

BAB III

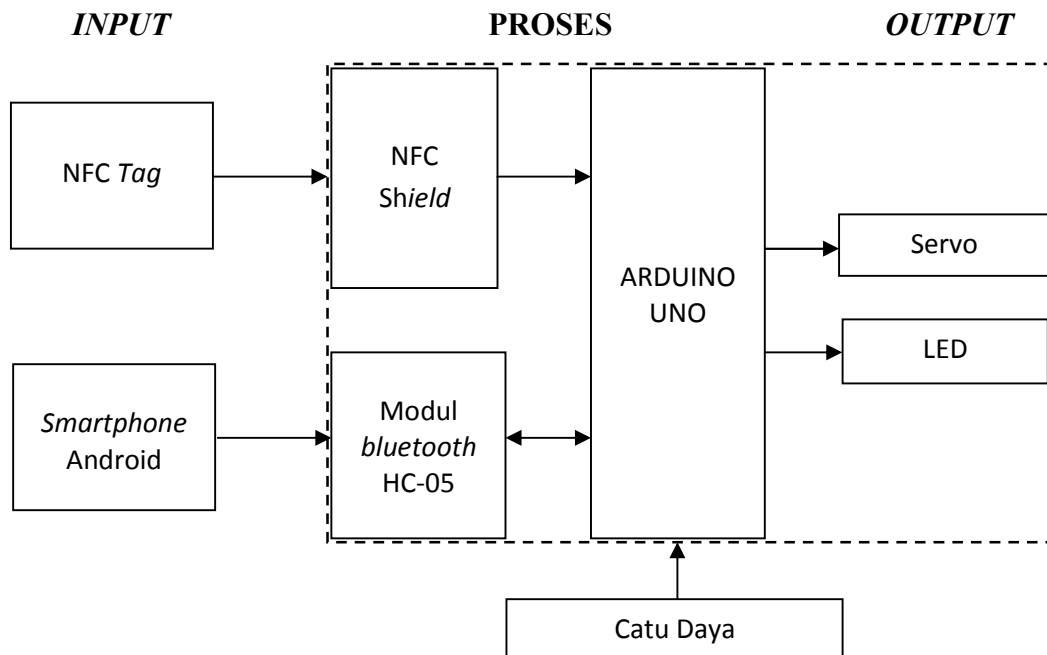
RANCANG BANGUN SISTEM

Pada bab ketiga ini, membahas tentang perancangan serta pembuatan sistem kendali pintu perangkat MSAN menggunakan *smartphone* berbasis *Near Field Communication* yang dikendalikan oleh Arduino UNO dengan komunikasi *bluetooth*. Sistem ini dimulai dengan perancangan perangkat keras (*hardware*), serta perancangan perangkat lunak (*software*).

Pada bagian perangkat keras akan dijelaskan mengenai perancangan mekanik dari sistem kendali pintu menggunakan Arduino UNO pada *smartphone user* yang memanfaatkan fasilitas *bluetooth* yang akan berkomunikasi dengan modul *bluetooth* HC-05 serta *NFC Shield* dengan *NFC Tag*. Kemudian, untuk perangkat lunak akan dibahas pertama kali tentang perancangan program bahasa Arduino sebagai pusat kendali dari perangkat – perangkat keras pada sistem ini, yang dimana program ini akan menentukan kinerja dari setiap perangkat keras yang terintegrasi. Sehingga *bluetooth* dan *NFC Shield* dapat bekerja secara bergiliran untuk mengendalikan motor servo sebagai *output* gerakan membuka dan mengunci dari pintu perangkat MSAN serta indikasi visual berupa LED yang akan menandakan pintu sedang terkunci atau terbuka.

3.1. PERANCANGAN SISTEM

Sistem ini memiliki bagian tugas masing – masing, antara lain *smarthpone android* yang memiliki fitur *bluetooth* akan berkomunikasi dengan modul *bluetooth* HC-05 dengan memanfaatkan pancaran sinyal modul *bluetooth* HC-05. *NFC Shield* sebagai perangkat yang berkomunikasi dengan *NFC Tag*, yang kemudian terhubung dengan Arduino UNO. Kedua perangkat tersebut menjadi sistem kendali dari pintu perangkat MSAN dengan menjadikan mikrokontroler Arduino UNO sebagai otak dari sistem kendali keseluruhan, catu daya sebagai sumber tegangan utama, lalu motor servo dan LED sebagai *output* dari sistem kendali ini. Gambaran umum mengenai sistem kerja dan pembagian blok sistem dari Tugas Akhir ini diilustrasikan pada gambar 3.1.



Gambar 3.1 Blok Diagram Sistem.

Pada gambar 3.1 diatas, menjelaskan mengenai cara kerja sistem kendali pintu perangkat MSAN menggunakan *smartphone* berbasis *Near Field Communcation* dengan *bluetooth* dan Arduino UNO. Sistem kerja alat dimulai dengan Inisialisasi mikropengendali Arduino, selanjutnya mengaktifkan *bluetooth* pada *smartphone* untuk melakukan *pairing* dengan modul *bluetooth* HC-05. *Bluetooth* digunakan sebagai media komunikasi untuk membuka pintu dari perangkat MSAN. Setelah saling terhubung antara *smartphone* dengan modul *bluetooth* lalu membuka aplikasi pada *smartphone* untuk melakukan kendali untuk membuka pintu pada perangkat MSAN. Kemudian dengan menggunakan *NFC Tag* berupa *sticker* yang nantinya tertempel pada *smartphone* berfungsi sebagai pengunci pintu dari perangkat MSAN. *Output* berupa servo sebagai simulasi untuk membuka serta mengunci pintu dan *output* LED sebagai peringatan indikator visual jika akses pintu berhasil dibuka dan dikunci kembali. Dalam melakukan akses membuka pintu, alat sistem kendali ini dirancang dalam keadaan *standby* sehingga pada saat melakukan pengendalian dapat dilakukan dengan jarak yang cukup jauh. Dengan memanfaatkan komunikasi *bluetooth*, memudahkan pengguna dalam melakukan pengendalian. Lalu *NFC tag sticker* menambah

keamanan pada perangkat MSAN dan memudahkan pengguna dalam melakukan penguncian.

3.2. PERANCANGAN PERANGKAT

Perancangan perangkat dalam alat sistem kendali pintu perangkat MSAN menggunakan *smartphone* dengan *bluetooth*, *Near Field Communication* (NFC) dan Arduino UNO, meliputi pembuatan perancangan perangkat keras dan perangkat lunak. Perancangan perangkat keras terdiri dari komponen berupa Arduino UNO sebagai sistem utama, modul *bluetooth* digunakan sebagai media untuk melakukan pembukaan pintu, sedangkan NFC sebagai perangkat yang akan melakukan penguncian pintu, lalu motor servo sebagai perangkat *open-lock* dari pintu perangkat MSAN, dan LED sebagai indikator visual dari pintu terkunci ataupun terbuka. Sedangkan dalam perancangan perangkat lunak yaitu perancangan program untuk mikrokontroler dan program aplikasi untuk pengendalian. Perancangan program pada mikrokontroler Arduino UNO menggunakan bahasa pemrograman *sketch* Arduino berbasis C++, dan perancangan program aplikasi menggunakan *App Inventor 2*.

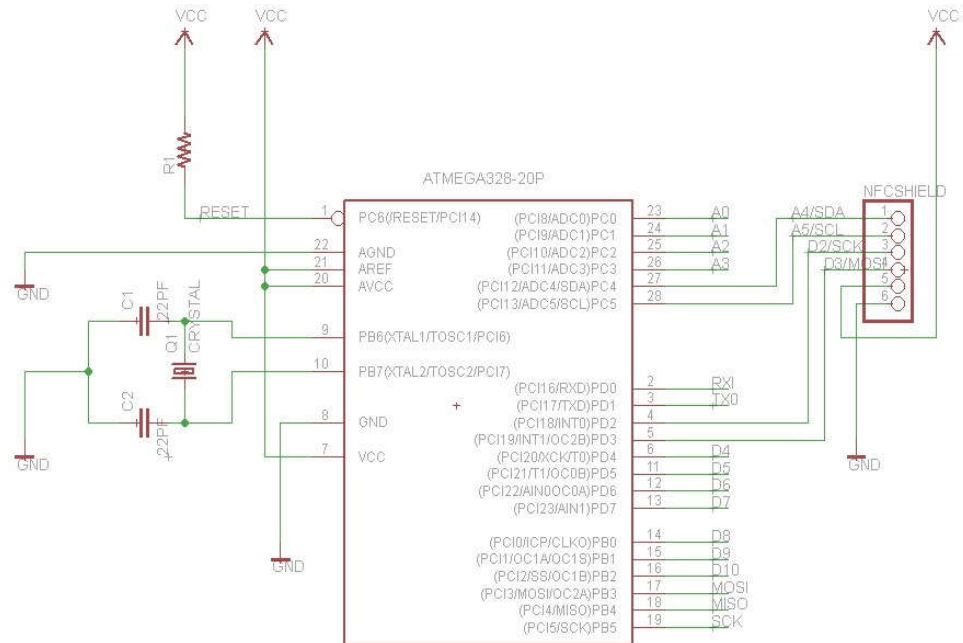
3.2.1 PERANGKAT KERAS

Perancangan *hardware* dalam alat sistem kendali ini menggunakan mikrokontroler Arduino UNO, modul *bluetooth*, motor servo, dan LED. Serta meliputi pembuatan rangkaian secara *schematic* menggunakan program EAGLE 7.1.0 serta penjelasan tiap perancangannya.

