



TUGAS AKHIR - TE 145561

REMOTE CONTROL PEMBUKA PINTU PAGAR OTOMATIS
YANG TERINTEGRASI OLEH *SMARTPHONE* DENGAN SISTEM
KEAMANAN *PASSWORD* ATAU SENSOR SIDIK JARI

Ilvira Zahida
NRP 2214030107

Dosen Pembimbing
Ir. Gatot Kusrahardjo, MT.

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 145561

*AUTOMATIC REMOTE CONTROL TO OPENED GATE
INTEGRATED BY SMARTPHONE WITH USING PASSWORD
OR FINGERPRINT SECURITY*

Ilvira Zahida
NRP 2214030107

Supervisor
Ir. Gatot Kusrahardjo, MT.

*DEPARTEMENT OF ELECTRICAL ENGINEERING AUTOMATION
Faculty of Vokasi
Sepuluh Nopember Institute of Technology
Surabaya 2017*

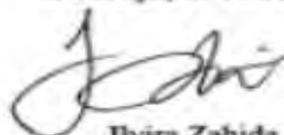
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "Remote Control Pembuka Pintu Pagor Otomatis yang Terintegrasi oleh *Smartphone* dengan Sistem Keamanan *Password* atau Sensor Sidik Jari" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 19 Juli 2017



Ilvira Zahida
NRP 2214030107

-----Halaman ini sengaja dikosongkan-----

***REMOTE CONTROL PEMBUKA PINTU PAGAR OTOMATIS
YANG TERINTEGRASI OLEH *SMARTPHONE* DENGAN
SISTEM KEAMANAN *PASSWORD* ATAU SENSOR SIDIK JARI***

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Bidang Studi Komputer Kontrol
Program Studi D3 Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing

Ir. Gatot Kusrahardjo, MT.
NIP. 1959 02 18 1986 10 1001

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

**REMOTE CONTROL PEMBUKA PINTU PAGAR OTOMATIS
YANG TERINTEGRASI OLEH *SMARTPHONE* DENGAN
SISTEM KEAMANAN *PASSWORD* ATAU SENSOR SIDIK JARI**

Nama : Ilvira Zahida
NRP : 2214030107

Pembimbing : Ir. Gatot Kusrahardjo, MT.
NIP : 195902181986101001

ABSTRAK

Adanya kepadatan penduduk dan kejahatan pencurian maupun perampokan yang semakin marak menjadikan masyarakat harus lebih hati-hati dalam pengamanan rumah mereka.

Dengan permasalahan tersebut maka adalah ide pembuatan alat ini. Perancangan alat ini bertujuan untuk mempermudah membuka pintu pagar dan efisiensi waktu karena dapat membuka secara otomatis. Cara kerja alat ini yaitu dengan menyambungkan *bluetooth* pada *smartphone* dan *bluetooth* pada arduino dahulu, setelah tersambung maka dapat memasukkan *password* kedalam aplikasi *smartphone*. Kemudian menyentuh pilihan '*Open Gate*' untuk membuka dan '*Close Gate*' untuk menutup. Dan jika pada *fingerprint*, cukup menempelkan jari untuk membuka atau menutup pagar dengan cara bergeser.

Hasil yang didapatkan dalam pembuatan alat ini adalah Alat ini mempunyai *bluetooth* yang dapat disambungkan dengan *bluetooth* pada *smartphone* dengan jarak kurang lebih 0,5 sampai 1 meter sesuai dengan uji coba yang telah dilakukan. Pagar akan membuka dengan cara bergeser dengan *delay* kurang lebih 1 detik apabila menggunakan *Fingerprint* dan akan menutup otomatis setelah *delay* 5 detik. Sedangkan apabila menggunakan *Smartphone* dengan aplikasi dilengkapi dengan *password* akan ada *delay* kurang lebih 15 detik setelah menyentuh pilihan '*Open gate*' maupun '*Close Gate*'.

Kata Kunci : *Fingerprint, Arduino, Smartphone*

-----Halaman ini sengaja dikosongkan-----

**REMOTE CONTROL PEMBUKA PINTU PAGAR OTOMATIS
YANG TERINTEGRASI OLEH SMARTPHONE DENGAN
SISTEM KEAMANAN PASSWORD ATAU SENSOR SIDIK JARI**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Bidang Studi Komputer Kontrol
Program Studi D3 Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing



[Handwritten Signature]
Ir. Gatot Kusrahardjo, MT.
NIP. 1959 02 18 1986 10 1001

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

**REMOTE CONTROL FOR OPENING GATE WHICH IS
INTEGRATED BY SMARTPHONE USING SECURITY
PASSWORD OR FINGERPRINT SYSTEM**

Name : Ilvira Zahida
Register Number : 2214030107

Supervisor : Ir. Gatot kusrahardjo, MT.
ID Number : 195902181986101001

ABSTRACT

The presence of population density and crime that is increasingly prevalent makes people should be more careful in securing their homes.

With these problems then is the idea of making this tool. Then is the idea of making this tool. The design of this tool for the purpose of opening the gate and efficiency of time because it can open automatically. How this tool works by connecting bluetooth on smartphones and bluetooth on arduino first, after connecting it can enter keywords into the smartphone application. Then meet the 'Open Gate' option to open and 'Close Gate' to close. And if on fingerprints, simply stick a finger to open or close the fence by shifting.

The results obtained in the manufacture of this tool is that this tool has bluetooth that can be connected with bluetooth on a smartphone with a distance of approximately 0.5 to 1 meter in accordance with the experiments that have been done. The fence opens with a delay of approximately 1 second using the Fingerprint and will close automatically after a 5 second delay. Being using the Smartphone with the application comes with a password will have a delay of approximately 15 seconds after filling options 'Open gate' and 'Close Gate'.

Keywords : *Fingerprint, Arduino, Smartphone*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Bidang Studi Komputer Kontrol, Jurusan Teknik Elektro, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

REMOTE CONTROL PEMBUKA PINTU PAGAR OTOMATIS YANG TERINTEGRASI OLEH SMARTPHONE DENGAN SISTEM KEAMANAN PASSWORD ATAU SENSOR SIDIK JARI

Dalam Tugas Akhir ini dirancang Remote Control Pembuka Pintu Pagar Otomatis yang dikontrol menggunakan *smartphone* dengan *password* atau dengan sensor sidik jari pada pagar berbasis Arduino

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Gatot Kusrahardjo, MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini, Angga Astanto yang selalu memberikan doa, semangat, dan dukungannya kepada penulis. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 19 Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN

HALAMAN JUDUL	i
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
ABSTRAK	ix
<i>ABSTRACT</i>	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Laporan	3
1.7 Relevansi	4
2 BAB II TEORI DASAR	5
2.1 Motor DC Toshiba DGM-202-2A [1]	5
2.2 <i>Bluetooth</i> HC-05 [2]	7
2.3 <i>Arduino Uno</i> R3 [3]	7
2.4 Sensor <i>Fingerprint</i> [4]	10
2.5 <i>Smartphone</i> [5]	11
2.6 <i>App Inventor</i> [6]	12
3 BAB III PerANCANGAN SISTEM KONTROL	13
3.1 Perancangan <i>Hardware</i>	14
3.1.1 Perancangan <i>Model</i> Pintu Pagar	14
3.1.2 <i>Flowchart</i> Sistem	15
3.1.3 Perancangan <i>Hardware</i>	18
3.1.4 Pengaturan Modul <i>Bluetooth</i> HC-05	18
3.1.5 Pengaturan Motor DC DGM-204-2A	21

