diagram blok seperti Gambar 3.1, Gambar 3.2 dan Gambar 3.3.



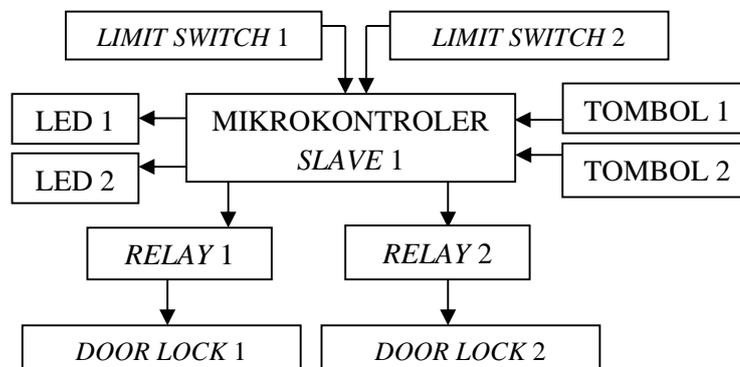
Gambar 3.1 Blok Diagram Perancangan Mikrokontroler

Dari Gambar 3.1 di atas dapat diperjelas bahwa digunakan 3 buah mikrokontroler, 1 buah mikrokontroler *master* dan 2 buah mikrokontroler sebagai *slave*. Mikrokontroler *master* pada perancangan menggunakan Arduino Uno dan Mikrokontroler *slave* menggunakan ATmega32. Mikrokontroler *master* akan saling berhubungan dengan mikrokontroler *slave*. Arti dari hubungan *input* dan *output* antara mikrokontroler *master* dengan mikrokontroler *slave* pada diagram blok adalah hubungan komunikasi I2C yaitu proses penerimaan dan pengiriman data. Pengiriman data dilakukan oleh mikrokontroler *master* ke mikrokontroler *slave*, atau sebaliknya.



Gambar 3.2 Blok Diagram Perancangan Mikrokontroler *Master*

Dari Gambar 3.2 dapat dijelaskan bahwa mikrokontroler *master* memiliki 2 buah komponen input dan 1 buah komponen output yang terhubung dengan mikrokontroler. Mikrokontroler *master* berfungsi untuk mengontrol seluruh komponen *input* dan *output* yang terhubung pada mikrokontroler *master* serta nantinya juga akan melakukan komunikasi dengan masing-masing mikrokontroler *slave*. Sistem diawali dengan *input* dari *tag* RFID yang dideteksi oleh RFID *Reader*. RFID *Reader* merupakan komponen *input* dari mikrokontroler *master* yang bertugas untuk melakukan scan ID. Kemudian terdapat tombol yang ditekan ketika *user* telah selesai menggunakan *locker* yang diikuti dengan dengan proses *input* dari *tag* RFID yang dideteksi oleh RFID *Reader*. I2C LCD merupakan komponen yang digunakan untuk komunikasi antara mikrokontroler *master* dengan LCD. LCD merupakan komponen *output* dari mikrokontroler *master* yang digunakan untuk memberi informasi kepada *user*.



Gambar 3.3 Blok Diagram Perancangan Mikrokontroler *Slave*

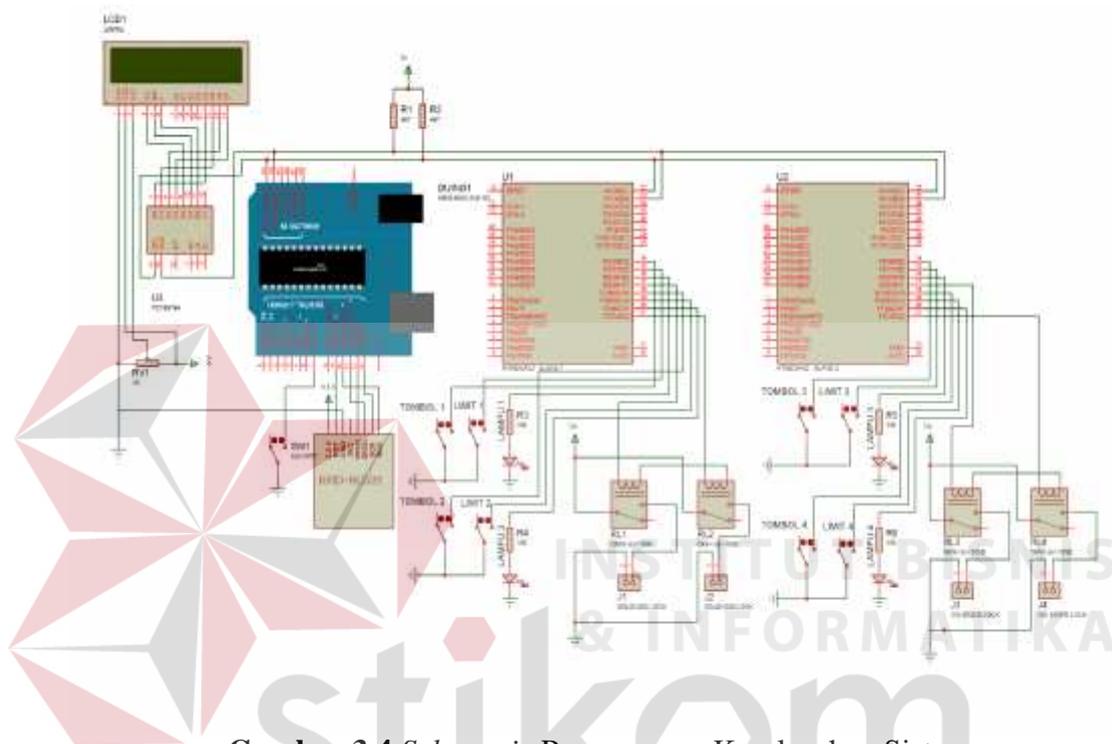
Gambar 3.3 menggambarkan bahwa *limit switch* merupakan komponen *input* dari masing-masing mikrokontroler *slave* yang berfungsi untuk mendeteksi kondisi *pintu locker* terbuka atau tertutup. Mikrokontroler *slave* berfungsi untuk mengontrol seluruh komponen *input*, *output* dan nantinya juga akan melakukan komunikasi dengan mikrokontroler *master*. Pada perancangan *limit switch* 1 dan *limit switch* 2 merupakan komponen *input* dari *slave* 1, sedangkan perancangan *limit switch* 3 dan *limit switch* 4 merupakan komponen *input* dari *slave* 2. Selain *limit switch*, tombol juga merupakan komponen *input*, pada perancangan tombol 1 dan tombol 2 terhubung dengan *slave* 1, sedangkan tombol 3 dan tombol 4 terhubung dengan *slave* 2. Kemudian komponen *output* pada mikrokontroler *slave* adalah led dan *relay* yang terhubung dengan *door lock*. Led pada mikrokontroler *slave* berfungsi sebagai penanda pada *locker* dan tombol yang berfungsi untuk membuka kunci *locker*.