3.2.1.1 Perancangan Arduino UNO dengan NFC *Shield*

Perancangan Arduino dan NFC *Shield* pada kenyataannya sangat sederhana, karena dengan jumlah pin *male* pada NFC *Shield* yang sama dengan pin header *female* pada Arduino sehingga hanya dengan memasangkannya menyesuaikan posisi dari tiap pin *male-female* pada NFC *Shield* dan Arduino.

Pada perancangan ini NFC *Shield* hanya membutuhkan 7 pin yang terhubung dengan Arduino sebab menggunakan komunikasi I2C. Untuk lebih jelasnya perhatikan gambar 3.3.



Gambar 3.2 Rangkaian *schematic* NFC *Shield* dan Arduino
Penjelasan mengenai pin – pin yang digunakan pada NFC *Shield* dijelaskan pada tabel 3.1 dibawah ini.

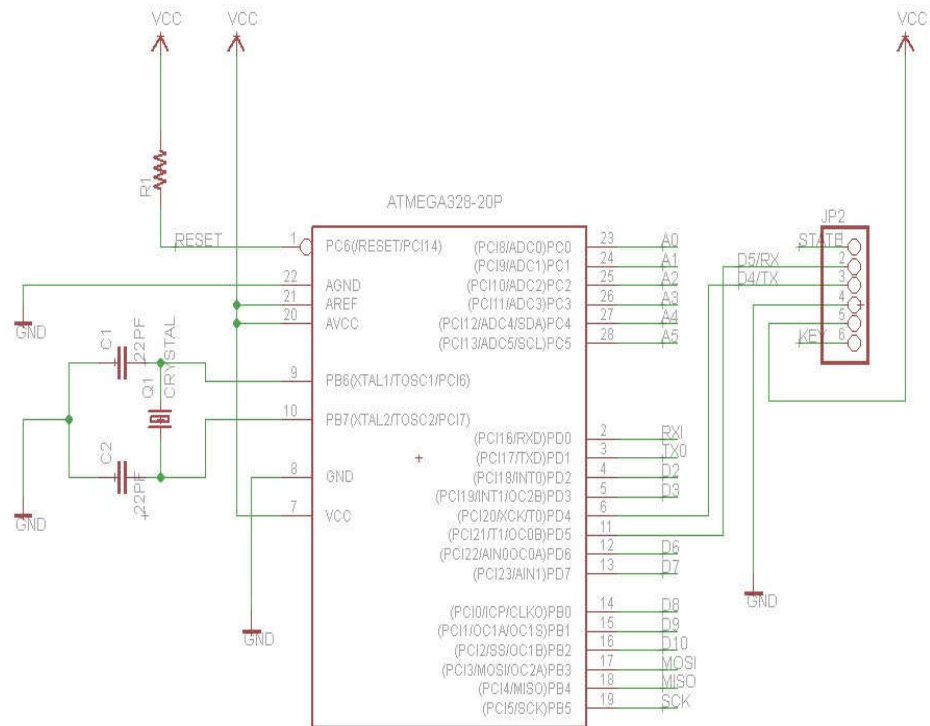
Tabel 3.1 Penjelasan pin pada NFC *Shield*

No.	Pin Arduino	Pin NFC	Penjelasann
1	Vcc 5V	Vcc 5V	Tegangan input 5V
2	RESET	RESET	Menghubungkan pin RESET pada Arduino
3	Analog 4	Analog 4/SDA	Mengaktifkan protokol I2C
4	Analog 5	Analog 5/SCL	
5	Ground	Ground	Menghubungkan rangkaian ground
6	D2	D2/SCK	Menghubungkan pin SCK pada NFC <i>Shield</i>
7	D3	D3/MOSI	Menghubungkan pin MOSI pada NFC <i>Shield</i>

3.2.1.2 Perancangan Arduino dengan *bluetooth* HC-05

Pada Arduino terdapat aplikasi untuk pengiriman data serial dengan bantuan Serial Monitor. Dalam sistem kendali ini konfigurasi pin yang dilakukan menggunakan *port digital* 4 untuk TX dan *port digital* 5 untuk RX pada Arduino. Dengan

memanfaatkan serial monitor tersebut dapat diketahui perintah atau informasi yang diberikan oleh *smartphone* ke modul *bluetooth*. Sedangkan untuk Vcc pada modul *bluetooth* dihubungkan ke Vcc dan *ground* dihubungkan sesuai Vcc dan *ground* pada Arduino.



Gambar 3.3 Rangkaian *schematic* Arduino dan *bluetooth* HC-05

Penjelasan mengenai setiap pin yang digunakan pada modul *bluetooth* HC-05 dijelaskan pada tabel 3.2 dibawah ini.

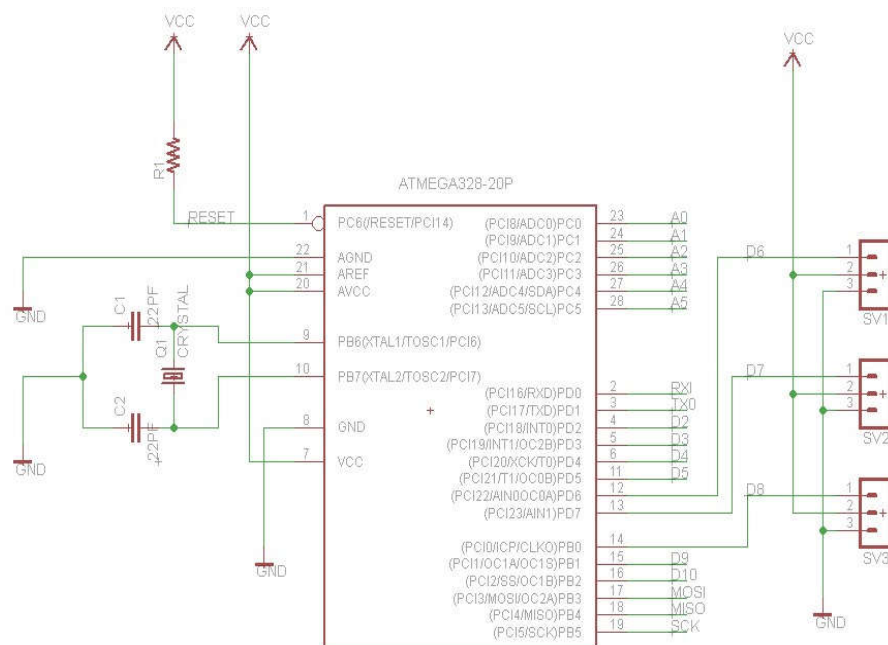
Tabel 3.2 Penjelasan pin pada modul *bluetooth* HC-05

No.	Pin Arduino	Pin <i>bluetooth</i> HC-05	Penjelasann
1	Vcc 5V	Vcc 5V	Tegangan input 5V
2	Ground	Ground	Menghubungkan rangkaian ground
3	D4	TXO	Menghubungkan pin Tx pada modul <i>bluetooth</i> HC-05
4	D5	RXI	Menghubungkan pin Rx pada modul <i>bluetooth</i> HC-05

3.2.1.3 Perancangan Arduino dengan motor servo

Motor servo termasuk dalam bagian *output* dari sistem kendali ini. Motor servo bekerja dengan tegangan *input* 5 Volt yang diambil dari tegangan arduino. Pada sistem ini motor servo

sebagai *open-lock* dari pintu perangkat MSAN dan akan bergerak ketika mendapatkan perintah dari aplikasi yang ada di *smartphone* yang dihubungkan dengan modul *bluetooth*. Motor servo yang digunakan sebanyak 3 buah, sebab pada perangkat MSAN terdapat 3 pintu, meliputi pintu depan, pintu samping kanan dan pintu samping kiri. Untuk 3 buah servo yang digunakan membutuhkan 3 *port* yang terhubung dengan Arduino. *Port* yang digunakan pada Arduino yaitu *port digital 6*, *port digital 7*, dan *port digital 8*. Untuk *Vcc* dan *ground* pada servo terhubung dengan *Vcc* dan *ground* pada Arduino.