3.2	Perancangan <i>Software</i>	21
3.2.1	Perancangan Program Pada Arduino	23
3.3	Perancangan Sistem <i>Back Up</i>	24
4	BAB IV HASIL SIMULASI DAN IMPLEMENTASI	25
4.1	Variabel dan Parameter yang Digunakan	25
4.2	Hasil Implementasi Pada <i>Fingerprint</i>	28
4.3	Hasil Pengambilan Data Pada <i>Fingerprint</i>	32
4.4	Hasil Pengambilan Data Pada Aplikasi <i>Smartphone</i>	36
4.5	Data Percobaan Jarak Aplikasi <i>Smartphone</i>	41
	BAB V	47
4.1	Hasil Pengambilan Data Pada Aplikasi <i>Smartphone</i>	47
a.	Saran	47
	DAFTAR PUSTAKA	49
G.	LAMPIRAN	51

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	Motor DC DGM-202-2A	6
Gambar 2.2	<i>Bluetooth</i> HC-05	7
Gambar 2.3	Arduino <i>Uno</i> R3.....	9
Gambar 2.4	Sensor <i>Fingerprint</i>	11
Gambar 2.5	Samsung <i>J5</i>	11
Gambar 2.6	Tampilan <i>App Inventor</i>	12
Gambar 3.1	Diagram Alur Keseluruhan Sistem	13
Gambar 3.2	Perancangan Mekanik Penggeser Pintu Pagar	14
Gambar 3.3	Desain Pintu Pagar	15
Gambar 3.4	<i>Flowchart</i> Sistem <i>Fingerprint</i>	16
Gambar 3.5	<i>Flowchart</i> Sistem <i>Smartphone</i>	17
Gambar 3.6	Perancangan <i>Hardware</i>	18
Gambar 3.7	Pengkabelan Modul <i>Bluetooth</i> HC-05 Dengan Arduino.....	20
Gambar 3.8	Rancangan Program Pada <i>App Inventor</i>	22
Gambar 3.9	Tampilan Hasil Program Pada <i>Smartphone</i>	22
Gambar 3.10	Tampilan <i>Software</i> Arduino <i>IDE</i>	23
Gambar 4.1	Tampilan <i>Smartphone</i>	25
Gambar 4.2	Tampilan <i>Smartphone</i>	26
Gambar 4.3	Tampilan <i>Smartphone</i>	26
Gambar 4.4	Tampilan <i>Smartphone</i>	27
Gambar 4.5	Tampilan <i>Smartphone</i>	27
Gambar 4.6	Uji Coba <i>Fingerprint</i> Sebelum Diberi Masukan Data	28
Gambar 4.7	Tampilan Arduino Diberi Masukan Data Sidik Jari.....	29
Gambar 4.8	Sketsa Ujicoba Posisi Awal Pagar	29
Gambar 4.9	Uji Coba Pada Sensor <i>Fingerprint</i>	30
Gambar 4.10	Tampilan Arduino Diberi Masukan Dua Kali.....	31
Gambar 4.11	Tampilan Uji Coba Kondisi Sensor	32
Gambar 4.12	Sketsa Ujicoba	32
Gambar 4.13	Sketsa uji Coba	33
Gambar 4.14	Tampilan Uji Coba Pada <i>Fingerprint</i>	34
Gambar 4.15	Sketsa Uji Coba	35
Gambar 4.16	Uji Coba Pada <i>Smartphone</i>	37

Gambar 4.17 Uji Coba Pada <i>Smartphone</i>	37
Gambar 4.18 Uji Coba Pada <i>Smartphone</i>	38
Gambar 4.19 Tampilan Uji Coba pada <i>Smartphone</i>	41
Gambar 4.20 Tampilan Uji Coba pada <i>Smartphone</i>	41
Gambar 4.21 Sketsa Uji Coba Jarak Pada Aplikasi <i>Smartphone</i>	42
Gambar 4.22 Sketsa Uji Coba Jarak Pada Aplikasi <i>Smartphone</i>	43

DAFTAR TABEL

	HALAMAN
Tabel 2.1 Spesifikasi Arduino <i>Uno</i>	9
Tabel 3.1 <i>Default</i> Status <i>Bluetooth</i> HC-05	19
Tabel 3.2 <i>Pin Mapping</i> Modul <i>Bluetooth</i> HC-05 Dengan Arduino Saat Perintah <i>AT Command</i>	20
Tabel 4.1 Percobaan Pada <i>Fingerprint</i>	35
Tabel 4.2 Data Pada Aplikasi <i>Smartphone</i>	39
Tabel 4.3 Data Percobaan Aplikasi <i>Smartphone</i>	44

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era perkembangan zaman yang semakin maju ini, terlahir banyak solusi yang dapat memecahkan permasalahan manusia. Permasalahan yang timbul akibat keterbatasan manusia ataupun dari faktor lain, kini sedikit demi sedikit sudah dapat diatasi. Salah satu solusi yang dapat memecahkan permasalahan manusia yaitu dengan menggunakan sistem kendali berbasis komputer. Dengan menggunakan sistem kendali berbasis komputer, diharapkan dapat membantu dan meringankan pekerjaan manusia serta menjadi solusi untuk setiap permasalahan manusia.

Seperti yang telah ada sebelumnya aplikasi berbasis Android sangat ramah dengan kehidupan sehari-hari yang mempermudah kinerja penggunanya untuk beradaptasi dengan lingkungan sekitar. Disamping perkembangan dunia telekomunikasi, dalam kehidupan sehari-haripun semakin banyak terjadi perkembangan dan perubahan, ditempat-tempat umum misalnya, sebelumnya pintu dibuka secara manual menggunakan kunci yang dari sisi keamanan cukup riskan karena bisa dengan mudah kuncinya untuk di duplikasi, kemudian berkembang dengan adanya pintu geser yang bisa dibuka dengan menggunakan tombol atau *remote control* sampai dengan akhirnya seiring perkembangan zaman, kini telah ada pintu geser otomatis menggunakan sensor gerak seperti yang ada di *mall*, bandara, dan tempat umum lainnya. Hal ini membuktikan bahwa semakin berkembangnya dunia *modern* sekarang ini.

Berdasarkan kondisi tersebut, diperlukan suatu sistem yang dapat mengontrol pintu pagar secara otomatis dalam jarak kurang dari 10 meter dan tetap menjaga keamanan. Dimana masyarakat sekarang pasti memiliki HP Android yang selalu dibawa kemanapun. Dengan begitu, akan lebih efisien dalam hal waktu juga aman karena memakai sistem *password*. Selain itu, jika menuju rumah dengan berjalan kaki maka tetap bisa membuka pintu pagar otomatis lebih cepat dari membuka HP dengan memakai *fingerprint*.