Relay pada mikrokontroler *slave* berfungsi sebagai saklar otomatis untuk menghidupkan *door lock*. *Door lock* pada mikrokontroler *slave* berfungsi untuk mengunci *pintu*. Pada perancangan led 1, led 2, *relay* 1 yang terhubung dengan *door lock* 1 dan *relay* 2 yang terhubung dengan *door lock* 2 merupakan komponen-komponen yang terhubung dengan *slave* 1, sedangkan led 3, led 4, *relay* 3 yang terhubung dengan *door lock* 3 dan *relay* 4 yang terhubung dengan *door lock* 4 merupakan komponen-komponen yang terhubung dengan *slave* 2.

3.3 Perancangan Perangkat Keras

Perancangan tugas akhir ini diawali dengan melakukan perancangan perangkat keras yang menjadi satu buah sistem yang saling terintegrasi. Perancangan terdiri dari perancangan mikrokontroler *master*, perancangan

mikrokontroler *slave*, perancangan komunikasi I2C, perancangan RFID, perancangan LCD dan perancangan komponen *slave*. Pada Gambar 3.4 dapat dilihat *Schematic* perancangan seluruh sistem pada rancang bangun *locker* otomatis.



Gambar 3.4 *Schematic* Perancangan Keseluruhan Sistem

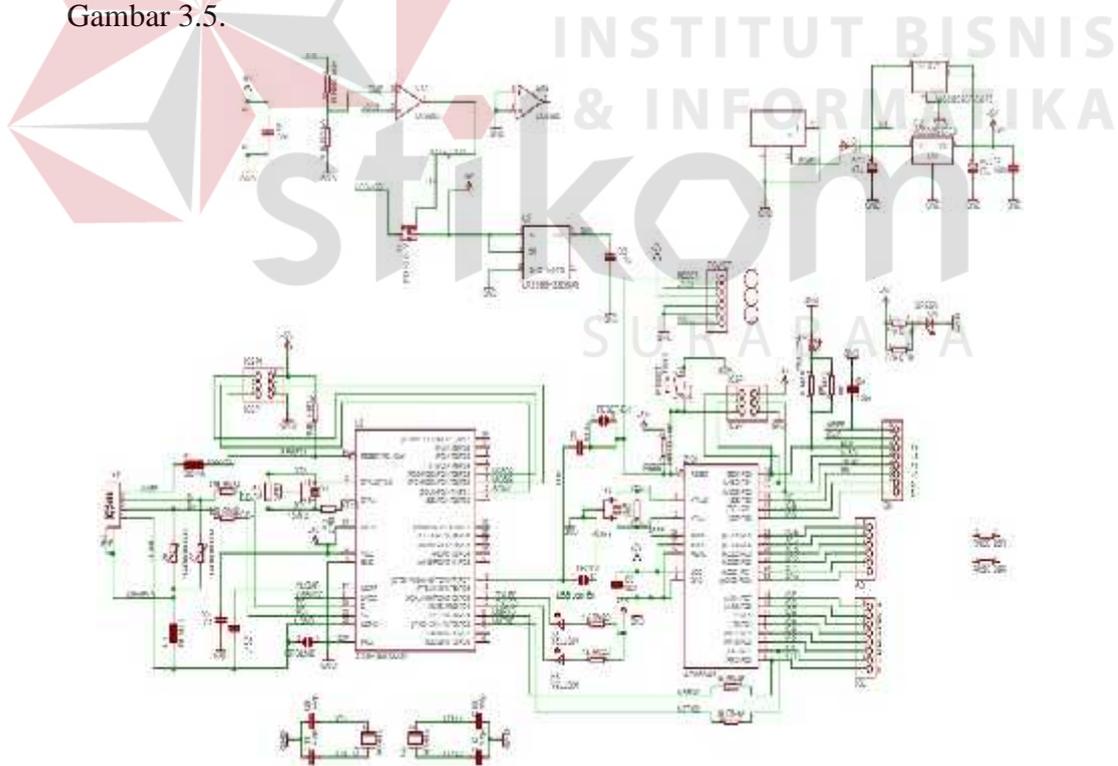
3.3.1. Perancangan Mikrokontroler *Master*

Mikrokontroler Arduino Uno adalah mikrokontroler yang digunakan sebagai *master* pada perancangan. Mikrokontroler Arduino Uno merupakan sistem elektronika sederhana yang terdiri dari komponen-komponen yang dirangkai menjadi satu dan sudah menjadi modul mikrokontroler. Perancangan arduino dibuat dengan menentukan *port* atau *pin* pada arduino yang akan digunakan untuk mengakses semua komponen. Arduino Uno merupakan sistem mikrokontroler yang menggunakan IC ATmega328. Karena Arduino Uno sudah dalam bentuk modul IC pada arduino ada yang telah tertanam atau terhubung pada

modul dan ada yang dapat dilepas atau diganti ketika IC mengalami kerusakan. Mikrokontroler Arduino Uno memiliki beberapa fitur utama, adapun fitur utama dari mikrokontroler Arduino Uno adalah sebagai berikut:

- 6 Pin input Analog Digital Converter (ADC), berada pada port A0, A1, A2, A3, A4 dan A5.
- 6 channel Pulse Width Modulation (PWM), berada pada port 3, 5, 6, 9, 10 dan 11.
- 14 channel I/O digital, berada pada port 1 sampai 13.
- Menggunakan tegangan 5 Volt untuk beroperasi.
- Mempunyai 1 pin supply 5 Volt dan 1 pin supply 3.2 Volt.