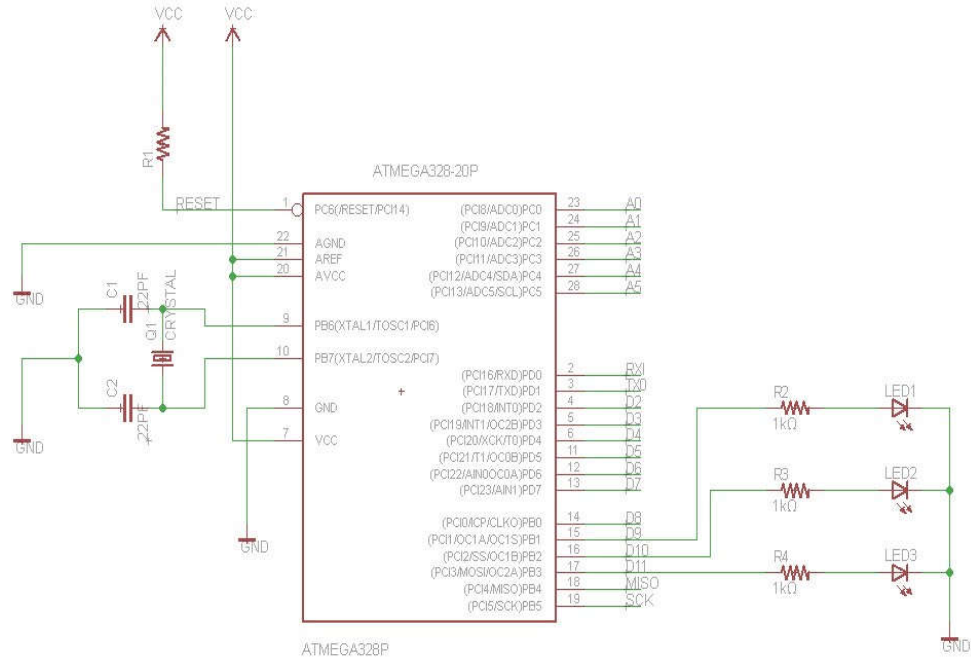
Gambar 3.4 Rangkaian *schematic* Arduino dan motor servo
Penggunaan pin – pin yang digunakan pada motor servo dijelaskan pada tabel 3.3 dibawah ini.

Tabel 3.3 Pin yang digunakan pada motor servo

No.	Pin Arduino	Pin motor servo	Penjelasan
1	Vcc 5V	Vcc 5V	Tegangan input 5V
2	Ground	Ground	Menghubungkan rangkaian ground
3	D6	Input Servo 1	Menghubungkan pin motor servo pertama
4	D7	Input Servo 2	Menghubungkan pin motor servo kedua
5	D8	Input Servo 3	Menghubungkan pin motor servo ketiga

3.2.1.4 Perancangan Arduino dengan LED

Pada sistem kendali ini, *Light Emitting Diode* (LED) difungsikan sebagai indikator visual.



Gambar 3.5 Rangkaian *schematic* Arduino dan LED

Pada perancangan Arduino dan LED, LED yang dibutuhkan berjumlah 3 buah LED. LED akan menyala ketika keadaan pintu pada perangkat MSAN dalam keadaan terkunci. Sedangkan LED akan mati ketika pintu pada perangkat MSAN dalam keadaan terbuka. Penggunaan *port* pada Arduino untuk LED pada Rangkaian *schematic* Arduino dan 3 buah LED pada gambar 3.5.

Penggunaan pin – pin yang digunakan pada LED dijelaskan pada tabel 3.4 dibawah ini.

Tabel 3.4 Pin yang digunakan pada LED

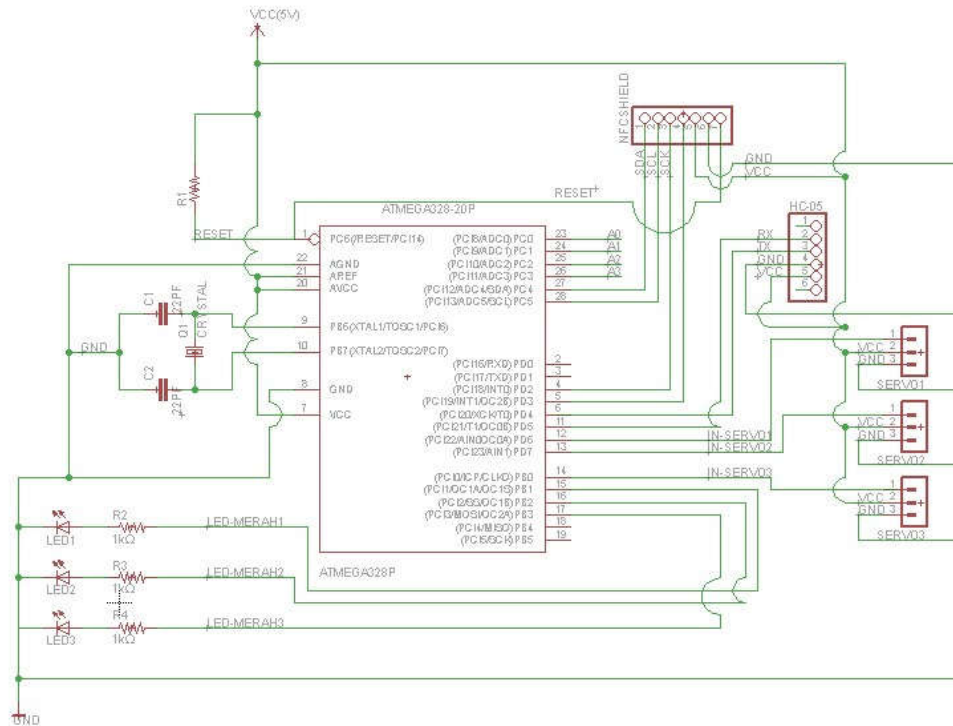
No.	Pin Arduino	Pin LED	Penjelasan
1	Vcc 5V	Vcc 5V	Tegangan input 5V
2	Ground	Ground	Menghubungkan rangkaian ground
3	D9	Input LED 1	Menghubungkan pin LED pertama
4	D10	Input LED 2	Menghubungkan pin LED kedua
5	D11	Input LED 3	Menghubungkan pin LED ketiga

3.2.1.5 Rangkaian Sistem Secara Keseluruhan

Perancangan rangkaian perangkat keras menggunakan *software* aplikasi Eagle 7.1.0. Dengan aplikasi Eagle 7.1.0 dirancang sebuah rangkaian *schematic* dari komponen utama hingga komponen – komponen lainnya. Pada *schematic* ini juga dijabarkan nama dari tiap komponen dan *port – port* yang digunakan untuk setiap komponen terhadap komponen utama. Rangkaian yang dibentuk terdiri dari bagian yang disusun dengan Arduino UNO, lalu NFC *Shield*, kemudian *Board* untuk komponen *bluetooth* HC-05, motor servo, dan LED.

Mikropengendali Arduino UNO sebagai sistem utama dari pengendalian sistem ini, harus terhubung langsung dengan sumber tegangan dari catu daya. Sumber catu daya menggunakan USB *interface* pada mikropengendali Arduino yang berasal dari catu daya PLN. Dalam sistem kendali ini pula modul *bluetooth* HC-05 juga mendapat catuan daya dari Arduino, namun dengan daya 5V untuk menjadikannya dalam keadaan *standby*. Setelah modul *bluetooth* HC-05 aktif dalam keadaan *standby*, selanjutnya *smartphone user* dihubungkan dengan modul *bluetooth*, apabila berhasil terhubung lalu membuka aplikasi kendali pada *smartphone* untuk memberikan perintah, namun sebelum masuk ke menu utama user perlu memasukkan *password* untuk melakukan kendali. Jika *password* benar maka Arduino akan mengeksekusi program untuk menjalankan *output*. Begitu pula dengan NFC *Shield*, apabila *tag* yang di-*scan* terdapat informasi khusus maka Arduino akan mengeksekusi program agar *output* dapat berjalan. Dalam sistem kendali pintu perangkat MSAN menggunakan *smartphone* berbasis *Near Field Communication* dengan *bluetooth* dan Arduino, fungsi dari *bluetooth* sebagai pembuka pintu. Sedangkan untuk NFC berfungsi untuk pengunci pintu. *Output* berupa motor servo difungsikan sebagai pengunci dan pembuka

dari pintu perangkat MSAN. Sedangkan untuk *output* LED warna merah yang masing – masing terhubung dengan resistor 1 k Ω menjadi indikator visual apabila pintu terbuka atau terkunci. Rangkaian sistem secara keseluruhan dapat dilihat pada gambar 3.6 dibawah ini.



