

BAB IV

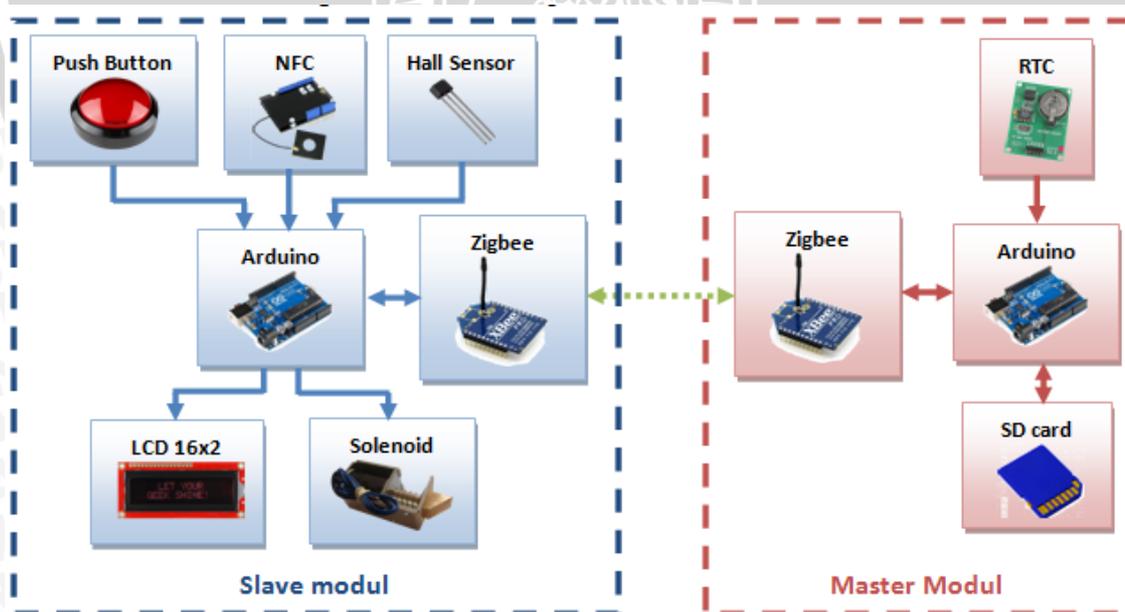
PERANCANGAN DAN PEMBUATAN ALAT

Perancangan sistem akses kontrol kunci pintu elektrik ini dilakukan secara bertahap sehingga akan memudahkan analisis pada setiap bloknnya maupun secara keseluruhan. Perancangan ini terdiri dari beberapa hal sebagai berikut :

- 1). Perancangan Sistem akses kontrol kunci pintu elektrik
- 2). Perancangan perangkat keras (*hardware*) terdiri dari rangkaian antarmuka mikrokontroler dan modul NFC, *magnetic switch*, rangkaian LCD, serta sistem mekanik purwarupa pintu.
- 3). Perancangan perangkat lunak (*software*) terdiri dari pengolah dan pengiriman UID e-KTP, Pengiriman data Xbee, serta pemilahan dan penyimpanan presensi pada *database*.

4.1 Perancangan Sistem

Sistem sistem akses kontrol kunci pintu elektrik secara umum terbagi menjadi dua bagian utama yaitu sistem pembaca e-KTP melekat pada pintu (*slave*) dan sistem penerima dan *logger* (*master*). Diagram blok masing-masing sistem ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram Blok Sistem

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

1. Blok Sistem Pembaca e-KTP

Blok sistem pembaca e-KTP terdiri dari modul NFC, mikrokontroler Arduino, *magnetic switch*, LCD dan *solenoid*. Berikut adalah fungsi dari masing-masing bagian blok sistem pembaca e-KTP/pengirim:

a. Mikrokontroler Arduino (*slave*)

Berfungsi untuk mengolah UID (*Unique Identification*), mengirimkan dan menerima umpan balik dari mikrokontroler Arduino (*master*).

b. Modul NFC

Berfungsi membaca UID (*Unique Identification*) 32-bit pada e-KTP atau RF tag/card 13.56 MHz.

c. *Push button*

Berfungsi untuk *interrupt* mengaktifkan solenoid tanpa adanya pembacaan oleh modul NFC.

d. *Magnetic Switch*

Magnetic switch terdiri dari sensor *hall effect* yang berfungsi untuk mengembalikan posisi *solenoid* pada kondisi pintu tertutup dari kondisi awal pintu telah terbuka.

e. Rangkaian *Driver Solenoid*

Solenoid bekerja pada tegangan 12V sedangkan keluaran yang dihasilkan pin mikrokontroler hanya 5V, oleh karena itu diperlukan rangkaian driver agar keluaran mikrokontroler dapat mengaktifkan *solenoid*. Rangkaian ini terdiri dari transistor dan *relay* yang berfungsi untuk *switching*.

f. Modul Xbee

Berfungsi untuk berkomunikasi, memberi dan menerima umpan balik dari pasangan mikrokontroler dan xbee lainnya. Pengalamatan xbee dapat dirancang agar sesuai topologi yang diinginkan.

2. Blok Sistem Penerima/logger

Blog sistem penerima merupakan *database* data dan presensi yang terdiri dari mikrokontroler Arduino, modul Xbee, *Real-time clock* dan modul SD card.

Data hasil presensi dapat dibuka dengan MS excel. Berikut adalah fungsi dari masing-masing bagian blok sistem penerima/logger:

a. Mikrokontroler Arduino (slave)

Berfungsi untuk mencocokkan UID (*Unique Identification*) yang dikirim oleh *slave*, melakukan *logger* dan mengirimkan umpan balik pada modul *slave*.

b. Real-time clock

Berfungsi untuk memberikan waktu dan tanggal pada mikrokontroler untuk *logger* presensi masuk.

c. Modul SD card

Berfungsi untuk menyimpan deret UID (*Unique Identification*) dengan format .txt dan menyimpan hasil logger presensi dalam bentuk .csv (*comma separated value*) dapat dibuka menggunakan MS excel.

4.2 Perancangan Perangkat Keras

4.2.1 Alokasi Pinout Mikrokontroler Arduino

Arduino merupakan papan rangkaian elektronik *open source* yang memiliki fungsi utama yaitu melakukan kontrol penuh terhadap antarmuka modul dan sensor yang digunakan, mikrokontroler Arduino yang digunakan dalam perancangan adalah Arduino R3 berbasis AVR 8 bit yaitu ATmega 328 dan memiliki ATmega 16U2 sebagai *converter* USB to serial.