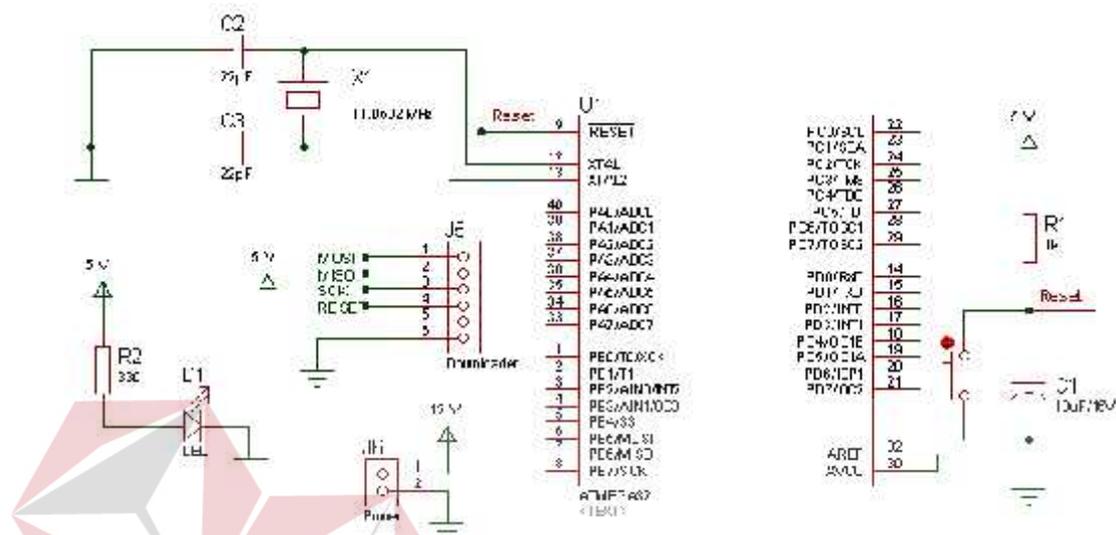
Berikut adalah Gambar *schematic* Arduino Uno yang ditunjukkan oleh Gambar 3.5.



Gambar 3.5 Schematic Mikrokontroler Arduino Uno

Pengujian pada mikrokontroler Arduino Uno dilakukan dengan menguji apakah program berhasil atau tidak di *upload* pada mikrokontroler Arduino Uno.

3.3.2. Perancangan Mikrokontroler *Slave*



Gambar 3.6 Schematic Mikrokontroler ATmega32

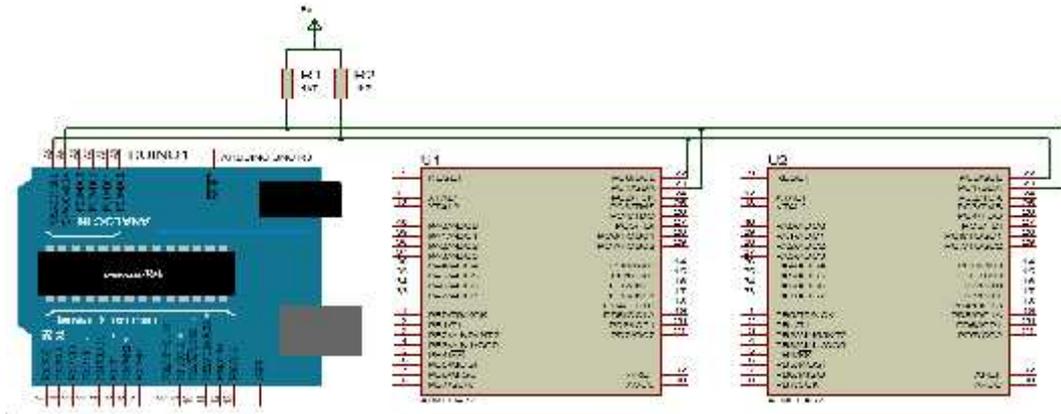
Pada perancangan mikrokontroler yang digunakan sebagai *slave* adalah mikrokontroler ATmega32. Mikrokontroler ATmega32 adalah sistem elektronika sederhana yang terdiri dari komponen-komponen dasar yang dirangkai menjadi satu karena dibutuhkan suatu mikrokontroler agar dapat berfungsi dengan baik. Suatu mikrokontroler membutuhkan dua komponen tambahan selain power untuk dapat berfungsi. Rangkaian tersebut adalah Kristal *Oscillator* (XTAL) dan rangkaian reset, kedua rangkaian tersebut pada umumnya selalu ada pada minimum sistem mikrokontroler. Rangkaian Kristal *Oscillator* (XTAL) merupakan bagian komponen yang sangat penting pada mikrokontroler ATmega32 karena berfungsi untuk memompa data yaitu bersifat timer (semacam *clock*)/pulsa digital oleh karena itu kristal memiliki sebuah frekuensi dengan

satuan MHz (*MegaHertz*) dengan perumpamaan jantung pada manusia. Rangkaian Kristal *Oscillator* (XTAL) ini terdiri dari sebuah kristal dan dua buah kapasitor.