1.2 Permasalahan

Pada Tugas Akhir ini yang menjadi permasalahan utama adalah bagaimana membuka pagar otomatis secara efisien terhadap waktu dan tetap terkendali keamanannya. Jadi saat akan membuka (menggeser) pintu pagar rumah harus meletakkan sidik jari (ibu jari) pada *fingerprint* dahulu, lalu jika sidik jari yang dimasukkan sesuai maka pintu pagar rumah akan menggeser.

Sebenarnya, sudah ada inovasi tentang pembuka pagar otomatis yang seperti ini, berdasarkan TA yang telah dilakukan oleh kakak kelas kami yaitu Dimas Bagoes dan Wahyu Nanto yang berjudul “Pengendalian Gerbang Pintu Rumah Melalui Media *Bluetooth*. Dan Tugas Akhir lain yang dilakukan oleh kakak kelas kami yaitu Rio Dwi dan Ugan Sunarko tentang “Otomatisasi Buka Pintu Masuk Gudang Industri Menggunakan Sistem RFID Berbasis Mikrokontroler”. Dengan adanya dua Tugas Akhir tersebut, kami membuat inovasi baru dengan menambahkan sistem keamanan berupa sensor sidik jari atau *password*, dimana masih belum ada yang menggunakan sistem keamanan ini untuk pengaplikasiannya.

1.3 Tujuan

Penelitian ini bertujuan untuk merancang *Remote Control* Pintu Pagar Otomatis menggunakan *software* pada *smartphone*. Atau menggunakan *fingerprint* pada pintu pagar. Hasil yang diharapkan adalah *software* mampu membuka pintu dengan masukan data yang tepat dan waktu juga jarak yang akurat. Tujuan lainnya yaitu membuat Inovasi baru pada buka-tutup pintu pagar otomatis yang terintegrasi dengan *smartphone* dengan ditambahkan fitur sensor sidik jari dan pengaturan posisi. Mempermudah kegiatan konvensional manusia dengan teknologi terbaru. Menjadi bahan evaluasi dan inspirasi untuk orang lain bagi yang ingin meningkatkan proyek ini.

1.4 Batasan Masalah

Desain pagar pintu otomatis dengan penempatan motor menggunakan *sketch up*. Desain *software* pada *smartphone* menggunakan *App Inventor*. Alat ini menggunakan bahan Aluminium yang ringan. *Plant* yang digunakan bertujuan untuk mempermudah dalam pengaturan posisi. Alat ini hanya dapat digunakan untuk pagar geser, tidak untuk pagar lipat.

1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, studi literatur, perancangan sistem, perancangan algoritma kontrol, simulasi hasil desain, implementasi dan analisis data, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur akan dipelajari mengenai identifikasi fisik, *model* motor DC yang digunakan, *pemodelan* sistem, perancangan desain pintu pagar, analisis data motor dan data pada *fingerprint*, juga pada program yang ada pada *smartphone*, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur akan dipelajari mengenai identifikasi fisik, Kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi ruang terbuka diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet. *Current* Elektromagnet atau Dinamo. Dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. *Commutator* kegunaannya adalah untuk transmisi arus antara dinamo dan sumber daya. Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur. Motor akan berputar seiring dengan berjalannya program yang telah disetting pada Arduino. Arduino sendiri dihubungkan dengan motor dan *fingerprint*. Pertama analisis data pada motor, jika menggunakan tegangan 12V berapakah RPM dan amperenya, jika 24V dan 48V. Kedua buatlah perencanaan desain pintu pagar menggunakan *sketch up* sebagaimana ukuran 80cm x 40cm. Setelah itu analisis program yang akan dibuat di *App Inventor*. Dari data analisis akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penulisan laporan penelitian.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari motor DC secara garis besar, *fingerprint*, *App Inventor* dan *software* pada *smartphone*.

Bab III Perancangan Sistem

Bab ini membahas desain dan perancangan kontrol untuk membuka pintu pagar otomatis menggunakan *software* pada *smartphone*, kontrol dengan *fingerprint*.

Bab IV Simulasi, Implementasi dan Analisis Sistem

Bab ini memuat hasil simulasi dan implementasi serta analisis dari hasil tersebut.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan menjadi referensi perencanaan desain *Remote Control* pada pintu pagar otomatis, dengan sistem keamanan sensor sidik jari atau *password* yang terdapat pada *software* di *smartphone*.

BAB II

TEORI DASAR

Bab ini membahas mengenai teori dasar peralatan dari Tugas Akhir yang berjudul Remote Control Pembuka Pintu Pagar Otomatis yang Terintegrasi oleh *Smartphone* dengan Sistem Keamanan *Password* atau Sensor Sidik Jari yang meliputi Motor DC Toshiba DGM-202-2A, *Bluetooth* HC-05, *Arduino Uno R3*, sensor *Fingerprint*, *Smartphone*, *App Inventor*.

2.1 Motor DC Toshiba DGM-202-2A [1]

Motor DC adalah jenis motor listrik yang bekerja menggunakan sumber tegangan DC. Motor DC atau motor arus searah sebagaimana namanya, menggunakan arus langsung dan tidak langsung/*direct-unidirectional*. Motor DC digunakan pada alat ini adalah Motor DC seri DGM-202-2A merk Toshiba. Dengan spesifikasi yang tertera di seri motor yaitu dengan *gear ratio* (1:200). Motor ini dapat diberi tegangan 12 sampai 24 volt dengan kecepatan rotasi sebesar 22rpm. Arus 0,85A dan dapat memikul torsi sampai dengan 25kg. Panjang motor ini adalah 15,5cm termasuk *shaft*. *Gearbox* seukuran 6 x 6 x 3,8cm dan dengan berat sekitar 0,75kg. Torsi dan kecepatan pada motor ini mudah dikendalikan. Sistem kontrolnya relatif mudah dan sederhana. Prinsip kerja Motor DC adalah jika suatu arus melewati suatu konduktor, timbul medan magnet di sekitar konduktor. Medan magnet hanya terjadi di sekitar sebuah konduktor jika ada arus mengalir pada konduktor tersebut. Arah medan magnet ditentukan oleh arah aliran arus pada konduktor.

Untuk menentukan arah putaran motor digunakan kaedah *Flamming* tangan kiri. Kutub-kutub magnet akan menghasilkan medan magnet dengan arah dari kutub utara ke kutub selatan. Jika medan magnet memotong sebuah kawat penghantar yang dialiri arus searah dengan empat jari, maka akan timbul gerak searah ibu jari. Gaya ini disebut gaya *Lorentz*, yang besarnya sama dengan F. Prinsip motor : aliran arus di dalam penghantar yang berada di dalam pengaruh medan magnet akan menghasilkan gerakan. Besarnya gaya pada penghantar akan bertambah besar jika arus yang melalui penghantar bertambah besar.