Hasil pembacaan data secara *Real-time* dapat ditampilkan pada *serial monitor* yang tersambung dengan pin UART Arduino menyebabkan pengunggahan atau tulis ulang perangkat lunak harus memutus apapun beban yang tersambung pada pin Tx-Rx Arduino. Pemutusan beban ini juga berlaku pada tulis ulang mikrokontroler AVR yang menggunakan ICSP dengan antarmuka SPI sebagai sambungan pada PC atau writer.

Mikrokontroler Arduino Uno berbasis ATmega328 terdiri dari 20 pin I/O yang terpisah menjadi 14 pin I/O digital dan 6 pin I/O analog-digital konfigurasi pada mikrokontroler pembaca e-KTP sebagai berikut:

- 1). Pin digital 0: dihubungkan dengan pin Din pada Xbee.
- 2). Pin digital 1: dihubungkan dengan pin Dout pada Xbee.

- 3). Pin digital 2: dihubungkan dengan interrupt berupa bacaan *Hall effect*.
- 4). Pin digital 3: dihubungkan dengan interrupt berupa *push button*.
- 5). Pin digital 4: dihubungkan dengan pin data *driver relay solenoid*.
- 6). Pin digital 10: dihubungkan dengan pin CS modul NFC.
- 7). Pin digital 11: dihubungkan dengan pin MOSI modul NFC.
- 8). Pin digital 12: dihubungkan dengan pin MISO modul NFC.
- 9). Pin digital 13: dihubungkan dengan pin SCK modul NFC.
- 10). Pin analog 0/ digital 14: dihubungkan dengan pin RS pada LCD 16x2.
- 11). Pin analog 1/ digital 15: dihubungkan dengan pin *Enable* pada LCD 16x2.
- 12). Pin analog 2/ digital 16: dihubungkan dengan pin D4 pada LCD 16x2.
- 13). Pin analog 3/ digital 17: dihubungkan dengan pin D5 pada LCD 16x2.
- 14). Pin analog 4/ digital 18: dihubungkan dengan pin D6 pada LCD 16x2.
- 15). Pin analog 5/ digital 19: dihubungkan dengan pin D7 pada LCD 16x2.

Mikrokontroler Arduino Uno yang digunakan untuk master sebagai *logger* dan presensi dihubungkan dengan komunikasi UART nirkabel dengan Xbee pro S1 sehingga dapat menjangkau jarak kurang lebih 200 feet/90 meter kondisi Urban. Berikut konfigurasi pin pada mikrokontroler master logger:

- 16). Pin digital 0: dihubungkan dengan pin Din pada Xbee.
- 17). Pin digital 1: dihubungkan dengan pin Dout pada Xbee.
- 18). Pin digital 10: dihubungkan dengan pin CS modul SD card.
- 19). Pin digital 11: dihubungkan dengan pin MOSI modul SD card.
- 20). Pin digital 12: dihubungkan dengan pin MISO modul SD card.
- 21). Pin digital 13: dihubungkan dengan pin SCK modul SD card.
- 22). Pin analog 4/ digital 18: dihubungkan dengan pin SDA modul RTC.
- 23). Pin analog 5/ digital 19: dihubungkan dengan pin SCL modul RTC.

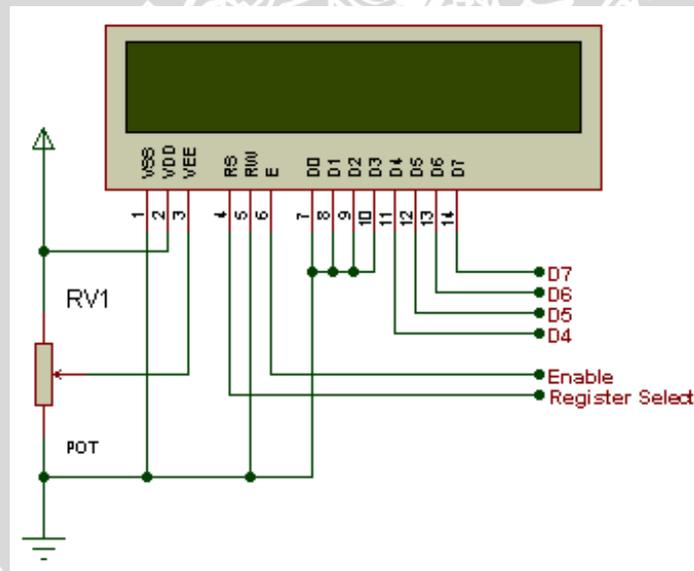
Kedua konfigurasi pin modul master dan slave berlaku kondisi pin digital 0/ data receive pin/Rx, pin digital 1/ data transeive pin/ Tx. Pin digital 2 dan 3 sebagai interrupt dengan deklarasi nilai `attach interrupt = 0` untuk pin 2 dan `attach interrupt = 1` untuk pin 3. Pin 10 hingga 13 telah dideklarasikan sebagai SPI oleh library NFC dan SD card. Pin 18 dan 19 telah dideklarasikan sebagai I2C oleh

library real-time clock. Selain pin yang disebutkan sebagai pin output digital untuk menampilkan data pada LCD dan perintah mengaktifkan driver solenoid.

4.2.2 Perancangan Rangkaian LCD (*Liquid Crystal Display*) 2X16 Karakter

Pada perancangan sistem ini digunakan LCD modul QC1602A yang merupakan sebuah modul LCD *dot matrix* yang membutuhkan daya kecil. LCD modul QC1602A dilengkapi panel LCD dengan tingkat kontras yang cukup tinggi. Rangkaian koneksi LCD karakter 2X16 ke mikrokontroler ditunjukkan dalam Gambar 4.2. LCD modul QC1602A mempunyai spesifikasi sebagai berikut:

- Memiliki 16 karakter dan 2 baris tampilan yang terdiri atas 5X7 dotmatrik.
- Memerlukan catu daya DC 5 V.
- Menggunakan 4bit data dan 3bit kontrol.
- Adjustable contrast*.