Rangkaian reset pada mikrokontroler ATmega32 adalah aktif low, berfungsi untuk mikrokontroler memulai kembali pembacaan program, hal tersebut dibutuhkan pada saat mikrokontroler mengalami gangguan dalam mengeksekusi program. Rangkaian ini terdiri dari 1 buah tombol, 1 buah kapasitor dan 1 buah resistor yang dihubungkan pada *pin* reset ATmega32. Rangkaian tambahan lain yang digunakan pada minimum sistem terutama yang digunakan pada perancangan ini adalah rangkaian led indikator dan konektor ISP (*In System Chip Program*) untuk *upload* program ke mikrokontroler. Pengujian pada mikrokontroler ATmega32 dilakukan dengan menguji apakah program berhasil atau tidak di *upload* pada mikrokontroler ATmega32.

3.3.3. Perancangan Komunikasi I2C

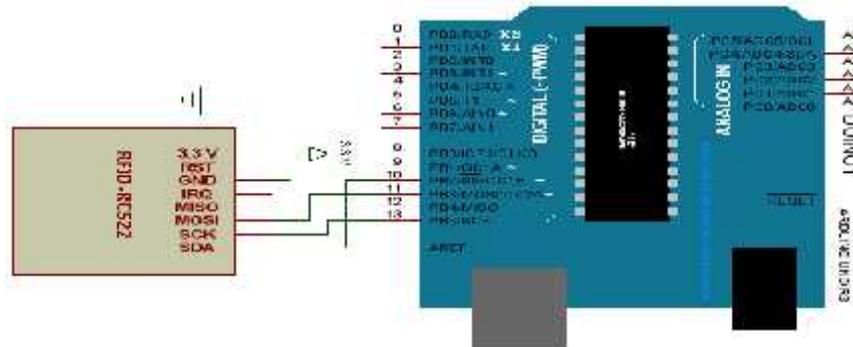
Komunikasi I2C pada perancangan digunakan agar tiap mikrokontroler dapat saling terhubung. Masing-masing mikrokontroler ATmega32 sebagai *slave* akan dihubungkan pada mikrokontroler Arduino Uno sebagai *master*. *Pin* SDA pada mikrokontroler saling terhubung antara mikrokontroler *slave* dan mikrokontroler *master*, begitu juga dengan *pin* SCL. Agar antara mikrokontroler *slave* dan mikrokontroler *master* dapat saling berkomunikasi, maka di tambahkan rangkain *pull up* dengan resistor 4,7 K Ohm. Pengujian perancangan ini dilakukan dengan melakukan proses pengiriman data dan melihat hasil pengiriman data apakah berhasil atau tidak. Perancangan komunikasi I2C ini, *master* dapat mengontrol *slave* sebanyak 256 buah *device*. Perancangan komunikasi I2C dapat dilihat pada Gambar 3.7.



Gambar 3.7 Schematic Perancangan Komunikasi I2C

3.3.4. Perancangan RFID

RFID yang digunakan pada perancangan adalah sensor RFID RC-522 berfungsi untuk membaca tag RFID yang akan dibaca oleh Arduino Uno. RFID ini merupakan komponen *input* dari mikrokontroler *master* dan merupakan proses *input* awal dari sistem *locker* otomatis ini. *Input* tegangan dari RFID RC-522 adalah 3,3 Volt. RFID RC-552 terdiri dari 8 pin, pada perancangan hanya menggunakan 7 pin karena pin IRQ tidak dibutuhkan untuk terhubung pada Arduino Uno. Dalam perancangan ini, RFID dihubungkan dengan port D9 – D13. Adapun perancangan RFID dengan Arduino Uno dapat dilihat pada Gambar 3.8.



Gambar 3.8 Schematic Perancangan RFID

Pada Tabel 3.1 dapat dilihat *allocation list* dari perancangan RFID yang dihubungkan dengan Arduino Uno :

Tabel 3.1 Perancangan RFID RC-522 Dengan Arduino Uno

No	PORT RFID RC-522	PORT Arduino Uno
1.	SDA	Digital 10
2.	SCK	Digital 13
3.	MOSI	Digital 11
4.	MISO	Digital 12
5.	IRQ	-
6.	GND	GND
7.	RST	Digital 9
8.	3,3 V	3,3 V

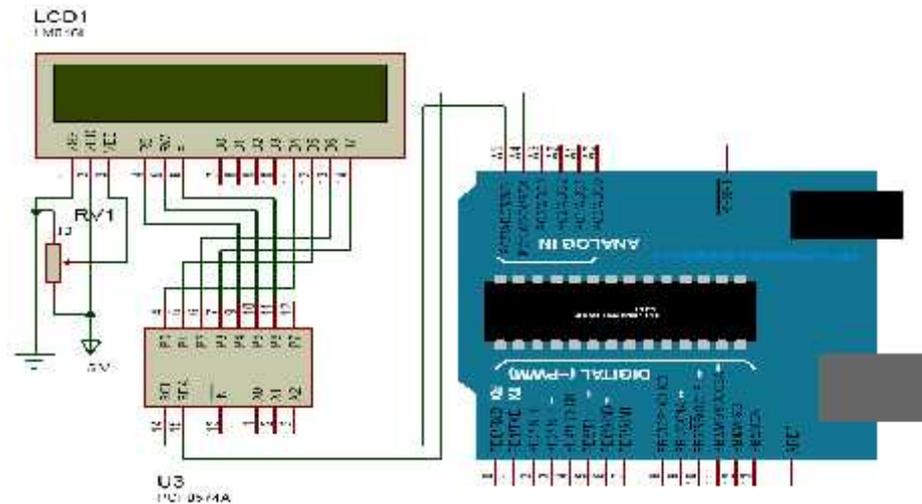
Pengujian RFID dilakukan dengan cara melihat hasil proses *scan tag* RFID, apakah ID dari *tag* RFID dapat terdeteksi atau tidak.

3.3.5. Perancangan LCD (*Liquid Crystal Display*)

LCD yang digunakan untuk sistem ini adalah *liquid* LCD dengan ukuran 16 x 2 yang menggunakan jenis komunikasi I2C. Pada LCD ini terdapat dua *output* yaitu *pin* SDA dan *pin* SCL yang merupakan *pin output* dari jenis komunikasi I2C. Pada sistem ini LCD digunakan sebagai informasi untuk pemakaian daya yang telah terpakai. Pada Tabel 3.2 dapat dilihat penjelasan dari *pin-pin display* LCD I2C dan pada Gambar 3.9 dapat dilihat bagaimana *Schematic Perancangan LCD (Liquid Crystal Display)*.