Motor DC merupakan motor listrik yang memerlukan suplai tegangan arus searah pada kumparan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada Motor DC disebut dengan stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Keuntungan utama Motor DC adalah sebagai pengendali kecepatan, yang tidak memengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur tegangan dinamo dan arus medan (menurunkan arus medan akan meningkatkan kecepatan). Motor DC menggunakan arus langsung yang tidak langsung atau *direct-unidirectional*. Motor DC memiliki tiga bagian atau komponen utama, yaitu:

1. Kutub medan. Motor DC sederhana memiliki dua kutub medan, kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi ruang terbuka antara kutub-kutub dari utara menuju selatan. Untuk motor yang lebih kompleks, terdapat satu atau lebih elektromagnetik.
2. Dinamo atau current elektromagnetik. Dinamo ini berbentuk silinder yang dihubungkan ke as penggerak untuk menggerakkan beban. Dalam kasus Motor DC berukuran kecil, dinamo berputar dalam medan magnet yang dibentuk kutub-kutub hingga kutub utara dan selatan magnet berganti lokasi.
3. *Commutator*. Komponen ini berguna untuk transmisi arus antara dinamo dan sumber daya.



Gambar 2.1.Motor DC DGM-202-2A

Pada Gambar 2.1 merupakan tampilan Motor DC dengan kebutuhan arus sebesar 2 A. Dan kebutuhan tegangan sebesar 24 V.

2.2 *Bluetooth HC-05* [2]

Bluetooth Module HC-05 merupakan Modul komunikasi nirkabel pada frekuensi 2,4GHz dengan pilihan koneksi bisa sebagai *slave*, ataupun sebagai *master*. Dibuat dengan LED sebagai indikator koneksi *Bluetooth*. Dalam penggunaannya, HC-05 dapat beroperasi tanpa menggunakan *driver* khusus. Jarak efektif jangkauan sebesar 10 meter, meskipun dapat mencapai lebih dari 10 meter, namun kualitas koneksi makin berkurang. Untuk berkomunikasi antar *Bluetooth*, minimal harus memenuhi dua kondisi yaitu komunikasi harus antara *master* dan *slave*. Dan yang kedua *password* harus benar (saat melakukan *pairing*). Adapun spesifikasi *Hardware* dari HC-05 adalah :

- 1) Sensitivitas -80dBm (Typical)
- 2) Daya *transmit* RF sampai dengan +4dBm.
- 3) Kontrol PIO.
- 4) Antarmuka UART dengan *baudrate* yang dapat diprogram.
- 5) Dengan antenna terintegrasi.



Gambar 2.2 *Bluetooth HC-05*

Gambar 2.2 merupakan tampilan *Bluetooth HC-05* yang akan digunakan pada alat ini.

2.3 *Arduino Uno R3* [3]

Arduino Uno memiliki 14 *digital pin input/output* (atau biasa ditulis I/O, dimana 6 pin diantaranya dapat digunakan sebagai *output PWM*), 6 pin *input analog*, menggunakan *crystal* 16 MHz, koneksi USB, jack listrik, *header ICSP* dan tombol *reset*. Hal tersebut adalah semua yang diperlukan untuk mendukung sebuah rangkaian mikrokontroler. Cukup dengan menghubungkannya ke komputer dengan kabel USB atau diberi *power* dengan adaptor AC-DC atau baterai, sudah dapat bermain-main dengan *Arduino UNO* tanpa khawatir akan melakukan sesuatu yang salah. Kemungkinan paling buruk hanyalah kerusakan pada chip

ATMega328, yang bisa diganti sendiri dengan mudah dan dengan harga yang relatif murah. dan dipilih untuk menandai peluncuran *Software Arduino (IDE)* versi 1.0. Arduino. Sejak awal peluncuran hingga sekarang, *Uno* telah berkembang menjadi versi Revisi 3 atau biasa ditulis REV 3 atau R3.

Arduino *UNO* adalah sebuah *board* mikrokontroler yang didasarkan pada ATMega328. Arduino *UNO* mempunyai 14 pin *digital input/output* (6 diantaranya dapat digunakan sebagai *output* PWM), 6 *input* analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset*. Arduino *UNO* memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkan ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.

Arduino *UNO* berbeda dari semua *board* Arduino sebelumnya. Arduino *UNO* tidak menggunakan *chip driver* FTDI USB-to-serial. Sebaliknya, fitur-fitur ATmega16U2 (ATmega8U2 sampai ke versi R2) deprogram sebagai pengubah USB ke *serial*. Revisi 2 dari *board* Arduino *UNO* mempunyai sebuah resistor yang menarik gaya 8U2 HWB ke ground, yang membuatnya lebih mudah untuk diletakkan ke dalam DFU *mode*. Revisi 3 dari *board* Arduino *UNO* memiliki fitur-fitur baru sebagai berikut:

1. *Pinout* 1.0: ditambah pin SDA dan SCL yang dekat dengan pin AREF dan dua pin baru lainnya yang diletakkan dekat dengan pin *RESET*, IOREF yang memungkinkan *shield-shield* untuk menyesuaikan tegangan yang disediakan dari *board*. Untuk ke depannya, *shield* akan dijadikan kompatibel/cocok dengan *board* yang menggunakan AVR yang beroperasi dengan tegangan 5V dan dengan Arduino Due yang beroperasi dengan tegangan 3,3V.
2. Sirkuit *RESET* yang lebih kuat.
3. ATmega16U2 menggantikan 8U2.
4. Arduino *UNO* adalah seri terakhir dari *board* Arduino USB dan *model* referensi untuk papan Arduino, untuk suatu perbandingan dengan versi sebelumnya.

Berikut adalah ringkasan spesifikasi dari Arduino *UNO* pada Tabel 2.1:

Tabel 2.1 Spesifikasi Arduino *Uno*

Mikrokontroler	ATmega328
Tegangan Pengoperasian	5 Volt
Tegangan <i>input</i> yang disarankan	7-12 Volt
Batas tegangan <i>input</i>	6-20 Volt
Jumlah pin I/O <i>digital</i>	14 (6 diantaranya menyediakan keluaran PWM)
Jumlah pin <i>input</i> analog	6
Arus DC tiap pin I/O	40mA
Arus DC untuk pin 3,3 Volt	50 mA
Memori <i>Flash</i>	32 KB (ATmega328), sekitar 0,5 KB digunakan oleh <i>bootloader</i>
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
<i>Clock Speed</i>	16 MHz



Gambar 2.3 Arduino *Uno* R3

Pada Gambar 2.3 merupakan tampilan Arduino Uno R3 dengan spesifikasi yang telah dijelaskan pada Tabel 2.1

2.4 Sensor *Fingerprint* [4]

Sensor *Fingerprint* adalah sebuah perangkat elektronik yang digunakan untuk menangkap Gambar *digital* dari pola sidik jari. Gambar tersebut disebut pemindaian. Pemindaian adalah proses *digital* untuk membuat sebuah *template* yang disimpan dan digunakan untuk pencocokan. ini merupakan contoh dari beberapa sidik jari yang lebih umum digunakan sensor teknologi. Sensor ini memiliki spesifikasi suplai tegangan 3,6-6 V. Arus maksimal 120mA. Waktu pemindaian sidik jari 1s. *Mode* pencarian 1:N. Kapasitas penyimpanan 162 *template*. Tingkat penerimaan kesalahan 0,001%, Tingkat penolakan kesalahan 1%. *Baudrate*: 9600, 19200, 28800, 38400, 57600bps (*default* adalah 57600). *Interface* : *Serial TTL*. Suhu kerja: -20 ~ +50°C.