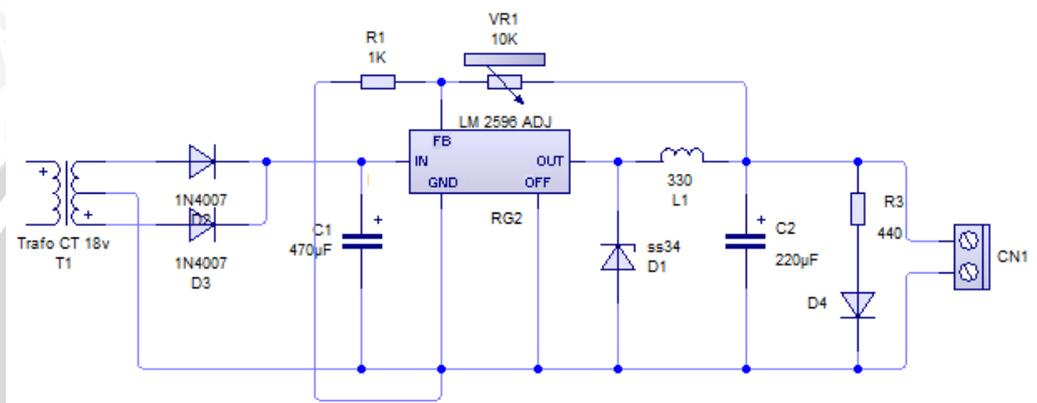


Gambar 4.2 konfigurasi pin LCD 2X16 Karakter untuk Mikrokontroler Arduino Uno

Dalam perancangan diinginkan agar kontras dan *backlight* dari LCD dapat diatur, maka dipasangkan sebuah variabel resistor dengan nilai 10k Ω pada pin *contrast* dari LCD 2X16 karakter untuk mengatur nilai tegangan yang masuk ke pin *contrast*.

4.2.3 Perancangan Rangkaian Catu Daya

Catu daya yang digunakan bersumber dari *power plug* rumah 220V AC, sedangkan mikrokontroler Arduino membutuhkan tegangan 12V DC melalui 2.1 mm jack DC (unregulated) atau 5V DC (regulated) melalui port Vin-GND. Diperlukan rangkaian penyearah tegangan AC dengan trafo dan meregulasikan tegangan dengan IC *adjustable converter* yaitu IC LM 2596-ADJ. Skematik rangkaian catu daya ditunjukkan dalam Gambar 4.3.



Gambar 4.3 Rangkaian catu daya 12V
Sumber: Texas Instruments, 2013: 3

Pada perancangan diatas, trafo yang digunakan adalah center tap transformator dengan beban maksimal 2A, dua buah diode 1N4007 sebagai penyearah gelombang penuh pada sisi sekunder trafo. Tegangan keluaran regulator dapat dihitung dengan perhitungan:

$$V_{OUT} = V_{REF} \left(1 + \frac{VR}{R_1} \right)$$

Dimana $V_{REF} = 1,23 \text{ V}$ sehingga jika tegangan keluaran diinginkan adalah 12 V dan R_1 ditetapkan 1 k Ω maka perhitungan :

$$VR = R_1 \left(\frac{V_{OUT}}{V_{REF}} - 1 \right)$$

$$VR = 1k\Omega \left(\frac{12V}{1,23V} - 1 \right)$$

$$VR = 8,75k\Omega \approx 10k\Omega$$

Untuk mengindikasikan sistem sedang menyala digunakan LED (*Light Emiting Diode*) superbright ukuran 5mm berwarna biru, forward voltage LED

tersebut sebesar 3,4 V sedangkan arus nominal yang melewati LED (I_{LED}) sebesar 20 mA. Nilai R_{LED} yang harus dipasang seri terhadap LED dapat dihitung berdasarkan perhitungan:

$$V_{CC} = I_{LED} \cdot R_{LED} + V_{LED}$$

$$12V = 20mA \cdot R_{LED} + 3,4V$$

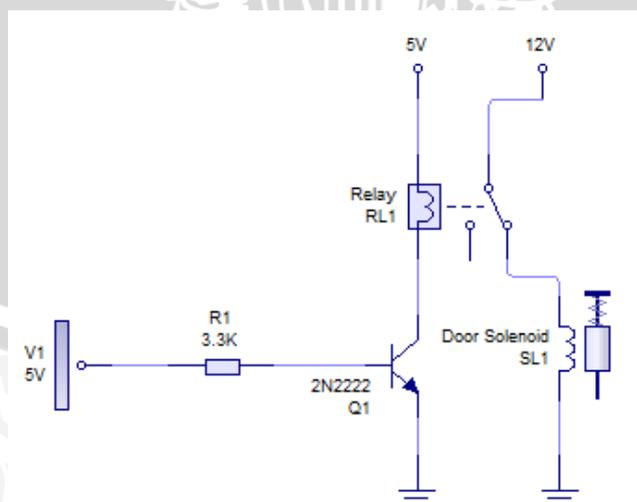
$$R_{LED} = \frac{(12 - 3,4)V}{20 \text{ mA}}$$

$$R_{LED} = 430\Omega$$

R_{LED} sebesar 430 Ω merupakan nilai maksimum agar LED tidak putus, maka dari itu digunakan dua buah resistor 220 Ω sehingga diperoleh nilai R_{LED} seri sebesar 440 Ω .

4.2.5 Perancangan *Driver Solenoid*

Tujuan dari perancangan ini adalah untuk mengintegrasikan *solenoid* dengan mikrokontroler. Mikrokontroler bekerja pada level tegangan TTL yaitu 5V, sedangkan *solenoid* membutuhkan tegangan 12V. Rangkaian ini terdiri dari komponen resistor, transistor 2N2222, dan relay tipe HRS4-H-S-DC5V. Relay ini bekerja pada tegangan minimal 3,75V dan memiliki resistansi *coil* sebesar 70 Ω dengan simpangan $\pm 10\%$. Skematik dari rangkaian *driver* ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Rangkaian *Driver Relay*

Tegangan keluaran high (V_{OH}) minimum mikrokontroller sebesar 4,2V tidak bisa mengaktifkan *relay*, maka digunakan transistor 2N2222 yang difungsikan sebagai saklar untuk memicu *relay*. Berikut spesifikasi transistor 2N2222 menurut *datasheet* memiliki karakteristik sebagai berikut:

$$I_{C(max)} = 800 \text{ mA}$$

$$V_{CE(sat)} = 400 \text{ mV}$$

$$V_{BE(sat)} = 1,3 \text{ V}$$

$$h_{FE} = 75$$

Dalam pengoperasiannya, transistor 2N2222 memerlukan Resistor Basis (R_b) agar dapat bekerja. Hal ini bertujuan untuk membatasi arus yang masuk pada kaki basis. Nilai R_b dapat dicari dari perhitungan sebagai berikut:

Nilai resistansi coil (R_{coil}) didapatkan dari pengukuran adalah 73Ω . Nilai arus kolektor (I_C) yang mengalir pada kaki kolektor transistor 2N2222 pada saat kondisi saturasi:

$$\begin{aligned} V_{CC} &= I_C \cdot R_{Coil} + V_{CE} \\ 5V &= I_C \cdot 73\Omega + 0,4V \\ I_C &= \frac{5V - 0,4V}{73\Omega} \\ &= 63,01mA \end{aligned}$$