Tabel 3.2 Penjelasan *Pin-Pin Display* LCD I2C

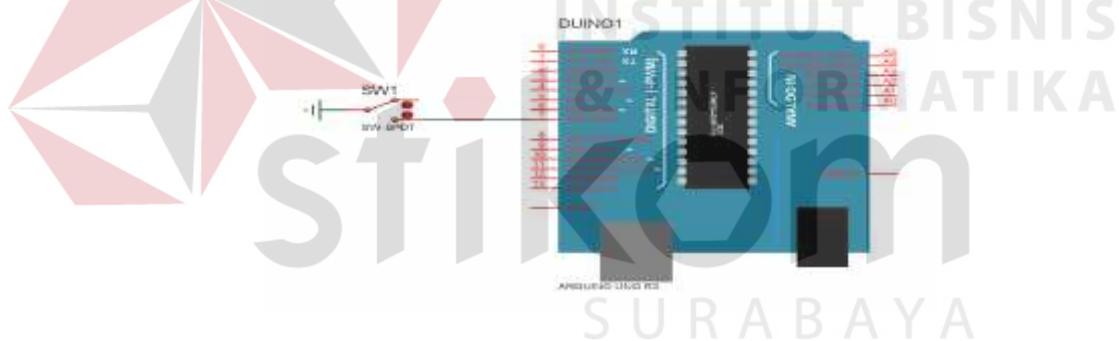
Pin	Symbol	External connection	Function
1	V _{SS}	Power supply	Signal ground for LCM (GND)
2	V _{DD}		Power supply for logic (+5V) for LCM
3	V ₀		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7-10	DB0-DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11-14	DB4-DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL "A" (5V)
16	LED-		Power supply for BKL "K" (GND)



Gambar 3.9 Schematic Perancangan LCD (*Liquid Crystal Display*)

Pengujian LCD dilakukan dengan cara melihat apakah LCD dapat menampilkan karakter yang sesuai dengan program yang diinginkan.

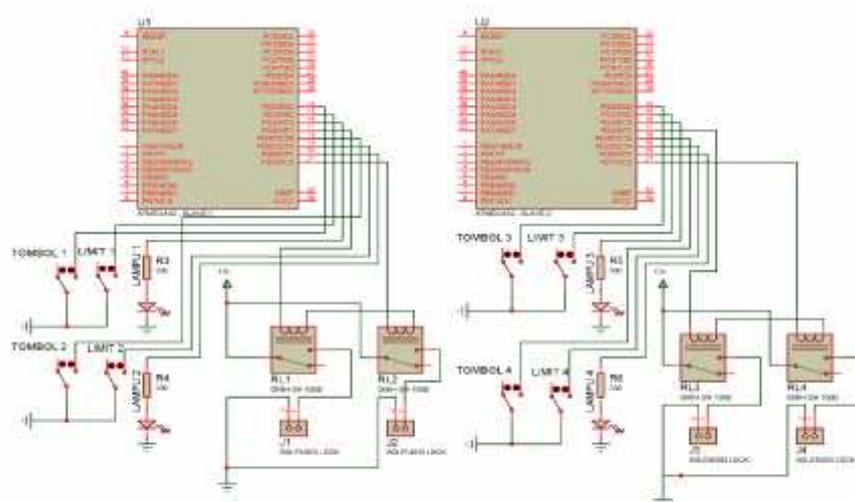
3.3.6. Perancangan Tombol Pada Mikrokontroler *Master*



Gambar 3.10 Schematic Perancangan Tombol Pada Mikrokontroler *Master*

Pada Gambar 3.10 dapat di analisa tombol pada mikrokontroler *master* dihubungkan pada *pin* D7. Bentuk rangkaian tombol adalah salah satu *pin* dihubungkan pada *pin* mikrokontroler dan *pin* yang lain dihubungkan pada *input ground*. Fungsi komponen ini adalah sebagai penanda bahwa *user* telah selesai menggunakan *locker* dan mengambil barang pada *locker*. Pada saat proses penekanan tombol pada mikrokontroler *master*, akan dilakukan juga proses scan RFID untuk menganalisa bahwa ID dari *tag* RFID telah terdaftar.

3.3.7. Perancangan Komponen *Slave*



Gambar 3.11 Schematic Perancangan Komponen *Slave*

Dari Gambar 3.11 dapat di analisa bahwa pada bagian mikrokontroler *slave* terdapat komponen *input* dan komponen *output*. Komponen *input* yang terhubung pada mikrokontroler *slave* adalah *limit switch*. Kemudian komponen *output* yang terhubung pada mikrokontroler *slave* adalah led, *relay* yang terhubung dengan *door lock* dan tombol. *Limit switch* ada sebanyak 4 buah, 2 buah dihubungkan dengan *slave* 1 dan 2 buah dihubungkan dengan *slave* 2. Komponen-komponen *output* juga ada sebanyak masing-masing 4 buah, 2 buah led terhubung dengan *slave* 1 dan 2 buah led terhubung dengan *slave* 2. Kemudian 2 buah rangkaian *relay* beserta *door lock* terhubung dengan dengan *slave* 1 dan 2 buah lagi terhubung dengan *slave* 2. Begitu juga dengan tombol, 2 buah tombol terhubung dengan *slave* 1 dan 2 buah lagi terhubung dengan *slave* 2. Dari Tabel 3.3 dapat dilihat rincian *allocation list* komponen-komponen *slave* pada perancangan.