Sensor yang digunakan untuk mendeteksi sidik jari menggunakan sistem optikal, dimana pendeteksian dilakukan dengan pembacaan kontur (tinggi rendahnya permukaan) sidik jari dan listrik statis tubuh. Hal ini menghasilkan tingkat keamanan yang tinggi karena tidak dapat dipalsukan dengan *fotocopy* sidik jari atau sidik jari tipuan. Sensor ini memiliki lapisan kaca yang tahan lama dan juga memiliki sensor gerak, yaitu jika ada jari yang menempel pada sensor maka sensor akan langsung menyala untuk mengambil sidik jari. Pada kondisi tidak dipakai maka sensor akan padam, dengan demikian usia sensor akan lebih awet karena sensor hanya menyala pada saat dipakai saja.

Pada umumnya komponen dalam *fingerprint* terdiri dari beberapa komponen pokok, dan komponen tambahan. komponen pokok diantaranya:

1. *Mainboard*
2. RAM
3. Sensor/Kaca Prisma
4. Layar *display*
5. *Speaker*

Dari ke-5 komponen tersebut, yang paling penting adalah RAM, karena semua data dan pengaturan yang ada di *fingerprint* terdapat didalam RAM tersebut. Jadi jika terjadi kerusakan pada RAM, biasanya data yang ada didalamnya tidak bisa diambil lagi. untuk mencegah hal

tersebut, sebaiknya *dibackup* data sidikjari, sehingga jika terjadi kerusakan pada RAM, tidak perlu khawatir mengenai data *template* sidikjari tidak harus registrasi ulang.



Gambar 2.4 Sensor *Fingerprint*

Pada Gambar 2.4 merupakan tampilan Sensor Sidikjari atau *Fingerprint* yang akan digunakan pada alat ini.

2.5 Smartphone [5]

Smartphone adalah telepon pintar yang memiliki kemampuan seperti komputer. *Smartphone* diklasifikasikan sebagai *high end mobile phone* yang dilengkapi dengan kemampuan *mobile computing*. Dengan kemampuan *mobile computing* tersebut, *smartphone* memiliki kemampuan yang tak bisa dibandingkan dengan ponsel biasa. *Smartphone* yang saya pakai disini adalah *smartphone merk* Samsung dengan tipe *Galaxy J5*



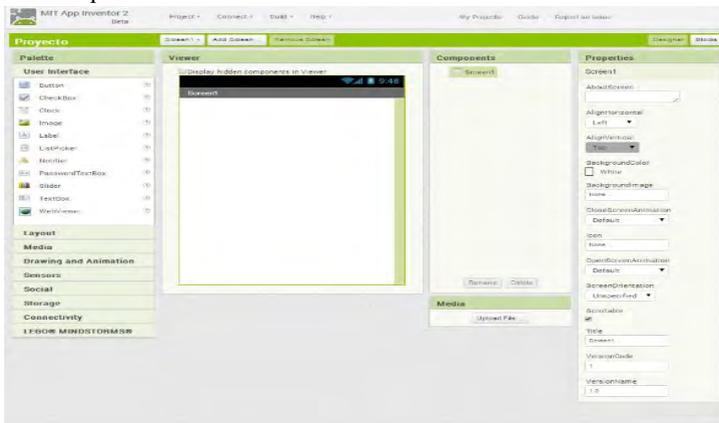
Gambar 2.5 Samsung *J5*

Pada Gambar 2.5 merupakan tampilan *Smartphone* Samsung *J5* yang digunakan pada Alat ini.

2.6 App Inventor [6]

App Inventor adalah aplikasi *web* sumber terbuka yang awalnya dikembangkan oleh *Google*, dan saat ini dikelola oleh *Massachusetts Institute of Technology (MIT)*. *App Inventor* memungkinkan pengguna baru untuk memprogram komputer untuk menciptakan aplikasi perangkat lunak bagi sistem operasi *Android*. *App Inventor* adalah sebuah tool untuk membuat aplikasi *Android*, yang spesial dari *tool* ini adalah karena berbasis visual *block programming*, jadi kita bisa membuat aplikasi tanpa kode satupun. Disebut visual *block programming* karena kita akan melihat, menggunakan, menyusun dan *drag-drops* “blok” yang merupakan simbol-simbol perintah dan fungsi – *event handler* tertentu dalam membuat aplikasi, dan secara sederhana kita bisa menyebutnya tanpa menuliskan kode program – *coding less*.

App Inventor tidak hanya untuk membuat aplikasi. Untuk *programmer* tentu ada opsi-opsi *advance* untuk membuatnya sesuai dengan level kita. *Framework* visual *programming* ini terkait dengan bahasa pemrograman *Scratch* dari MIT, yang secara spesifik merupakan implementasi dari *Open Block* yang didistribusikan oleh MIT Scheller *Teacher Education Program* yg diambil dari riset yang dilakukan oleh *Ricarose Roque*.



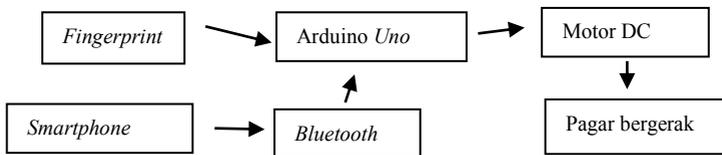
Gambar 2.6 Tampilan *App Inventor*

Pada Gambar 2.6 merupakan tampilan *App Inventor* menggunakan *Kawa Language Framework* dan *Kawa's dialect* – yg di develop oleh *Per Bothner* dan di distribusikan sebagai bagian dari *GNU Operating*.

BAB III

PERANCANGAN SISTEM KONTROL

Pada bab ini akan dibahas mengenai perencanaan dan pembuatan alat Tugas Akhir, yang meliputi perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*) yang menunjang alat Tugas Akhir bekerja dengan baik. Diagram fungsional dari keseluruhan sistem *Remote Control* Pembuka Pintu Pagar Otomatis yang Terintegrasi oleh *Smartphone* dengan sistem keamanan *Password* dan Sensor sidik jari. Megenai program yang dibuat pada *App Inventor* dan Program *Fingerprint* yang terhubung dengan Arduino.



Gambar 3.1 Diagram Alur Keseluruhan Sistem

Pada Gambar 3.1 merupakan Diagram Alur Keseluruhan Sistem. Cara kerja pertamanya adalah motor dan Arduino sudah harus terpasang dengan *power supply*, yaitu stop kontak. Pastikan Arduino dan *Bluetooth* sudah tersambung dengan benar dan bekerja dengan baik. Kemudian nyalakan *smartphone* dan buka *software* yang ada pada *smartphone*, hidupkan *Bluetooth* pada *smartphone* dan sambungkan dengan *Bluetooth* yang tersambung dengan Arduino. Setelah terhubung akan muncul layar masukkan *password*, setelah memasukkan *password* akan menampilkan fitur lanjutan yaitu membuka atau menutup pagar. Jika menyentuh pilihan '*open gate*' maka pintu pagar akan terbuka dan jika menekan '*close gate*' maka pintu pagar akan menutup. Motor DC akan berputar sesuai dengan perintah yang diberikan pada *software smartphone*.