Nilai I_C diatas masih dibawah arus yang mengalir pada kaki kolektor sebesar 800 mA. Nilai h_{fe} yang digunakan adalah yang terkecil karena untuk mencari arus basis sebesar mungkin. Nilai arus basis (I_B) yang mengalir pada kaki basis transistor 2N2222 dapat dihitung berdasarkan persamaan:

$$\begin{aligned} I_B &= \frac{I_C}{h_{fe}} \\ I_B &= \frac{63.01mA}{75} \\ &= 0,84mA \end{aligned}$$

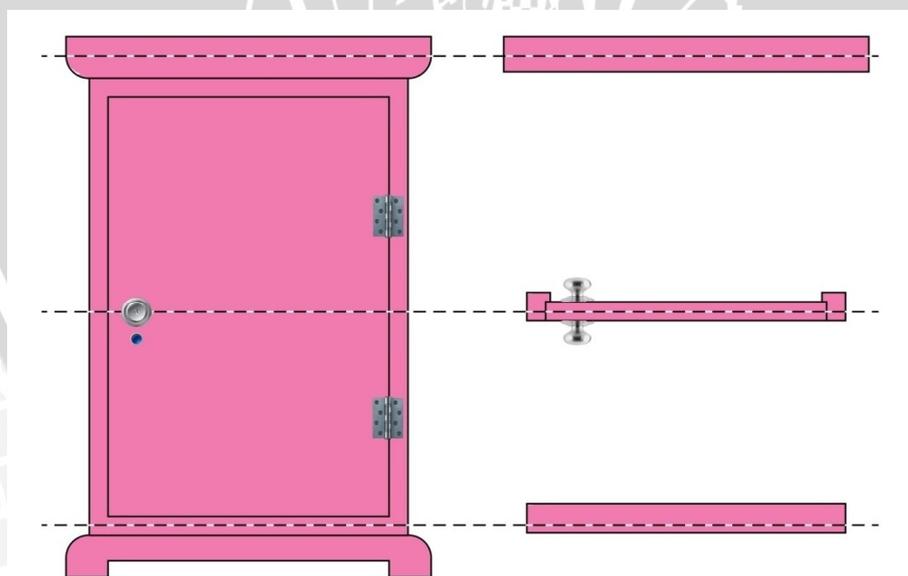
Nilai R_B dapat dihitung berdasarkan hubungan antara tegangan keluaran bias maju transistor (V_{BB}) dari mikrokontroller, arus bias (I_B) dan V_{BE} dengan persamaan:

$$\begin{aligned}
 V_{BB} &= I_B \cdot R_B + V_{BE} \\
 4,2V &= 0,84mA \cdot R_B + 1,3V \\
 R_B &= \frac{4,2V - 1,3V}{0,84mA} \\
 &= 3,45k\Omega \approx 3,3k\Omega
 \end{aligned}$$

Pembulatan nilai R_B dibawah perhitungan agar memastikan terjadi kondisi saturasi transistor dengan arus basis I_B seminimal mungkin.

4.2.6 Perancangan Mekanik Pintu

Pintu yang dibuat pada perancangan kunci elektrik ini adalah berbahan kayu lapis yang biasa terdiri dari 3 lapis sehingga disebut *three ply*/triplex. Ukuran pintu yang akan dibuat berdimensi sekitar tinggi 90 cm, lebar 60 cm, ketebalan 4.5 cm dengan isian kosong. kusen pintu berbahan kayu padat dengan lebar dan tinggi 4 cm lebih besar dari pintu ditunjukkan dalam Gambar 4.5. pada sisi belakang yang menghadap pada dalam ruangan diberikan sebuah *push button* sehingga keluarnya orang dalam ruangan menuju keluar ruangan dilakukan tanpa pembacaan e-KTP.

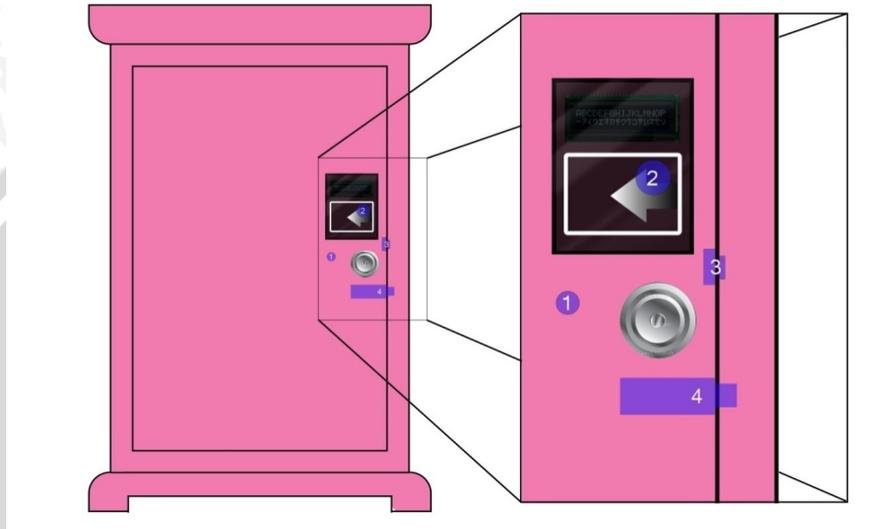


Gambar 4.5 Gambar rancangan pintu bagian belakang

Bagian dalam pintu yang kosong digunakan untuk meletakkan komponen yang terdapat pada *slave* ditunjukkan dalam Gambar 4.6 dengan keterangan:

- Komponen Arduino dan NFC diletakkan pada bawah lapisan nomor 1.

- b. LCD dan antenna NFC diletakkan pada lapisan bawah mika hitam transparan nomor 2.
- c. pada nomor 3 *Magnetic switch* yaitu sensor *Hall effect* diletakkan pada bagian pintu dan magnet neodmium diletakkan pada kusen pintu.
- d. *Solenoid door* diletakkan pada nomor 4.



Gambar 4.6 pintu sisi depan

Sumber listrik yang digunakan komponen modul slave dalam pintu adalah catu daya 5V DC dan solenoid menggunakan 12V DC. Modul *master* diletakkan terpisah dari modul *slave* dan dikemas bersama dengan catu daya 5V DC.

4.3 Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak menggunakan menggunakan bahasa C melalui *compiler* Arduino IDE dan selanjutnya diunggah pada mikrokontroler Arduino. Komunikasi antara *master* dan *slave* menggunakan Xbee yang telah dialamatkan oleh X-CTU.