Tabel 3.3 Allocation List Komponen Slave

No	PORT	Mikrokontroler	Komponen
1.	PORTD.1	Slave 1	Limit Switch 1
2.	PORTD.5	Slave 1	Limit Switch 2
3.	PORTD.1	Slave 2	Limit Switch 3
4.	PORTD.5	Slave 2	Limit Switch 4
5.	PORTD.3	Slave 1	Relay 1
6.	PORTD.7	Slave 1	Relay 2
7.	PORTD.3	Slave 2	Relay 3
8.	PORTD.7	Slave 2	Relay 4
9.	PORTD.0	Slave 1	Tombol 1
10.	PORTD.4	Slave 1	Tombol 2
11.	PORTD.0	Slave 2	Tombol 3
12.	PORTD.4	Slave 2	Tombol 4
13.	PORTD.2	Slave 1	Led 1
14.	PORTD.6	Slave 1	Led 2
15.	PORTD.2	Slave 2	Led 3
16.	PORTD.6	Slave 2	Led 4

3.4 Perancangan Mekanik

Selain perancangan *hardware*, dilakukan juga perancangan mekanik dari rancang bangun *locker* otomatis yang tentunya berguna untuk pengujian keseluruhan sistem pada kondisi yang dibutuhkan sesuai dengan ide perancangan. Perancangan terdiri dari perancangan *locker*, perancangan mikrokontroler, perancangan RFID, perancangan LCD dan perancangan komponen *slave*. Berikut pembahasan dari perancangan mekanik dari *locker* otomatis.

3.4.1 Perancangan Locker

Pada perancangan ini, *locker* dibuat dari bahan kayu yang menyesuaikan pembuatan *locker* pada umumnya. Dirancang 4 buah *locker* yang nantinya 2 buah akan di kontrol oleh masing-masing mikrokontroler *slave*. *Locker* 1 dan 2 akan di kontrol oleh *slave* 1 kemudian *locker* 3 dan 4 akan di kontrol oleh *slave* 2. *Locker* dirancang dengan bentuk 2 *locker* di bagian atas dan 2 *locker* di bagian bawah. Bentuk *locker* yang di gunakan pada rancang bangun untuk bagian luar *locker*

dapat dilihat pada Gambar 3.12 dan bagian dalam *locker* dapat dilihat pada Gambar 3.13 berikut.



Gambar 3.12 Bagian Luar *Locker*



Gambar 3.13 Bagian Dalam *Locker*

3.4.2 Perancangan Mikrokontroler

Mikrokontroler-mikrokontroler yang digunakan pada perancangan dibuatkan 1 buah *box* untuk meletakkan komponen-komponen mikrokontroler. *Box* ini akan diletakkan di samping *locker* sebelah kiri. Dengan *box* ini perancangan mikrokontroler akan lebih terlihat rapi. *Allocation list* dari perancangan mikrokontroler pada rancang bangun *locker* otomatis dapat dilihat pada Tabel 3.4.

Tabel 3.4 *Allocation List* Perancangan Mikrokontroler

No	Nama Pin	PORT Master	PORT Slave 1	PORT Slave 2
1.	SDA	A4	PORTC.1	PORTC.1
2.	SCL	A5	PORTC.0	PORTC.0

3.4.3 Perancangan RFID

RFID pada perancangan diletakkan pada *box* yang telah dibuat yang juga merupakan tempat komponen dan mikrokontroler yang digunakan pada

perancangan. RFID diletakkan pada tutup *box* dengan tujuan untuk memudahkan *user* ketika akan melakukan *scan tag* RFID pada *RFID Reader*. Seperti telah dijelaskan sebelumnya RFID akan dihubungkan dengan mikrokontroler *master*, sehingga jika RFID di letakkan pada *box* maka jarak RFID dengan mikrokontroler *master* berdekatan dan tidak membutuhkan kabel panjang.

3.4.4 Perancangan LCD (*Liquid Crystal Display*)

Pada perancangan LCD juga diletakkan di atas *box* yang telah dibuat, sama dengan tempat meletakkan RFID. Tujuan LCD diletakkan di atas *box* untuk memudahkan *user* melihat informasi yang ditampilkan LCD sesuai dengan program yang diinginkan. Sedangkan komponen I2C LCD diletakkan di dalam *box*, sama dengan mikrokontroler-mikrokontroler yang digunakan pada perancangan. Tujuan LCD dan I2C LCD diletakkan pada *box* agar pengaturan komponen lebih rapi dan tidak membutuhkan kabel panjang.

3.4.5 Perancangan Tombol Pada Mikrokontroler *Master*

Tombol yang terhubung dengan mikrokontroler *master* dengan fungsi sebagai penanda bahwa *user* telah selesai menggunakan *locker* diletakkan pada *box* yang sama dengan tempat komponen mikrokontroler. Tombol diletakkan berdekatan dengan RFID agar mempermudah *user* untuk melakukan proses penekanan tombol dan *scan tag* RFID pada saat yang bersamaan.

3.4.6 Perancangan Komponen *Slave*

Komponen-komponen dari *slave* diletakkan langsung pada bagian *locker*, di dalam *locker* dan di luar *locker*. Selain pada bagian *locker* komponen *slave* juga ada yang diletakkan pada *box*. Komponen yang ada di luar adalah led dengan rangkaian led berada di dalam *locker* dan tombol-tombol yang terhubung pada