Sedangkan perancangan pada *fingerprint*, pertama tempelkan jari kita pada sensor *fingerprint*, kemudian data akan diterima dan diolah, apabila benar maka pintu akan otomatis terbuka. Jika data yang diterima tidak ada atau tidak benar, maka pintu tidak akan tertutup. Pintu akan

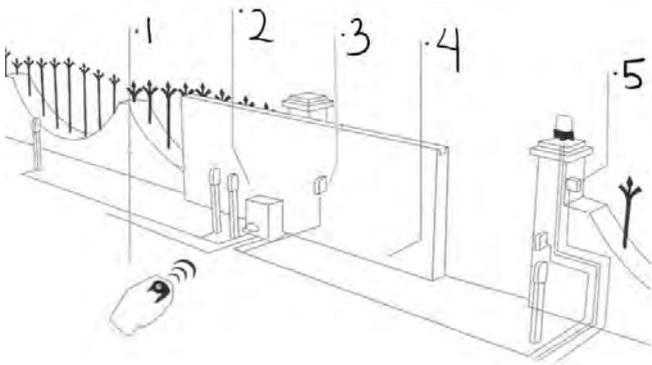
tertutup apabila kita memasukkan *input* kembali (menempelkan jari ke *fingerprint*), pintu otomatis tertutup.

3.1 Perancangan *Hardware*

Pada perancangan *hardware* meliputi perancangan *model* pintu pagar, pengaturan dan pengkabelan Modul *Bluetooth* HC-05 dengan *Arduino Uno R3*

3.1.1 Perancangan *Model* Pintu Pagar

Kegiatan ini dilakukan dengan perancangan serta perancangan baik dari rancang bangun bagian elektrik ataupun mekaniknya. Setelah dilakukan perancangan dan struktur dari sistem maka selanjutnya akan dilakukan pembuatan *hardware* dari sistem tersebut. Berikut adalah Gambaran dari sistem mekaniknya.



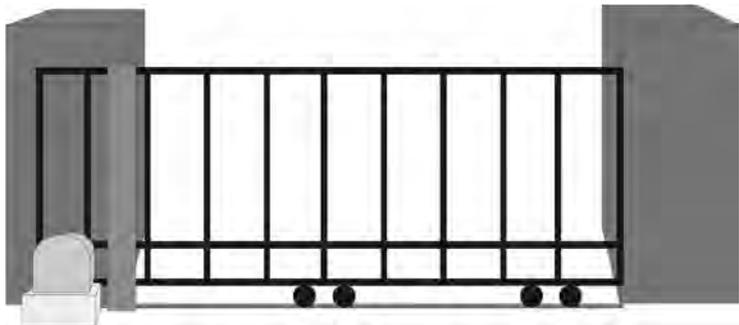
Gambar 3.2 Perancangan Mekanik Penggeser Pintu Pagar

Keterangan pada Gambar 3.2:

- I. *Smartphone*
- II. *Motor*
- III. *Fingerprint*
- IV. *Gear Rack*
- V. *Tombol Manual*

Berdasarkan Gambar 3.2 penggeser pintu pagar diletakkan pada punggung pagar terpisah pada Modul *Bluetooth* yang diletakkan di atas pilar agar tidak tertutup pagar sehingga tidak mengganggu transmisi

data. Selain itu juga terdapat *rack gear* yang dapat mengkonversi gerakan rotasi menjadi gerakan linear pada motor. Bahan pintu pagar yang dipilih adalah aluminium, karena teksturnya yang kuat namun tetap ringan. Desain yang dibuat adalah simpel agar memudahkan dalam masa pembuatannya dan juga menghemat biaya yang dikeluarkan. Pintu pagar ini memiliki ukuran 80 x 40cm dengan panjang rel kurang lebih 160 cm. Pemasangan motor diletakkan di belakang pintu pagar beserta rangkaiannya, sedangkan pemasangan *fingerprint* berada diluar pagar, yaitu didepan tembok yang berada disamping pintu pagar agar memudahkan pengguna untuk mengaksesnya.



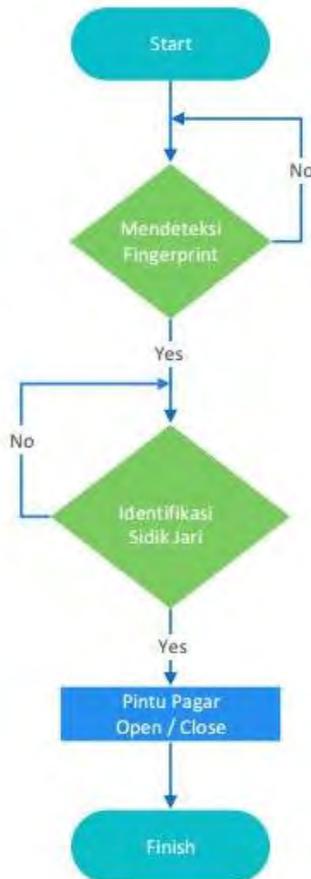
Gambar 3.3 Desain Pintu Pagar

Berdasarkan Gambar 3.3 menampilkan desain pintu pagar yang dirancang menggunakan *software Sketch Up*. Dimana pintu pagar akan bergerak dengan cara bergeser menggunakan roda dibawah pagar. Motor akan diletakkan di bagian kiri bawah dibelakang tembok. Dua balok berwarna abu abu adalah tembok. Sebuah balok kecil di samping kiri merupakan balok untuk menempatkan motor. Motor akan aman terlindungi apabila ada hujan dengan menggunakan balok tersebut. Dibagian bawah pagar terdapat lintasan roda yang disebut *rack gear*. Pagar akan bergeser diatas lintasan tersebut.