4.3.1 Pengalamatan Xbee pro S1

Pada perancangan ini menggunakan dua buah mikrokontroler arduino yang saling terkoneksi satu sama lain dengan UART Xbee. Masing dari modul Xbee harus mempunyai alamat tertentu dan spesifik untuk menghindari konflik dari *default* alamat yang disediakan. Untuk mengamati atau set ulang modul Xbee diperlukan perangkat komunikasi USB to UART langsung dari PC.

Konverter USB to UART dapat diperoleh dengan memasang modul Xbee pada Arduino dengan melepas mikrokontroler ATmega 328 sehingga board Arduino hanya terdapat ATmega 16U sebagai USB to UART yang langsung tersambung pada Xbee. Perancangan ini menggunakan *software* digi X-CTU untuk set parameter pada Xbee pro S1.

Parameter utama untuk membentuk komunikasi P2P atau *point to point* adalah:

1. *Baud Rate*
2. PAN ID
3. *Channel*

Default PAN (*Personal Area Network*) ID yang diberikan adalah 3332 dengan DH=0x0000 dan DL=0xFFFF membentuk *boardcast mode* yaitu komunikasi *point to multi point*. Pada *boardcast mode* ini Xbee akan mengirimkan pesan pada Xbee lain dala area jangkauan dengan channel yang sama dan semua alamat. Parameter pertama yaitu *baud rate* yang diset pada Xbee harus sama dengan *baud rate* yang inisialisasi pada mikrokontoler Arduino, dalam perancangan ini *baud rate* yang digunakan adalah 3-9600. Parameter dan software X-CTU ditunjukkan dalam Gambar 4.7.



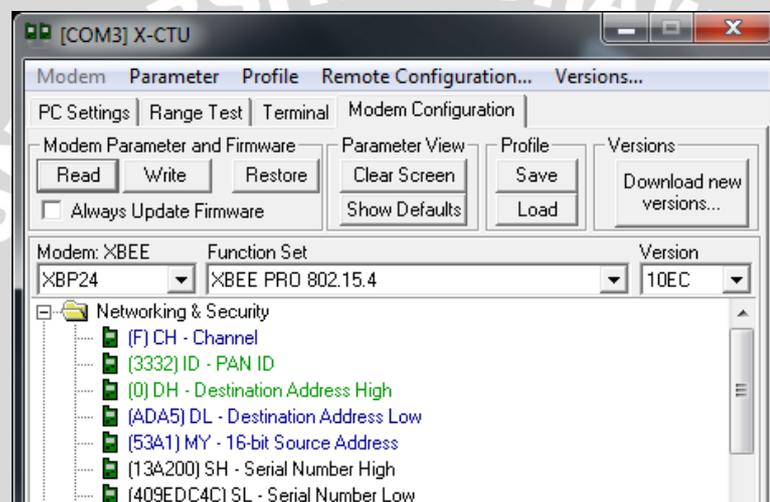
Gambar 4.7 set parameter pada X-CTU

Untuk membentuk *Unicast* atau *point to point* parameter ke-2 yaitu PAN ID harus sama. *Destination address* (parameter DH dan DL) menentukan modul yang akan menerima data yang ditransmisikan. Berikut beberapa aturan dalam penentuan parameter DH dan DL:

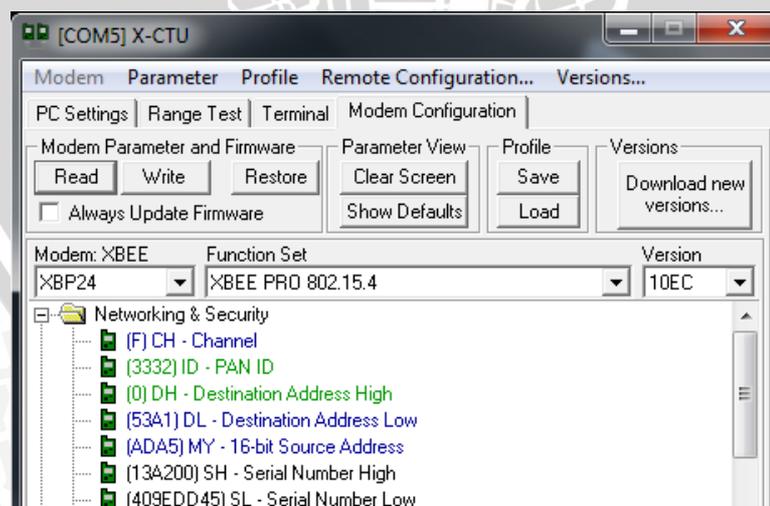
1. Jika modul DH adalah 0x0000 dan DL adalah kurang dari 0xFFFF (yaitu 16 bit), data yang dikirimkan oleh modul akan diterima oleh setiap modul dengan 16-bit alamat parameter MY sama DL.

2. Jika DH adalah 0 dan DL sama 0xFFFF, transmisi modul akan diterima oleh semua modul.
3. Jika DH adalah non-nol atau DL lebih besar dari 0xFFFF, transmisi hanya akan diterima oleh modul yang sama dengan nomor seri alamat tujuan transmisi modul (yaitu yang SH sama dengan transmisi modul DH dan yang sama dengan SL DL nya).

Parameter ketiga yaitu *channel*, *channel* yang digunakan kedua Xbee harus sama memiliki range 0x0B hingga 0x1A (16 *channel*). Pengalamatan dan set parameter pasang Xbee pada X-CTU ditunjukkan dalam Gambar 4.8 dan 4.9.