Gambar 3.6 Rangkaian *schematic* sistem secara keseluruhan

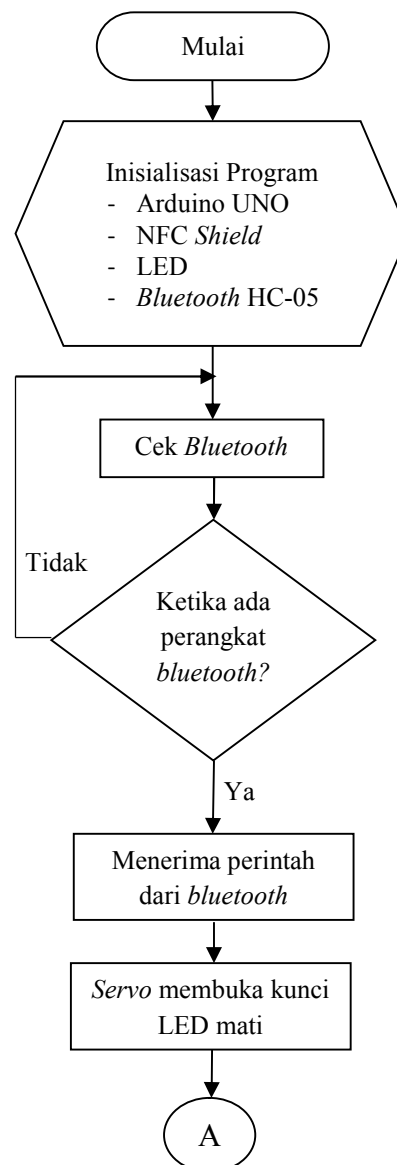
3.2.2 PERANGKAT LUNAK

Dalam perancangan perangkat lunak sistem kendali pintu perangkat MSAN menggunakan *smartphone* dengan *bluetooth*, *Near Field Communication* (NFC) dan Arduino, meliputi pembuatan *flowchart* program, perancangan program pada mikrokontroler Arduino, perancangan program untuk NFC, perancangan program *bluetooth*, perancangan program servo, dan perancangan program untuk LED. Pembuatan simulasi alat ini menggunakan *software* Arduino IDE 1.6.9 dengan bahasa pemrograman yang digunakan adalah bahasa C. Format file yang disimpan menjadi file tipe *.ino* atau *.pde* yang hanya dapat dijalankan dengan *software* Arduino IDE. Namun, jika pemrograman yang telah dibuat selesai mengalami *error* akan muncul notifikasi

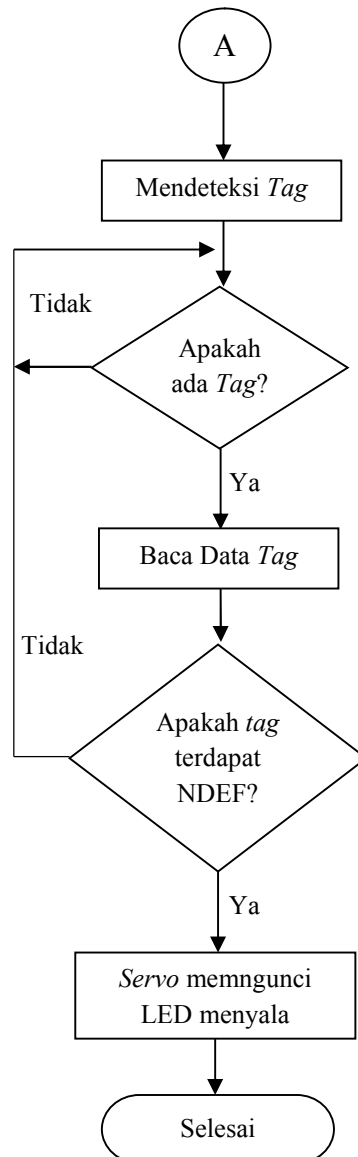
dimana letak *error* nya serta penjelasan mengenai *error* yang terjadi, akibatnya program tersebut tidak dapat dijalankan pada *board* Arduino.

3.2.2.1 Alur Perencanaan dan Pemrograman Arduino UNO

Dalam perancangan sistem kendali pintu perangkat MSAN dibagi menjadi beberapa bagian, terdapat *flowchart* pembuatan program yang akan menjelaskan langkah – langkah cara kerja dari program ini, dengan melihat *flowchart* langkah kerja dapat diketahui perintah serta intruksi kerja dari sebuah sistem yang akan dijalankan.



Gambar 3.7 *Flowchart* program utama



Gambar 3.7 Flowchart program utama [lanjutan]

Flowchart rancangan pembuatan program utama ditunjukkan pada gambar 3.7.

Dalam flowchart tersebut untuk memluai sebuah perencanaan program diawali dengan kata “mulai” kemudian diikuti oleh inisialisasi program. Inisialisasi program dalam sebuah flowchart pembuatan alat merupakan proses pengenalan atau persiapan agar setiap bagian atau perintah dalam program tersebut dapat dijalankan dan juga sebagai tanda bahwa program tersebut sudah siap untuk dijalankan. Selanjutnya apabila program sudah siap dijalankan, kemudian setting komunikasi

buetooth antara *smartphone* android dengan modul *bluetooth* pada Arduino.

a) Program inisialisasi Arduino

Inisialisasi program merupakan proses pengenalan agar setiap bagian ataupun perintah dalam program tersebut dapat dijalankan serta sebagai tanda bahwa program tersebut siap untuk dijalankan. Untuk inisialisasi program antara lain sebagai berikut:

```
#include <SoftwareSerial.h>
#include <Servo.h>
#include <Wire.h>
#include <PN532_I2C.h>
#include <PN532.h>
#include <NfcAdapter.h>
PN532_I2C pn532_i2c(Wire);
NfcAdapter nfc =
NfcAdapter(pn532_i2c);
Servo myservo1;
Servo myservo2;
Servo myservo3;
int bluetoothTx = 4;
int bluetoothRx = 5;
int pos;
int ledPin1 = 9;
int ledPin2 = 10;
int ledPin3 = 11;
SoftwareSerial
bluetooth(bluetoothTx, bluetoothRx);
```

Dalam struktur pembuatan program sintak Arduino terdapat bagian *setup ()*. Bagian *setup* ini termasuk dari bagian inisialisasi program yang berfungsi memanggil perintah untuk sekali saja pada waktu program pertama kali dijalankan. Keadaan awal dari pintu perangkat MSAN

dengan LED merah menyala yang menandakan keadaan pintu terkunci. Untuk sintak bagian *setup ()* yaitu sebagai berikut:

```
void setup()    {
    pinMode (ledPin1, OUTPUT);
    pinMode (ledPin2, OUTPUT);
    pinMode (ledPin3, OUTPUT);
    digitalWrite (ledPin1, HIGH);
    digitalWrite (ledPin2, HIGH);
    digitalWrite (ledPin3, HIGH);
    myservo1.attach(6);
    myservo2.attach(7);
    myservo3.attach(8);
    bluetooth.begin(9600);
    nfc.begin();
}
```

b) Setting komunikasi dengan *bluetooth*

Pada perancangan sistem kendali pintu perangkat MSAN digunakan komunikasi dengan *bluetooth* untuk pertama kali dilanjutkan komunikasi dengan NFC. Agar *bluetooth* dapat bekerja dengan baik tanpa adanya tumpang tindih dalam penggunaannya maka setting yang digunakan untuk *listing* program pada *bluetooth* adalah sebagai berikut :

```
void loop()    {
    while (bluetooth.available() > 0 )
    {
        int servopos = bluetooth.read();
    }
}
```

c) Setting pengecekan dan kesesuaian data informasi pada NFC *tag*

Kemudian untuk setting pengecekan dan kesesuaian data informasi pada NFC *tag* yang didekatkan ke NFC *Shield*, lalu NFC *Shield* akan mengecek kesesuaian informasi dan

meneruskannya ke Arduino UNO untuk diproses. Berikut *listing* program yang digunakan :

```
if (nfc.tagPresent()) {  
    NfcTag tag = nfc.read();  
    tag.print();  
    if (tag.hasNdefMessage())
```

d) *Listing* program pada motor servo dan LED

Motor servo akan bergerak dan LED akan menyala apabila modul *bluetooth* yang terhubung dengan *bluetooth* pada *smartphone user* menerima perintah dari aplikasi yang ada pada *smartphone* dan *NFC Shield* menerima komunikasi dari *NFC tag* yang memiliki sebuah informasi khusus berupa *NDEF Message*. Jika sistem kendali pada *bluetooth* berjalan, motor servo akan bergerak dengan sudut 90° untuk membuka pintu, LED merah mati apabila telah menerima perintah dari *smartphone*, berikut *listing* program yang digunakan dalam sistem kendali menggunakan *bluetooth* :

```
digitalWrite (ledPin1, LOW);  
digitalWrite (ledPin2, LOW);  
digitalWrite (ledPin3, LOW);  
myservo1.write(servopos);  
myservo2.write(servopos);  
myservo3.write(servopos); }
```