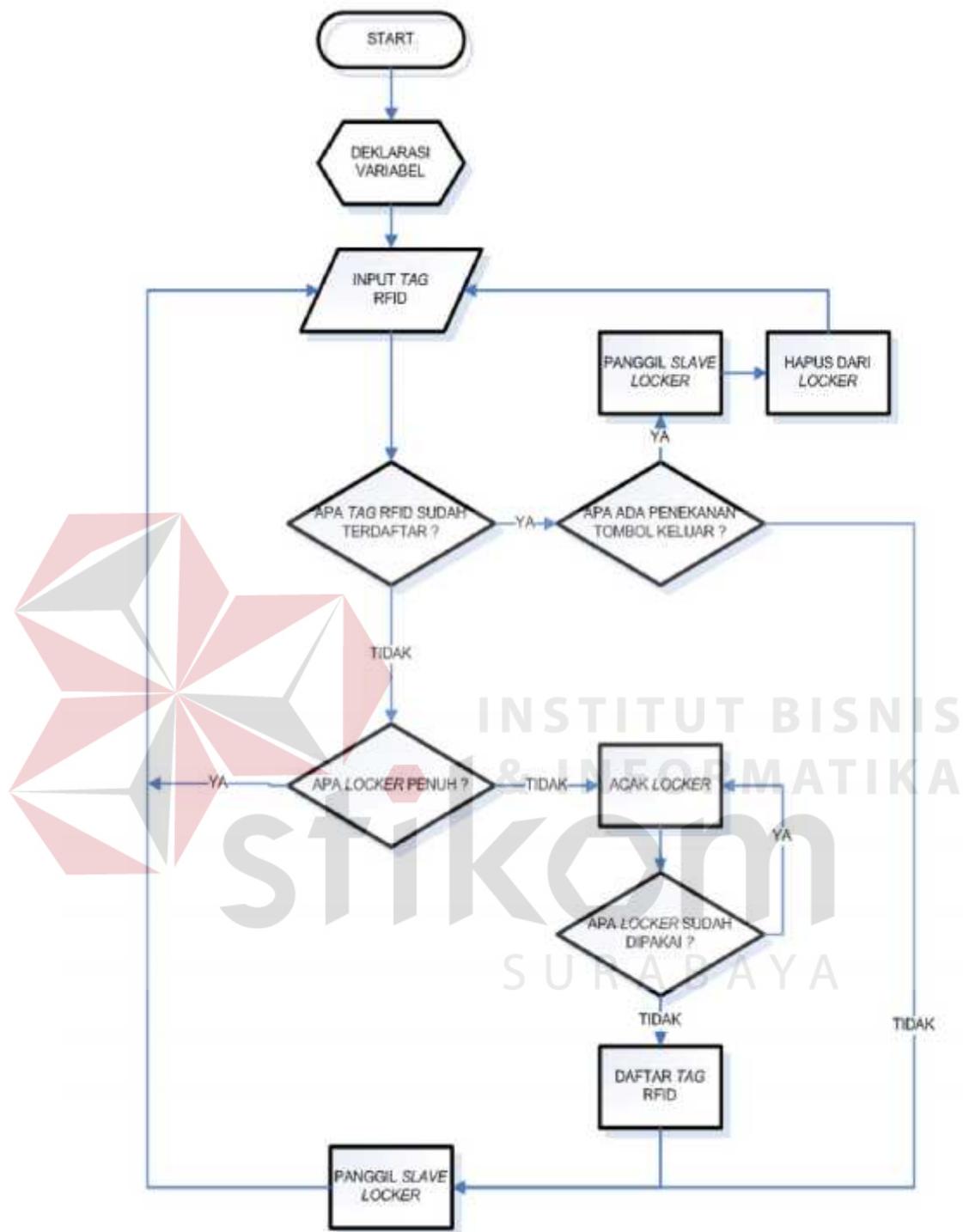
mikrokontroler *slave*, tepatnya diletakkan di *pintu locker* yang juga dapat dilihat pada Gambar 3.12. Kemudian pada Gambar 3.13 dapat di analisa komponen-komponen *slave* yang diletakkan pada bagian dalam *locker*. *Limit switch* sebagai komponen *input* dari mikrokontroler *slave*, diletakkan di atas pada bagian dalam *locker*, tepatnya di dinding-dinding penyekat *locker* 1 dengan 2 dan *locker* 3 dengan 4. Kemudian *relay* diletakkan di bagian belakang *locker*. *Relay* ini dihubungkan dengan *door lock* yang berguna sebagai *switch* untuk mengalirkan atau memutus tegangan *input* pada *door lock*. *Relay* akan membantu dalam proses ketika memutuskan *door lock* aktif atau tidak, terkunci atau tidak. *Door lock* pada perancangan diletakkan pada *pintu locker*, dengan menggunakan kabel panjang untuk terhubung dengan *relay* yang diletakkan pada *box* di luar *locker*.

3.5 Perancangan Perangkat Lunak

Selain perancangan perangkat keras dan perancangan mekanik, dibutuhkan juga perancangan perangkat lunak. Perancangan ini bertujuan agar sistem berjalan sesuai dengan yang diinginkan. Karena menggunakan 2 jenis fungsi mikrokontroler, yaitu mikrokontroler *master* dan mikrokontroler *slave*, maka dirancang perancangan program *master* dan perancangan program *slave*.

3.5.1 Perancangan Program Mikrokontroler Master

Program *master* pada perancangan akan mengontrol komponen *output* dan *input* dari mikrokontroler *master*. Komponen tersebut adalah RFID sebagai *input* dan LCD I2C sebagai *output* dan melakukan komunikasi I2C dengan mikrokontroler *slave*. *Flowchart* dari program mikrokontroler *master* dapat dilihat pada Gambar 3.14 berikut.