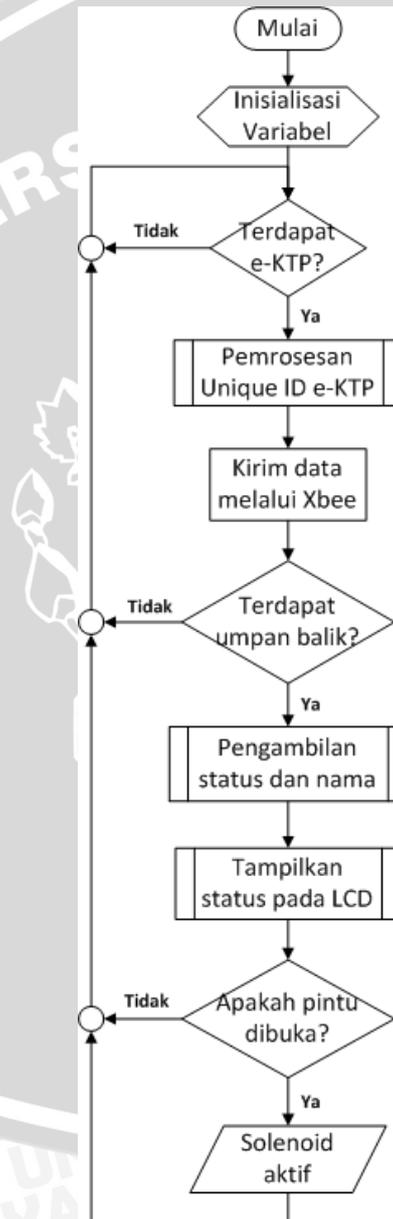
Gambar 4.8 Set parameter Xbee 1 pada *communication port 3*



Gambar 4.9 Set parameter Xbee 2 pada *communication port 5*

4.3.2 Perancangan Algoritma Blok Sistem Pembaca (*slave*)

Tujuan dari pembuatan algoritma adalah mengatur urutan kerja sistem dan sekaligus mempermudah dalam pemrograman sistem. Blok sistem pengirim mempunyai tugas utama yaitu memproses data e-KTP yang terbaca, menerima umpan balik dari modul *master* dan melakukan *interrupt*. Secara umum tugas yang harus dikerjakan oleh blok sistem pengirim ditunjukkan dalam Gambar 4.10.



Gambar 4.10 Algoritma blok sistem pembaca (*slave*)

Prinsip kerja alat dimulai pada modul *slave* yang tertanam pada pintu. Program dimulai dengan inisialisasi variabel, *library*, dan pengaturan *baudrate*

melalui komunikasi UART (*Universal Asynchronous Receiver Transmitter*) nirkabel xbee dengan 9600 bps. Data UID (*Unique Identification*) yang terbaca oleh NFC dikonversi dan ditambahkan karakter *header* sehingga data dapat dikenali oleh mikrokontroler *master*. Setelah melakukan pengiriman data, modul *slave* menunggu umpan balik yang mempunyai 2 *header* yang berbeda. Umpan balik tersebut digunakan untuk melakukan buka atau untuk tidak membuka *solenoid*.

Ketika terjadi pembukaan pintu, status nama dari pengakses ditampilkan pada LCD dan *solenoid* aktif sehingga pintu terbuka. Setelah pintu terbuka dan ditutup kembali *interrupt* picu tepi naik dari *magnetic switch* aktif dan posisi *solenoid* kembali pada posisi pintu terkunci. Selain *interrupt* dari *magnetic switch* terdapat *interrupt* picu tepi turun dari *push button* yang terletak pada pintu bagian dalam untuk membuka pintu dari dalam ruangan.

4.3.2.1 Perancangan Pembacaan dan Konversi UID e-KTP

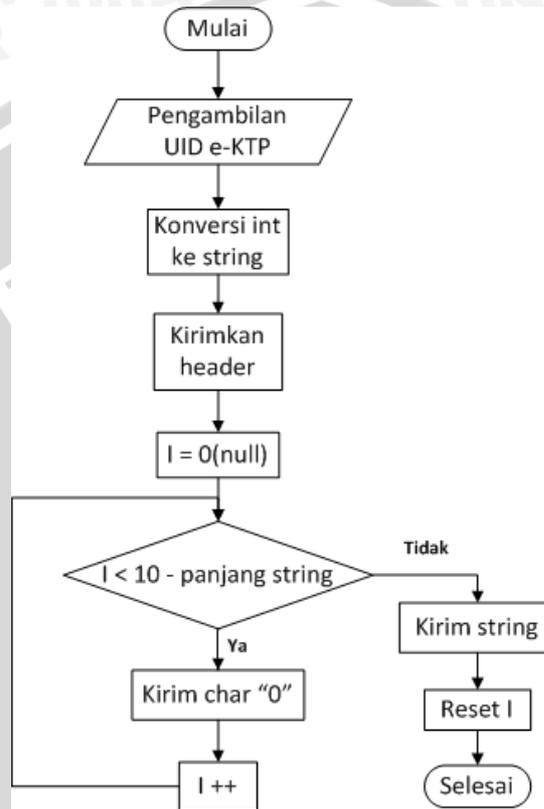
Data e-KTP yang terbaca oleh NFC adalah UID (*Unique Identification*) kombinasi 32bit yang telah dikonversi menjadi *unsigned integer* 32-bit yang mempunyai nilai dalam *range* 0 hingga 4.294.967.295. untuk memudahkan transmisi pada mikrokontroler lain dan komparasi terhadap *database* maka panjang dari data yang dikirim harus tetap dalam 10 digit. Namun tipe data *unsigned integer* maupun *integer* tidak dapat memberikan nilai “0” pada MSB dari contoh nilai yang terbaca adalah 9 digit atau “00” pada nilai terbaca 8 digit. Untuk pemberian nilai “0” pada MSB dilakukan dengan cara mengubah tipe data *integer* menjadi *string* dengan `snprintf` dengan format data:

```
snprintf(char * s, size_t n, format string, argument);
```

format data dengan `s` adalah *buffer* hasil *string* disimpan, `n` adalah maksimal byte yang digunakan dalam *string*, format *string* mengikuti format dalam `printf` dan *argument* adalah *format specifier* mengikuti format *string* sebelumnya.

Inisialisasi untuk mengirimkan *string* diawali dengan karakter “&” kemudian setelah data menjadi tipe data *string* maka panjang *string* akan dihitung menggunakan `strlen` dan setiap panjang *string* kurang dari 10 digit akan

ditransmisikan karakter "0" sebanyak 10 dikurangi panjang *string* terbaca. Setelah dilakukan transmisi karakter "0" maka diikuti dengan *string* dari nilai *string* terbaca. diagram alir pembacaan dan pengiriman UID e-KTP ditunjukkan dalam Gambar 4.11.