Kemudian akan bergerak kembali dengan sudut 90° seperti semula untuk mengunci pintu dan LED merah akan menyala setelah *NFC tag* terdeteksi. *Listing* program yang digunakan antara lain sebagai berikut :

```
for (pos = 90; pos > 0;)  
{  
    digitalWrite (ledPin1, HIGH);  
    digitalWrite (ledPin2, HIGH);  
    digitalWrite (ledPin3, HIGH);  
    myservo1.write(pos);
```

```
myservo2.write(pos);  
myservo3.write(pos);  
}  
} }
```

3.2.2.2 Sistem Operasi Android

Aplikasi pada *smartphone user* menggunakan sistem operasi android, sistem operasi android digunakan sebagai *input* untuk menjalankan sistem kendali pintu dari perangkat MSAN. Karena menggunakan sistem operasi android aplikasi dari sistem kendali ini memiliki format *.apk*.

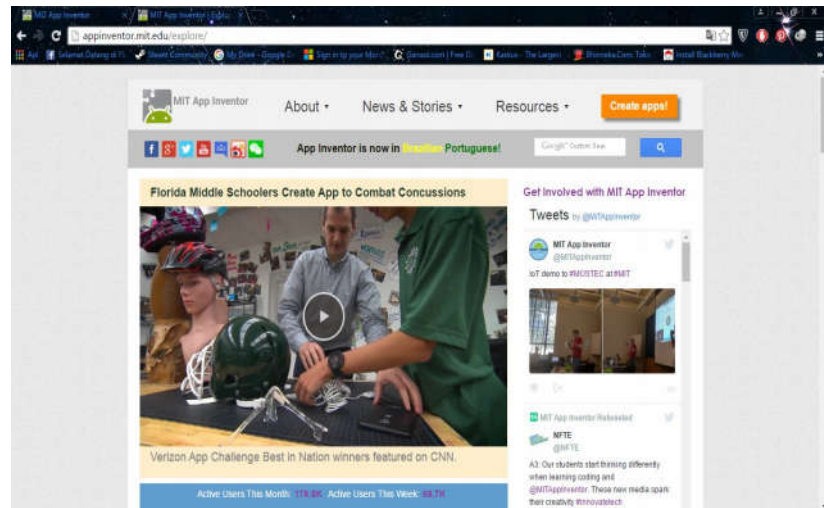
a) *App Inventor 2*

Perancangan aplikasi android dirancang menggunakan aplikasi *App Inventor 2*. *App Inventor 2* merupakan pengembangan *App Inventor Classic* yang pertama, dengan *design* dan penggunaannya yang sangat mudah. Caranya yaitu dengan menentukan tampilan aplikasi yang diinginkan pada blok *screen* serta mengatur tata letak dari blok – blok layaknya *puzzle* yang berfungsi sebagai *coding*, sehingga rancangan aplikasi dapat langsung menjadi sebuah aplikasi android yang dapat diinstal pada *smartphone* android dengan tipe file (*.apk*). *Project* yang sedang maupun telah dibuat akan tersimpan secara otomatis apabila dibuka pada lain waktu di alamat web *App Inventor 2*, sebab *App Inventor 2* telah terhubung dengan akun *google*.

Dalam perancangan tampilan aplikasi pada sistem kendali pintu perangkat MSAN menggunakan desain tampilan aplikasi pada *App Inventor 2* dengan kondisi *online*, jadi dalam pembuatannya memerlukan koneksi internet. Dengan mengakses alamat web <http://appinventor.mit.edu/explore>, dapat memulai membuat *project* aplikasi sistem kendali pintu perangkat MSAN menggunakan *smartphone* berbasis NFC

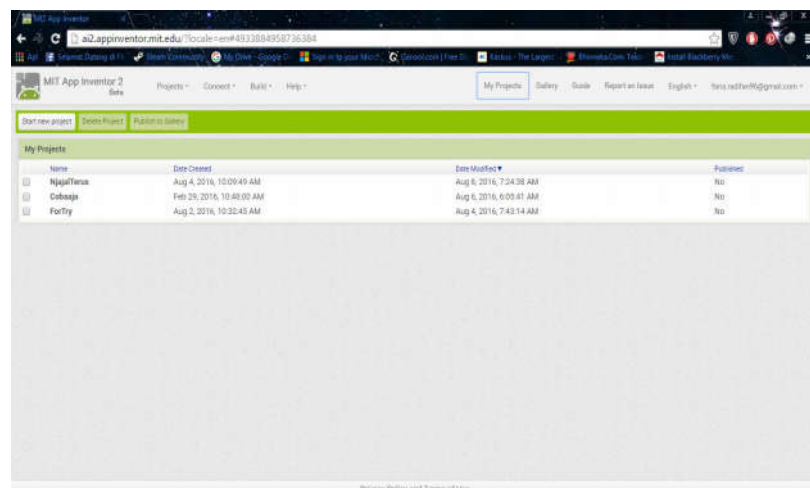
dengan *bluetooth* dan Arduino. Pada gambar 3.8 halaman utama dari *web App Inventor*.

Untuk memulai pembuatan sebuah *project* aplikasi pada halaman utama dengan memilih “*Create apps*” kemudian secara otomatis akan terhubung pada *App Inventor 2*.



Gambar 3.8 Halaman utama *App Inventor*

Kemudian setelah memilih “*Create apps*” akan muncul halaman selanjutnya yang menampilkan *project* baru untuk dibuat serta *project* apa saja yang telah dibuat dalam membuat aplikasi menggunakan *App Inventor 2*. Untuk tampilan *project* apa saja yang telah dibuat terdapat pada gambar 3.9.



Gambar 3.9 Tampilan halaman daftar *project* yang telah dibuat

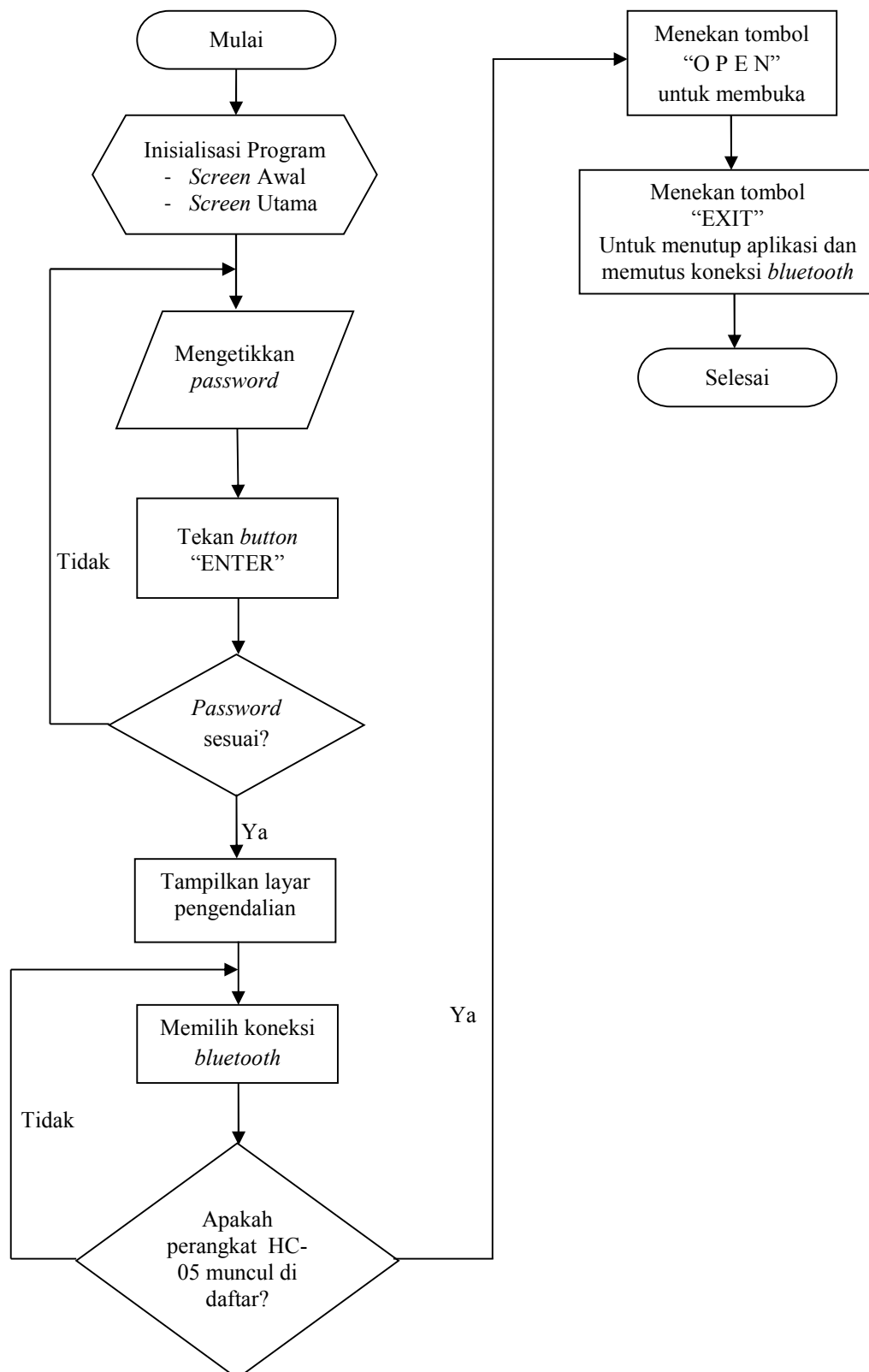
Flowchart langkah kerja dari aplikasi Android pada sistem kendali ini ditampilkan pada gambar 3.10. Dalam