3.1.2 Flowchart Sistem

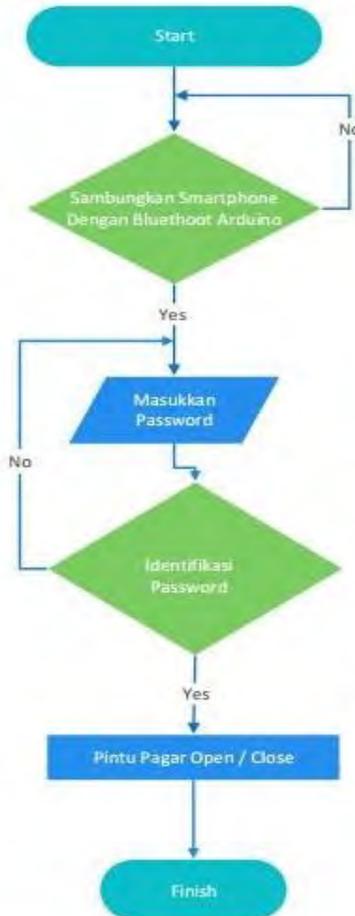
Pada Gambar 3.4 merupakan *Flowchart* sistem *Fingerprint*. Dari start untuk memulai sistem, kemudian menunggu data yang artinya kita memasukkan data berupa sidik jari yang nantinya akan diproses dan diidentifikasi, apabila cocok maka pagar akan terbuka dan apabila tidak cocok maka pagar tidak akan terbuka. Dan akan kembali

lagi ke menunggu data. Data yang dimaksud adalah data sidik jari yang akan diproses oleh sensor sidik jari. Kemudian akan diidentifikasi, jika data sidik jari yang dimasukkan ke sensor *fingerprint* sesuai dengan nomor ID yang telah tercantum maka arduino akan memproses data tersebut dan motor akan berputar sehingga pagar dapat bergeser membuka atau menutup.



Gambar 3.4 *Flowchart* Sistem *Fingerprint*

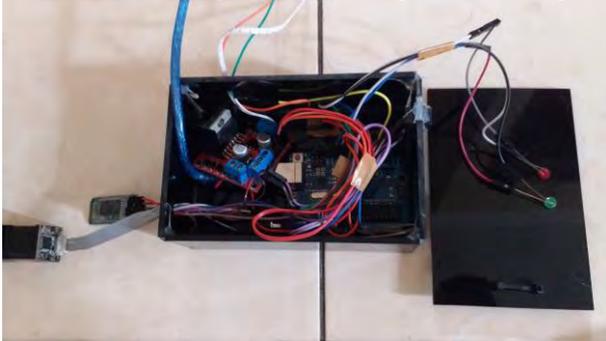
Pada Gambar 3.5 merupakan *Flowchart* Sistem *Smartphone*. Dari start untuk memulai sistem, kemudian menunggu data yang artinya kita memasukkan data berupa *password*, apabila kita memasukkan *password* secara tidak benar maka proses identifikasi tidak akan berjalan dan pagar tidak akan terbuka kemudian kembali lagi ke menunggu data. Jika *password* yang kita masukkan cocok maka pagar akan terbuka atau tertutup.



Gambar 3.5 *Flowchart* Sistem *Smartphone*

3.1.3 Perancangan *Hardware*

Berikut ini adalah gambar perancangan *Hardware* pada alat ini.



Gambar 3.6 Perancangan *Hardware*

Pada Gambar 3.6 menampilkan bentuk fisik *hardware* yang dipakai pada alat ini. Kotak tersebut dapat ditutup sehingga tidak menampilkan kabel-kabarnya.

3.1.4 Pengaturan Modul *Bluetooth HC-05*

Modul *Bluetooth HC-05* merupakan Modul komunikasi nirkabel pada frekuensi 2,4GHz dengan pilihan koneksi sebagai master maupun slave. Modul ini memiliki enam pin diantaranya :

- 1) EN fungsinya untuk mengaktifkan *mode AT Command Setup* pada Modul HC-05. Jika pin ini ditekan sambil ditahan sebelum memberikan tegangan ke Modul HC-05, maka Modul akan mengaktifkan *mode AT Command Setup*. Secara *default*, Modul HC-05 aktif dalam *mode Data*.
- 2) Vcc adalah pin yang berfungsi sebagai *input* tegangan. Hubungkan pin ini dengan sumber tegangan 5V.
- 3) GND adalah pin yang berfungsi sebagai *ground*. Hubungkan pin ini dengan *ground* pada sumber tegangan.
- 4) TX adalah pin yang berfungsi untuk mengirimkan data dari Modul ke perangkat lain (mikrokontroler). Tegangan sinyal pada pin ini adalah 3,3V sehingga dapat langsung dihubungkan dengan pin RX pada Arduino karena tegangan sinyal 3,3V dianggap sebagai sinyal bernilai *high* pada Arduino.

- 5) RX adalah pin yang berfungsi untuk menerima data yang dikirim ke Modul HC-05. Tegangan sinyal pada pin sama dengan tegangan sinyal pada pin TX, Untuk keamanan, sebaiknya gunakan pembagi tegangan jika menghubungkan pin ini dengan Arduino yang bekerja pada tegangan 5V. Pembagi tegangan tersebut menggunakan 2 buah resistor. Resistor yang digunakan sebagai pembagi tegangan pada tutorial ini adalah 1K ohm dan 2K ohm.
- 6) *STATE* adalah pin yang berfungsi untuk memberikan informasi apakah Modul terhubung atau tidak dengan perangkat lain.

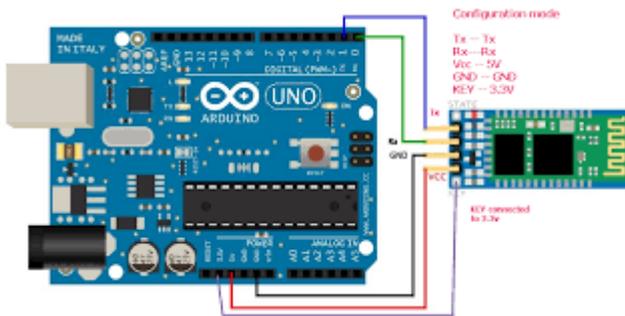
Meskipun terdapat enam pin, namun *interface* yang digunakan adalah *serial* VCC, GND, TXD dan RXD. Selain itu, Modul ini dilengkapi dengan LED yang berfungsi sebagai indikator koneksi *Bluetooth*. LED akan menyala apabila *Bluetooth* berfungsi.

Tabel 3.1 Default Status *Bluetooth* HC-05

<i>Device Type</i>	0
<i>Inquire code</i>	0x009e8b33
<i>Module work mode</i>	<i>Slave mode</i>
<i>Connection mode</i>	<i>Connect to the Bluetooth device specified</i>
<i>Serial parameter</i>	<i>Baudrate: 38400 bits/s; stop bit: 1 bit; Parity bit: None</i>
<i>Passkey</i>	1234
<i>Device name</i>	H-C-2010-06-01
<i>Default Baudrate</i>	9600, <i>Data bit 8, Stop bit 1; parity 0</i>

Pada Tabel 3.1 merupakan Tabel spesifikasi yaitu *Default Status Bluetooth* HC-05. Dimana Tabel 3.1 menjelaskan tentang spesifikasi *Bluetooth* secara lengkap.

Dalam pengaplikasiannya, Modul *Bluetooth* HC-05 membutuhkan di *setting* terlebih dahulu. Prosedur pengaturan tersebut menggunakan perintah *AT* atau yang biasa disebut dengan *AT command*. Sebelum menggunakan perintah *AT command*, Modul *Bluetooth* HC-05 harus dihubungkan terlebih dahulu dengan Arduino. Berikut ini merupakan *mapping* Modul *Bluetooth* HC-05 dengan Arduino.