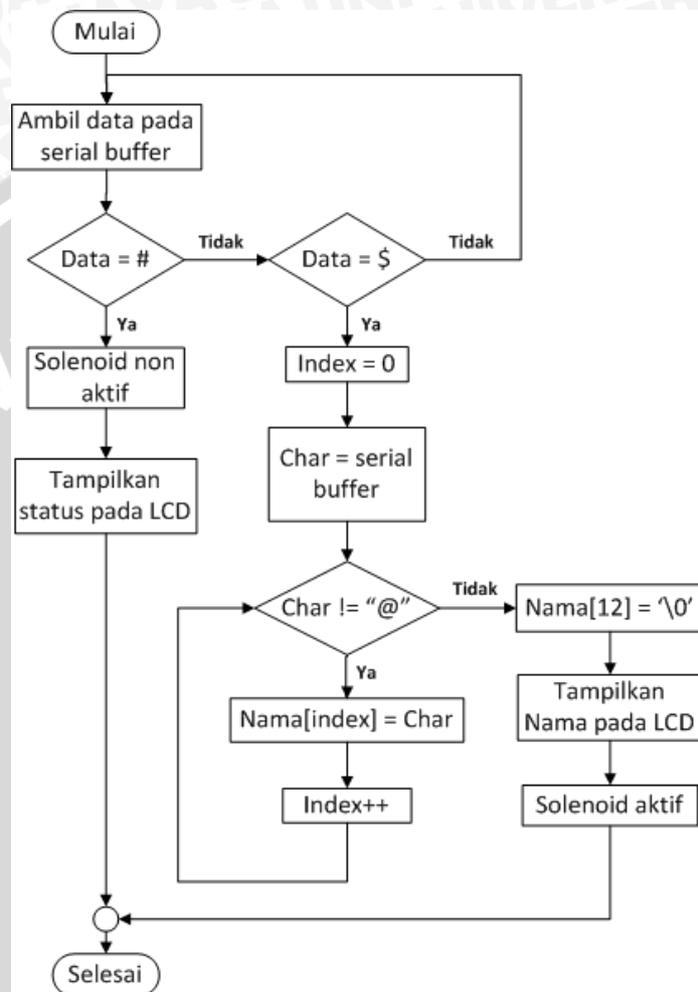
Gambar 4.11 Algoritma pengiriman UID e-KTP

4.3.2.2 Perancangan Algoritma Parsing Data Umpan Balik

Umpan balik yang ditransmisikan oleh modul *master* kepada *slave* merupakan hasil dari pencocokan UID (*Unique Identification*) dengan *database* yang terdapat pada SD card. Terdapat 2 macam perintah yang dikirimkan modul *master* untuk mengaktifkan *solenoid* atau untuk tidak mengaktifkan *solenoid*. *Header* yang dimulai dengan karakter "\$" adalah perintah untuk memberikan nilai HIGH pada *driver solenoid* dan akan diikuti dengan 12 karakter nama pengakses pintu dan kemudian diikuti dengan karakter penutup "%". 12 karakter pengakses pintu ni kemudian ditampilkan pada LCD.

Modul *master* akan mengirimkan data dengan karakter "#" dan tidak diikuti oleh data karakter selanjutnya untuk perintah tidak membuka atau

mengaktifkan *solenoid*. Data karakter “#” diterima oleh ,modul *slave* kemudian menampilkan *string* “ID tidak cocok!!” pada LCD *display* sebagai status. Diagram alir algoritma *parsing* data umpan balik ditunjukkan dalam Gambar 4.12

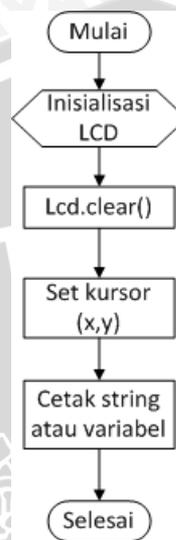


Gambar 4.12 Algoritma *parsing* data umpan balik

4.3.2.3 Perancangan Algoritma Display LCD

Algoritma display LCD dirancang untuk mengatur tampilan status pada baris *display* LCD 2x16 karakter. LCD 2x16 menampilkan data dengan koordinat kursor awal x dan y letak suatu variabel akan ditampilkan. Algoritma *display* pada LCD diawali inisialisasi jenis LCD dengan `lcd.begin(16,2)`. Karakter yang telah ditampilkan pada LCD tidak akan hilang dan menumpuk pada jarakter lain jika dilakukan penulisan tanpa adanya penghapusan memori dengan `lcd.clear()`. Penulisan pada LCD dilakukan dengan menentukan awal kursor pointer dengan

`lcd.setCursor(x,y)` dengan `x` adalah awal kolom dan `y` adalah awal baris, dengan urutan `x` dan `y` dimulai dengan angka 0. Variabel atau *string* dapat ditampilkan dengan perintah `lcd.print()`. Diagram alir penampilan data pada LCD ditunjukkan dalam Gambar 4.13.

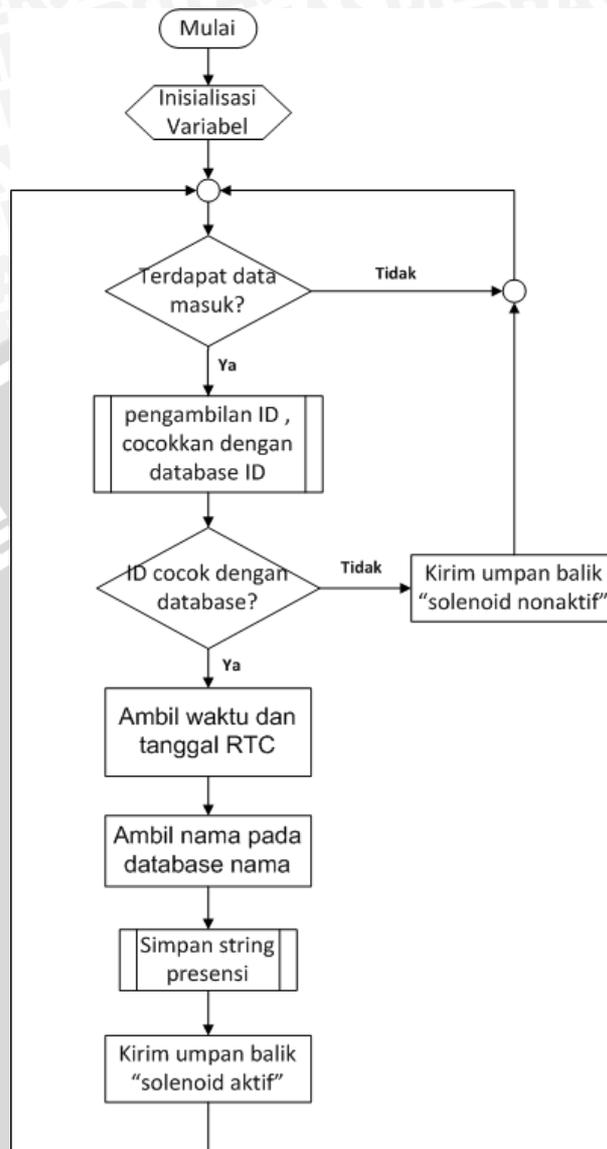


Gambar 4.13 diagram alir *display* data pada LCD

4.3.3 Perancangan Algoritma Blok Sistem Logger/Presensi (*master*)

Blog sistem *logger/presensi* memiliki tugas utama yaitu memberikan umpan balik pada modul *slave*. Data dari modul *slave* yang diterima pertama diparsing terlebih dahulu lalu dibandingkan dengan barisan data yang terdapat di database pada *SD card*. Jika data tersebut cocok maka mikrokontroler pada modul *master* akan melakukan pencatatan/*logging* presensi dengan menarik variabel waktu dan tanggal keluaran *real-time clock*. Mikrokontroler modul *master* lalu melakukan pembentukan *string* dan menyimpan data presensi masuk dengan format *.csv (comma separated value)* yang dapat dibuka dengan MS excel.