perancangannya dibuat 2 “*screen*” yang terdiri *screen* pertama untuk mengakses ke menu utama dengan mengisikan *password* pada kolom *password*, kemudian untuk *screen* kedua yaitu menu utama untuk mengendalikan pintu perangkat MSAN untuk membuka pintu dalam keadaan terkunci, di dalam *screen* kedua terdapat *button* yang bertujuan untuk menghubungkan *smartphone* ke modul *bluetooth*.

Dalam *flowchart* langkah kerja aplikasi sistem kendali pintu perangkat menggunakan *smartphone* berbasis NFC dengan *bluetooth* dan Arduino untuk memulainya dengan membuka aplikasi pada *smartphone*, kemudian muncul *screen* awal untuk masuk ke *screen* berikutnya yang merupakan menu kendali. Dengan memasukkan *password* berupa kombinasi huruf dan angka yang telah disesuaikan, apabila *password* benar maka menu utama akan muncul, namun jika *password* salah maka akan muncul notifikasi “*INVALID PASSWORD !!*”. Setelah muncul menu utama maka langkah awal yang dilakukan untuk melakukan pengendalian adalah menghubungkan *bluetooth* pada *smartphone* dengan modul *bluetooth* HC-05 pada Arduino.

Dengan menekan logo *bluetooth* pada *screen* utama maka akan muncul pilihan perangkat *bluetooth*. Pilih perangkat dengan keterangan HC-05 agar saling terhubung, jika keterangan pada *screen* utama berubah dari “*not connected*” dengan teks berwarna merah menjadi “*connected*” dengan teks berwarna hijau, tandanya *bluetooth* telah berhasil terhubung.

Kemudian, menekan *button* “*OPEN*” untuk membuka pintu, setelah itu *smartphone* akan mengirim informasi pada Arduino untuk menggerakkan servo supaya terbuka dan mematikan LED. Setelah selesai, tekan *button* “*EXIT*” untuk menutup aplikasi dan memutus hubungan (*disconnect*) antara *smartphone* dengan modul *bluetooth*.



Gambar 3.10 *Flowchart* langkah kerja dari aplikasi Android pada sistem kendali pintu perangkat MSAN

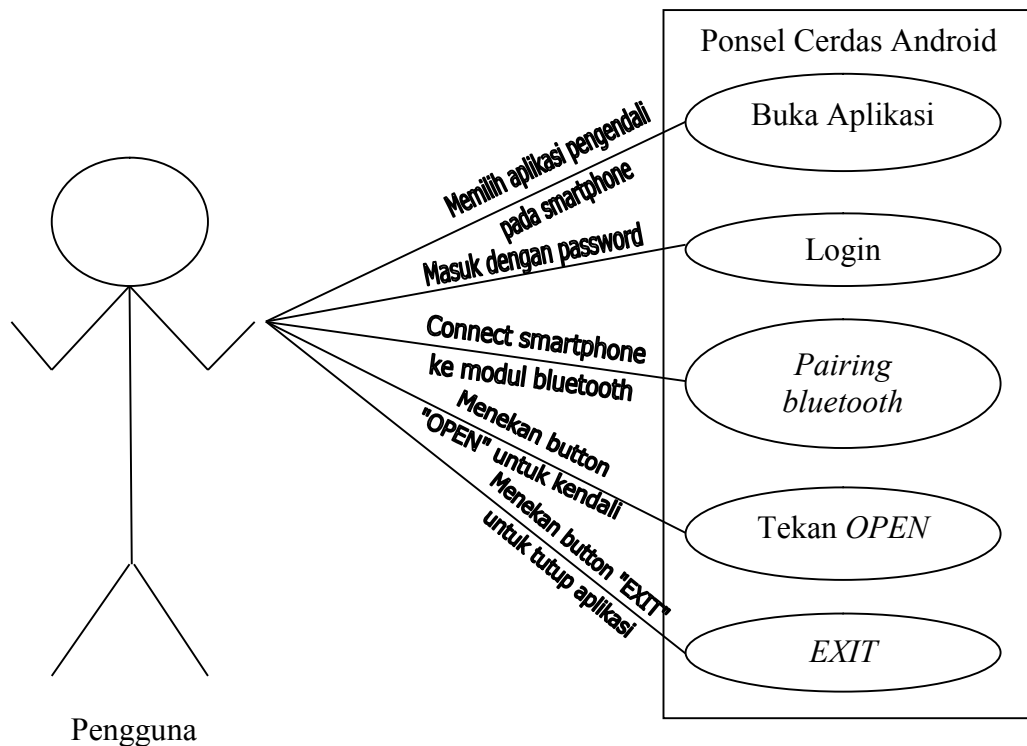
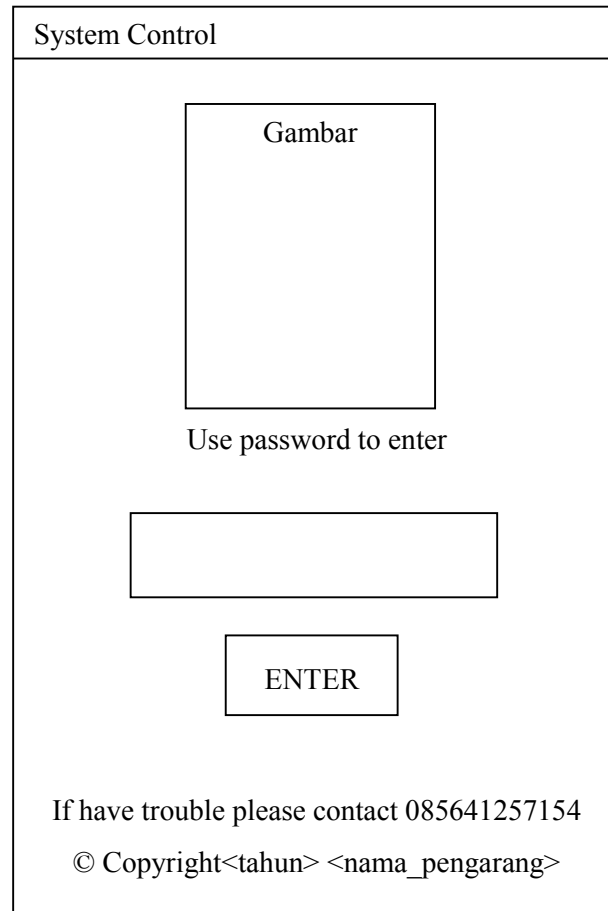
a) Diagram *Use Case*Gambar 3.11 Diagram *Use Case*

Diagram *use case* digunakan pada perancangan aplikasi sistem kendali ini sebab diagram *use case* pada prinsipnya mendeskripsikan interaksi yang terjadi antara pengguna (*user*) dengan sistem atau perangkat lunak yang sedang dijalankan. Aplikasi yang dibuat bertujuan untuk digunakan oleh kalangan tertentu yaitu hanya Perusahaan Telekomunikasi khususnya yang memiliki jarlokaf.