Gambar 3.14 Flowchart Program Mikrokontroler Master

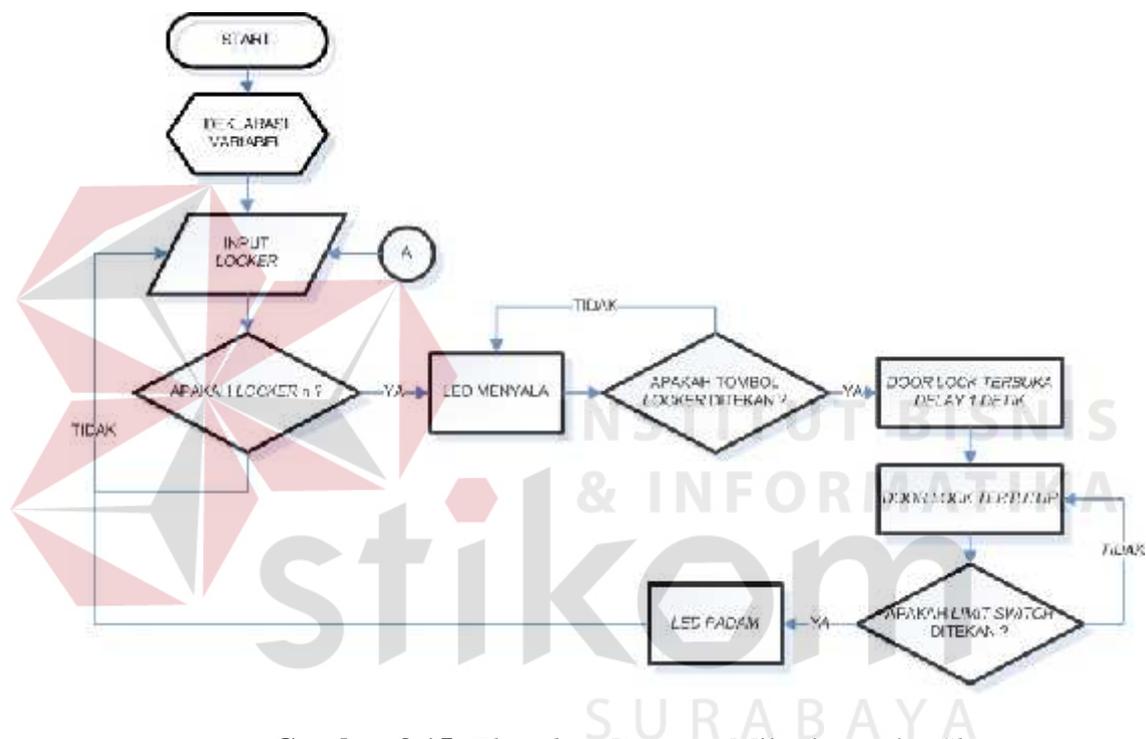
Dari Gambar 3.14 dapat di analisa bagaimana proses yang di kontrol oleh mikrokontroler master atau flow chart dari mikrokontroler master. Program di

awali dari deklarasi variabel *input* dan *output* yang dibutuhkan. Kemudian lakukan proses scan *Tag* RFID pada proses *input Tag* RFID. Setelah *input* berhasil, dilakukan pengecekan *Tag* RFID yaitu ID dari *Tag* RFID. Selanjutnya cek kondisi apakah ID pada *Tag* RFID telah terdaftar, jika ID telah terdaftar maka dilakukan cek kondisi apakah ada penekanan tombol untuk membuka *locker*. Ketika ID telah terdaftar dan ada penekanan tombol maka mikrokontroler *master* akan berkomunikasi dengan mikrokontroler *slave* untuk memberikan tanda nomor *locker* dari ID yang terdaftar telah selesai di gunakan kemudian dilakukan penghapusan data nomer dan ID *locker* dan kembali pada proses *input tag* RFID. Namun ketika ID telah terdaftar dan tidak ada penekanan tombol maka mikrokontroler *master* akan berkomunikasi dengan mikrokontroler *slave* agar kunci *locker* tempat penyimpanan barang dari *Tag* RFID yang telah terdaftar tersebut dapat terbuka, kemudian kembali pada poses *input ID Tag* RFID.

Namun setelah proses cek kondisi apakah ID pada *Tag* RFID telah terdaftar dan hasilnya belum terdaftar, maka akan dilakukan cek kondisi *locker* dalam keadaan penuh atau tidak. Jika kondisi *locker* dalam keadaan penuh maka proses akan kembali pada proses *input tag* RFID dan cek ID *tag* RFID terus-menerus sampai ada *locker* yang kosong. Jika kondisi *locker* dalam keadaan tidak penuh, lalu dilakukan pengacakan nomer *locker*. Selanjutnya cek kondisi apakah *locker* telah terpakai, jika telah terpakai dilakukan pengacakan nomer *locker* kembali, jika belum terpakai maka ID yang belum terdaftar akan didaftarkan, kemudian mikrokontroler *master* akan berkomunikasi dengan mikrokontroler *slave* agar kunci dari hasil acak nomer *locker* dapat terbuka, kemudian kembali lagi pada proses *input ID Tag* RFID.

3.5.2 Perancangan Program Mikrokontroler *Slave*

Program *slave* pada perancangan akan mengontrol komponen *output* dan *input* dari mikrokontroler *slave*. Komponen tersebut adalah *limit switch* sebagai *input*, kemudian *relay* yang terhubung dengan *door lock* dan led sebagai *output* serta melakukan komunikasi I2C dengan mikrokontroler *master*. *Flowchart* dari program mikrokontroler *slave* dapat dilihat pada Gambar 3.15 berikut.



Gambar 3.15 *Flowchart* Program Mikrokontroler *Slave*

Proses yang di kontrol oleh mikrokontroler *slave* atau *flowchart* dari mikrokontroler *slave* dapat dilihat pada Gambar 3.15. Program diawali dari deklarasi variabel *input* dan *output* dari mikrokontroler *slave*. Sistem diawali dengan proses input nomer *locker* dari mikrokontroler *master*. Huruf n pada Gambar 3.15 menandakan nomer *locker*. Selanjutnya cek kondisi apakah nomer *locker* yang dikirim dari mikrokontroler *master* adalah *locker* nomer n, jika iya maka led akan menyala. Selanjutnya cek kondisi apakah tombol untuk membuka *locker* telah ditekan, jika tidak maka proses akan terus menyalakan led dan

melakukan cek kondisi tombol sampai tombol ditekan. Jika tombol telah ditekan maka *door lock* akan terbuka, setelah itu *delay* selama 1 detik, selanjutnya *door lock* akan tertutup atau terkunci kembali. Lalu cek kondisi apakah *limit switch* tertekan, jika tidak maka *door lock* akan mengulang proses menutup *door lock* kembali sampai *limit switch* tertekan. Jika *limit switch* tertekan yang berarti *pintu locker* telah ditutup maka led pada *locker* n akan padam dan kembali pada proses *input* nomer *locker* dari mikrokontroler *master*.