Data umpan balik setelah terjadi *logging* adalah perintah untuk mengaktifkan *solenoid* dan memberikan nama pengakses pintu dengan panjang 12 karakter. Jika data yang diterima tidak cocok dengan *database* yang ada, maka mikrokontroler *master* tidak akan melakukan *logging* dan hanya memberikan umpan balik untuk tidak mengaktifkan *solenoid* tanpa diikuti nama pengakses. Diagram alir algoritma blog sistem *logger* ditunjukkan dalam Gambar 4.14.

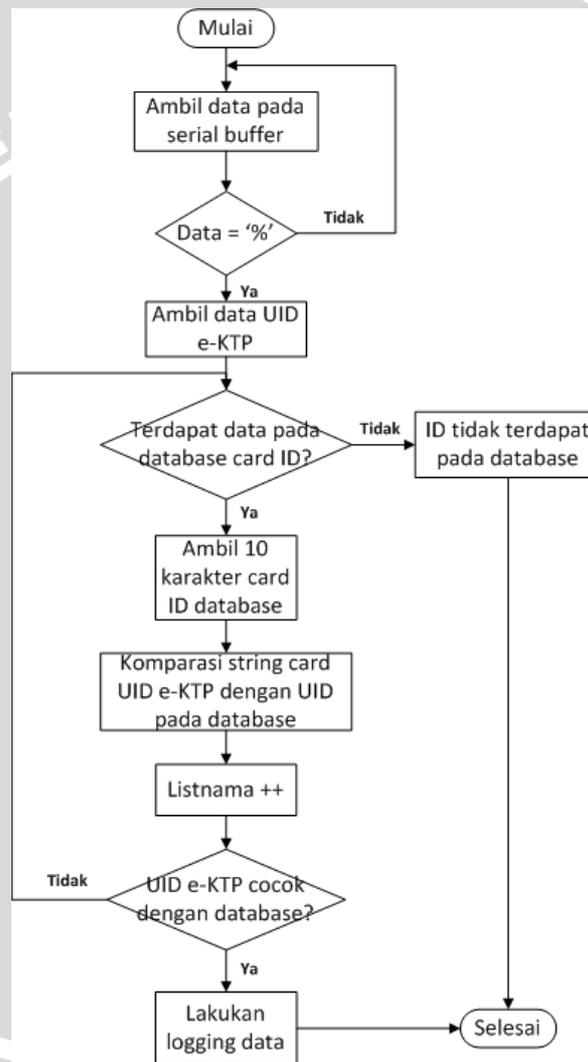


Gambar 4.14 diagram alir blog sistem *logger/presensi*

4.3.3.1 Perancangan Algoritma Parsing dan Komparasi Database

Pemilahan (*parsing*) data ditujukan untuk mencuplik data yang diinginkan. Data yang dikirimkan oleh modul *master* berupa tipe kartu dan UID (*Unique Identification*) agar hanya UID saja yang diambil maka transmisi data diawali dengan karakter *header* unik yang diselipkan sebelum UID sehingga pemotongan dan pengambilan data pada tengah *string* mungkin untuk dilakukan. Dalam perancangan ini *header* pada awal card ID menggunakan karakter "#". Modul *master* akan mengenali karakter "#" pada *buffer serial* dan mengambil 10 digit dan ditambahkan *string terminate* "/0", dan mikrokontroler akan mengambil 10

digit awal pada *database* untuk dilakukan komparasi. Jika komparasi tidak cocok maka mikrokontroler akan mengambil 10 digit dan akan melakukan komparasi hingga data pada *database* hingga habis. Setiap melakukan satu *loop* komparasi, terdapat suatu *variable* yang sifatnya mendapatkan *increment* untuk mengetahui posisi urutan UID, urutan tersebut dapat diambil untuk mengetahui urutan nama pada *database* nama. Diagram alir *parsing* dan komparasi *database* ditunjukkan dalam Gambar 4.15.



Gambar 4.15 diagram alir *parsing* data dan komparasi *database*

Sistem pemisahan *database* dengan menggunakan 2 *file* dengan format *.txt* difungsikan untuk memudahkan proses penambahan UID dan nama pada *database* masing-masing. Penambahan UID tidak diperbolehkan lebih atau kurang dari 10 karakter, penambahan nama tidak boleh lebih atau kurang dari 12 karakter. Jika

nama kurang dari 12 karakter maka akan dipenuhi dengan “ (spasi) hingga diperoleh 12 karakter. Urutan baris UID dan nama pada *file* masing-masing harus sama. Selain aturan diatas maka data *parsing* dan komparasi *database* akan gagal.

4.3.3.2 Perancangan String Logger Presensi

Pembentukan string yang digunakan untuk *logger* presensi didapatkan dari komparasi UID, data tanggal dan waktu data keluaran *real-time clock*, dan nama pengakses pintu. tipe data *integer* keluaran *real-time clock* dikonversi ke tipe data karakter dengan fungsi *itoa (integer to array)* yang disediakan *stdlib* dengan format:

```
char * itoa ( int value, char * str, int base );
```

format data dengan *value* adalah keluaran *real-time clock*, *str* adalah hasil variabel array yang tersimpan, dan *base* untuk menentukan basis tipe data integer yang terkonversi.

Seluruh data string lalu disatukan pada satu string utama untuk disimpan pada SD card. konversi penyatuan string dilakukan dengan menjumlahkan string masing masing dan menambah separator karakter “,”(*comma*). Berikut format data penyatuan string:

```
String result = string a + “,” + string b + “,” + . . .
```

String result kemudian disimpan pada SD card dengan ekstensi *.csv (comma separated value)*. Dalam *string a* tidak boleh terdapat *comma* karena setiap separator *comma* terbaca pada *string result* diidentifikasi sebagai data yang berbeda dengan kolom yang terpisah pada MS excel.