b) Sketsa Antarmuka Grafis

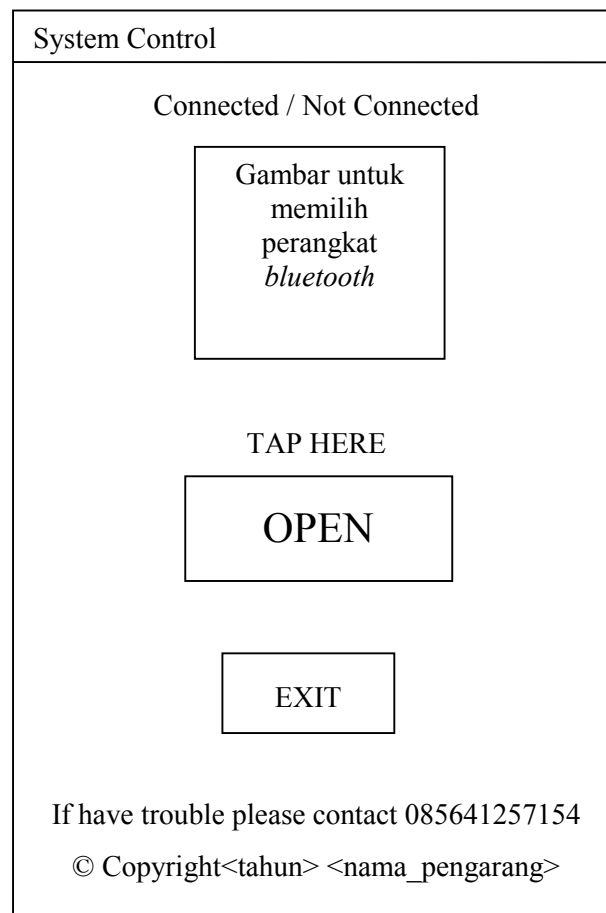
Sketsa antarmuka grafis aplikasi pada *screen* 1 yang merupakan tampilan awal untuk mengakses ke *screen* utama berupa menu untuk mengendalikan pintu. Dalam *screen* awal diharuskan mengisikan *password* yang sesuai agar dapat mengakses ke menu utama berupa menu pengendalian. Jika *password* yang dimasukkan salah maka akan muncul notifikasi dan *user* tidak dapat mengakses ke menu utama.



Gambar 3.12 Sketsa tampilan *screen* awal

Dalam pembuatan *screen* awal ini diperlukan beberapa tambahan gambar yang berfungsi untuk menambah desain tampilan, label teks digunakan untuk panduan mengakses aplikasi ke menu utama serta kolom *password* untuk mengisi *password* yang difungsikan sebagai fasilitas keamanan akses dan menentukan *password* yang diisikan telah sesuai.

Kemudian sketsa antarmuka grafis aplikasi pada *screen* 2 yaitu menu utama untuk melakukan pengendalian pintu perangkat MSAN. Terdapat beberapa *tools* yang digunakan untuk merancang *screen* utama antara lain label teks untuk membuat keterangan konektivitas antara modul *bluetooth* HC-05 dengan *smartphone* pengguna, lalu gambar digunakan untuk pemilihan perangkat *bluetooth* ketika *smartphone* pada *bluetooth* aktif.



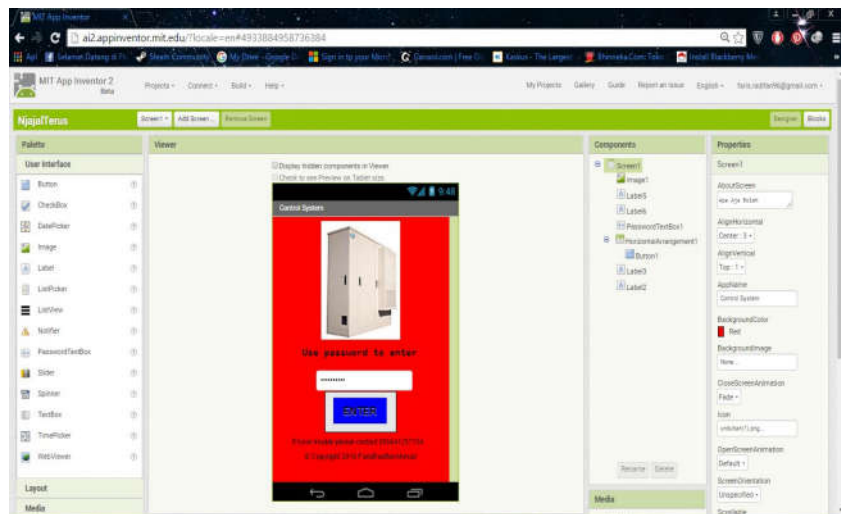
Gambar 3.13 Sketsa tampilan *screen* utama

Keterangan label teks yang berfungsi sebagai desain dan panduan untuk melakukan kendali membuka pintu dengan menekan *button* yang ada dibawahnya. *Button* “QUIT” berfungsi menutup aplikasi dan memutuskan konektivitas dengan modul *bluetooth*.

c) Perancangan desain *user interface*

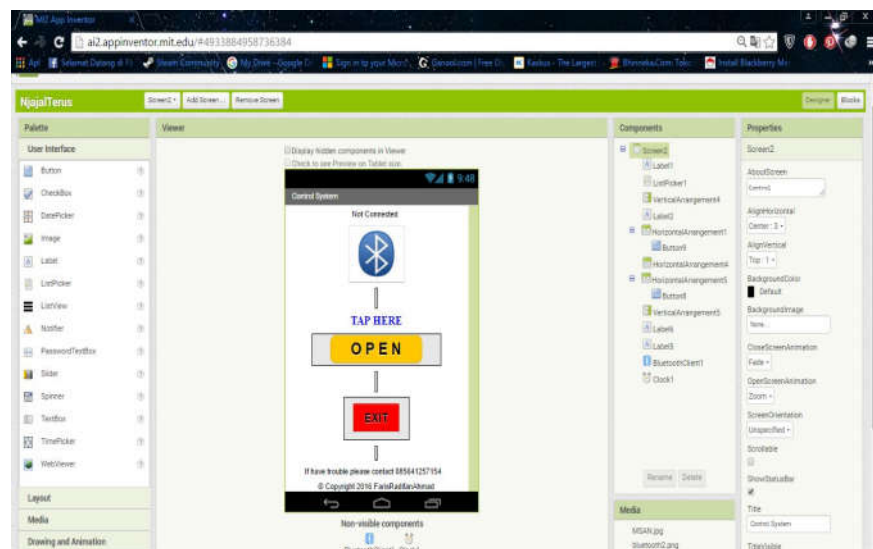
Desain *user interface* dirancang pada *App Inventor 2* dengan menggunakan koneksi internet. Untuk merancang *user interface* pada *screen* awal, dengan fasilitas yang ada pada *App Inventor 2* cukup dengan melakukan *drag-and-drop* komponen *user interface* yang ada di bagian kiri tampilan dari *project* yang sedang dibuat, lalu diletakkan pada blok tampilan *screen* yang akan menjadi *screen* awal pada layar *smartphone*. Pada *screen* awal digunakan gambar sebagai

desain tambahan dan kolom *password* sebagai akses menuju *screen* utama.



Gambar 3.14 Perancangan *user interface* pada *screen* awal

Setelah merancang *screen* awal, selanjutnya yaitu merancang *screen* utama sebagai pengendalian pintu dari perangkat MSAN. Pada *screen* utama ini, dirancang sesuai dengan *flowchart* langkah kerja dari aplikasi seperti pada gambar 3.10. Setelah selesai maka aplikasi ditutup dan otomatis *bluetooth* akan diputus-hubungkan antara *smartphone* dan modul *bluetooth* HC-05.



Gambar 3.15 Perancangan *user interface* pada *screen* utama

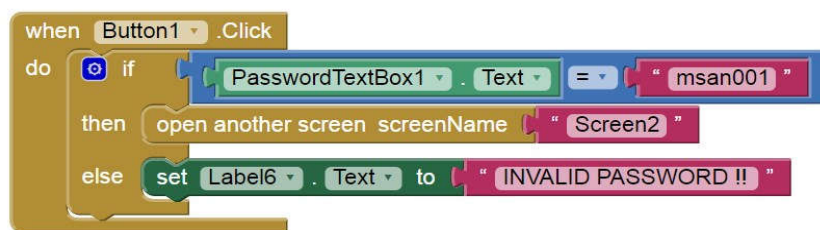
Dimulai dari proses menghubungkan *smartphone* dengan modul *bluetooth* HC-05 hingga keterangan label menjadi

“*Connected*” dan teks berwarna hijau dilanjutkan dengan menekan *button* agar pengendalian dapat bekerja.

d) Perancangan *block editor*

Kemudian apabila perancangan *user interface* telah selesai langkah selanjutnya merancang *block editor* pada aplikasi sistem kendali pintu perangkat MSAN. Pembuatan program untuk menjalankan perintah pada *App Inventor 2* tidaklah sama dengan menuliskan *coding* program pada pembuatan aplikasi berbasis Android lainnya. Pada *App Inventor 2*, *coding* program berbentuk blok-blok layaknya *puzzle* yang berisi perintah dalam pemrograman. *App Inventor 2 Blocks Editor* dibuka melalui pilihan menu yang berada disebelah pojok kanan atas dari *project* yang sedang dibuat bersebelahan dengan *App Inventor 2 Designer*.

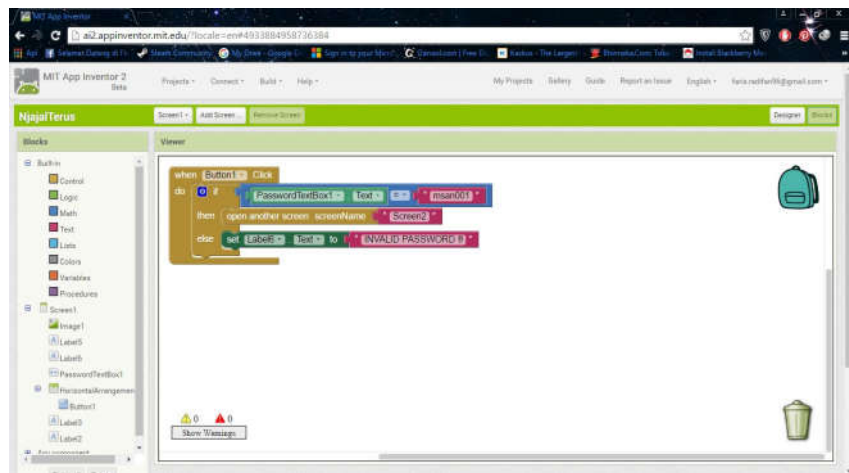
Gambar 3.16 menjelaskan *puzzle block* untuk *screen* awal berupa akses menuju *screen* utama, ketika *button 1* di tekan dengan verifikasi *password* “msan001”, jika *password* benar maka *screen* selanjutnya akan terbuka. Namun, apabila *password* salah atau tidak sesuai maka akan muncul notifikasi berupa “INVALID PASSWORD !!” dan *screen* selanjutnya tidak akan muncul.



Gambar 3.16 *Puzzle block* pada *screen* awal

Pada gambar 3.17 menunjukkan *block editor* pada *screen* awal pada saat aplikasi pertama kali dibuka. Dari *block editor* pada *screen* awal dapat dilihat perintah – perintah yang akan digunakan. Untuk merancang *block editor* masih sama dengan merancang desain *user interface* hanya dengan melakukan *drag-and-drop* lalu menyusunnya secara

berurutan menyesuaikan tools yang digunakan pada desain *user interface*.



Gambar 3.17 Tampilan *block editor* pada *screen* awal

Beberapa *tools* digunakan dalam perancangan *screen* utama seperti *button*, *listpicker*, label teks, serta fasilitas pendukung berupa *clock timer* dan *bluetooth*.

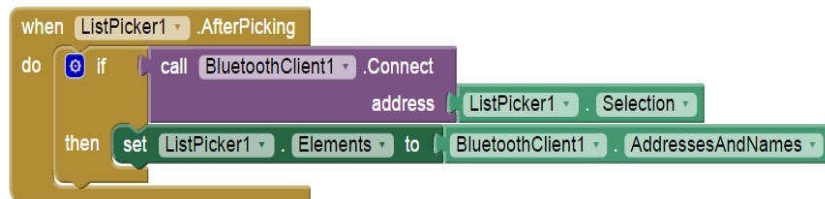
Perancangan *screen* utama diawali dengan *tools listpicker*. Terdapat dua kondisi untuk *tools listpicker* pada perancangan *screen* utama pada sistem kendali ini.



Gambar 3.18 *puzzle block* untuk *listpicker before.picking*

Kondisi pertama yaitu *listpicker before.picking*, dalam kondisi ini *listpicker* berfungsi untuk mempersiapkan aktifitas koneksi *bluetooth* pada *smartphone* sebelum pengguna memilih koneksi perangkat *bluetooth* yang akan dipilih. Lalu, untuk kondisi kedua yaitu *listpicker after.picking* berfungsi memberikan keterangan bahwa perangkat *bluetooth* yang dipilih telah terdeteksi oleh *smartphone*. Dalam *listpicker* akan tertera perangkat *bluetooth* mana yang aktif dan pernah terhubung dengan *smartphone* pengguna, kemudian memilih perangkat *bluetooth* dengan keterangan HC-05 untuk melakukan

pairing antara *smartphone* pengguna dengan perangkat *bluetooth*.



Gambar 3.19 puzzle block untuk *listpicker after.picking*

Selanjutnya penggunaan *tools clock timer* yang berfungsi menon-aktifkan waktu menyala untuk *bluetooth* pada *smartphone* sehingga *bluetooth* pada *smartphone* pada posisi *standby* tetap dapat mendeteksi adanya perangkat *bluetooth*.



Gambar 3.20 puzzle block untuk *clock timer*

Pada *puzzle block* untuk *clock timer* terdiri dari dua kondisi juga apabila *bluetooth* telah terhubung ataupun tidak. Untuk keadaan pertama apabila *bluetooth* pada *smartphone* terhubung dengan perangkat *bluetooth* maka label teks akan berubah menjadi keterangan "*Connected*" dan warna teks menjadi hijau, namun jika sebaliknya apabila *bluetooth* pada *smartphone* tidak terhubung dengan perangkat *bluetooth* maka keterangan label teks menjadi "*Not Connected*" dan warna teks menjadi merah.



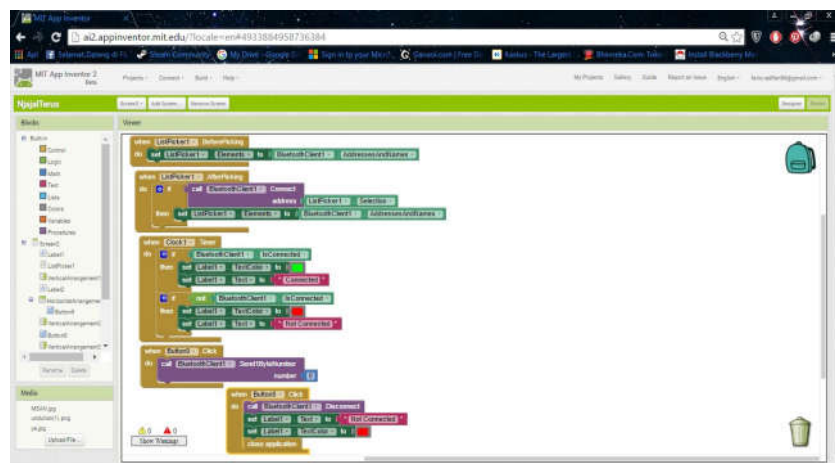
Gambar 3.21 puzzle block untuk *button* sebagai pengendalian

Puzzle block untuk *button 9* pada *screen* utama berfungsi sebagai pengendalian untuk membuka pintu perangkat MSAN. Ketika *button* ditekan maka melalui *bluetooth smartphone* yang telah terhubung dengan perangkat *bluetooth HC-05*, pengiriman *byte number* yaitu “0” yang akan dibaca dan dijalankan oleh Arduino dengan data berupa angka 0.



Gambar 3.22 *puzzle block* untuk *button disconnect* dan menutup aplikasi

Pada *puzzle block* untuk *button 8* sebagai *tools* bahwa pengendalian telah selesai dilakukan. Ketika *button 8* ditekan konektivitas *bluetooth* antara *smartphone* dan perangkat *bluetooth* terputus, kemudian label teks berubah menjadi keterangan “*Not Connected*” dan warna teks berwarna merah serta menutup dan keluar dari aplikasi sistem kendali pintu perangkat MSAN menggunakan *smartphone* berbasis NFC dengan *bluetooth* dan Arduino UNO. Berbeda dengan *screen* awal tadi, *screen* utama lebih banyak memiliki *puzzle block* sebab *screen* utama dirancang untuk melakukan pengendalian pada pintu perangkat MSAN.



Gambar 3.23 Tampilan *block editor* pada *screen* utama