Gambar 3.7 Pengkabelan Modul *Bluetooth* HC-05 Dengan Arduino

Pada Gambar 3.7 merupakan Pengkabelan Modul *Bluetooth* HC-05 dengan Arduino. Dengan keterangan konfigurasi kabel rangkaian :

- Pin 3,3V pada Arduino dihubungkan ke VCC pada HC-05
- Pin 3,3V pada Arduino dihubungkan ke *Pin Key*(34) pada HC-05
- Pin GND pada Arduino dihubungkan ke GND pada HC-05
- Pin RX pada Arduino dihubungkan ke TXD *Bluetooth* Modul
- Pin TX pada Arduino dihubungkan ke RXD *Bluetooth* Modul

Tabel 3.2 *Pin Mapping* Modul *Bluetooth* HC-05 Dengan Arduino Saat Perintah *AT Command*

No.	Pin Modul HC-05	Pin Arduino
1.	VCC	VCC
2.	GND	GND
3.	TXD	TXD-3
4.	RXD	RXD-3

Pada Tabel 3.2 merupakan penjelasan tentang *Pin Mapping* Modul *Bluetooth* HC-05 Dengan Arduino Saat Perintah *AT Command*. Selanjutnya, mengatur perintah *AT Command* pada *serial* monitor program Arduino *IDE*. Untuk mengetahui apakah Modul *Bluetooth* siap digunakan, maka dilakukan pengujian dengan cara megetikkan perintah ‘AT’ pada *serial* monitor. Apabila Modul tersebut siap digunakan maka akan muncul respon ‘OK’ pada *serial* monitor. Kemudian, mengatur

baudrate yang akan digunakan. Perancangan *software* dalam alat ini yaitu pembuatan *code program* pada Arduino IDE.

Untuk melakukan transfer data antara alat dan *smartphone* Android, dibutuhkan komunikasi data menggunakan *bluetooth*. Agar tujuan dari komunikasi data dapat tercapai, maka harus dilakukan penghubungan antara perangkat *bluetooth* pada alat dengan perangkat *bluetooth* pada *smartphone*. Keduanya juga harus terhubung dengan arduino dimana juga membuat program pada Arduino itu sendiri. Yang pertama harus dilakukan yakni mengaktifkan kedua perangkat *Bluetooth*, selanjutnya melakukan pencarian perangkat *bluetooth* yang akan dihubungkan menggunakan aplikasi android pada *smartphone*. Apabila kedua perangkat *bluetooth* telah terhubung, maka data yang ada pada alat akan tampil pada aplikasinya, namun jika belum terhubung, maka lakukan pengoneksian ulang.

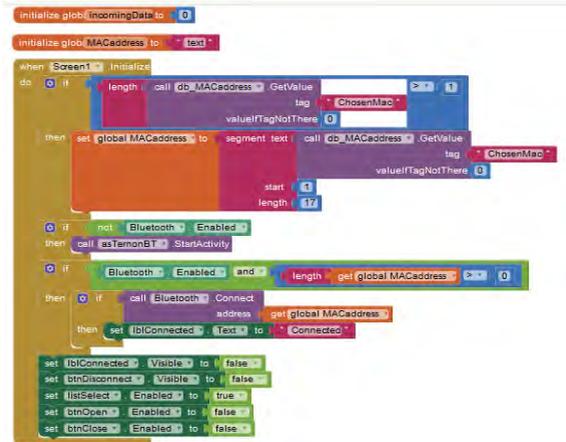
3.1.5 Pengaturan Motor DC DGM-204-2A

Motor *Direct Current* (DC) adalah jenis motor yang paling sederhana, yang memiliki dua buah kabel, yaitu catu daya dan *ground*. Pemberian catu daya boleh dibolak balik untuk memberikan efek arah putaran yang berbeda. motor akan berputar terus selama catu daya diberikan dan berhenti kalau catu daya diputus. motor jenis ini biasanya digunakan pada kipas angin atau untuk menggerakkan roda mobil mainan. Pada kali ini, motor DC digunakan untuk menggerakkan pintu pagar Otomatis. Motor DC memiliki dua bagian penting. Bagian pertama adalah stator, yaitu bagian yang tidak berputar dan bagian kedua adalah rotor, yaitu bagian yang berputar. Di rotor inilah, poros diletakan. Di ujung poros dapat dipasang objek (misalnya puli) yang ingin diputar.

3.2 Perancangan Software

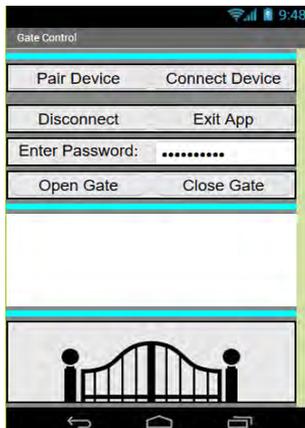
Pada perancangan *software* untuk alat ini dibagi menjadi beberapa bagian. yaitu perancangan *software* untuk alat dan komunikasi data menggunakan *bluetooth*, serta perancangan *software* untuk aplikasi android menggunakan *App Inventor* dan *software* Arduino IDE. Dan yang terakhir perancangan program untuk *fingerprint* yang terhubung dengan arduino. Kedua program ini sama sama menggunakan *bluetooth* HC-05 sebagai media penghubungnya.

Berikut adalah gambaran rancangan program menggunakan *App Inventor* yang menghubungkan antara *Smartphone* dengan Arduino dan motor atau program yang menghubungkan antara *fingerprinth* dengan Arduino.



Gambar 3.8 Rancangan Program Pada *App Inventor*

Pada Gambar 3.8 merupakan Perancangan *software* yang dibuat menggunakan *App Inventor* yang nantinya akan dijalankan dan kemudian akan menampilkan tampilan di layar *smartphone*.

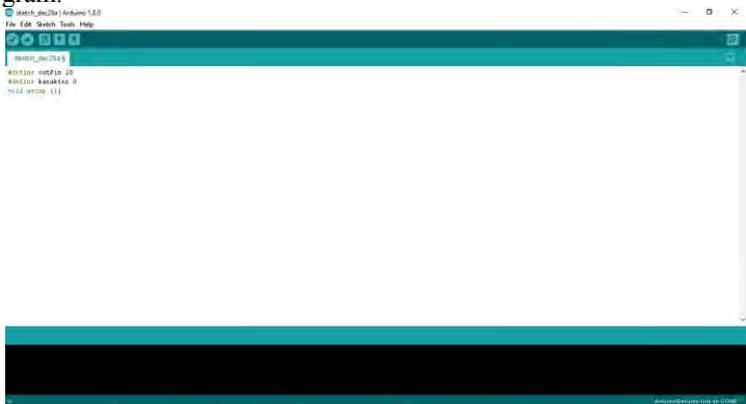


Gambar 3.9 Tampilan Hasil Program Pada *Smartphone*

Pada Gambar 3.9 menampilkan Hasil Program pada *smartphone* yang telah dibuat di *App Inventor*. Pilihan-pilihan yang ada pada Gambar 3.9 meliputi ;*Pair Device*' yang berfungsi untuk menghubungkan *bluetooth* pada *smartphone* dengan *bluetooth* pada *arduino*. *Disconnect* berfungsi untuk memutus sambungan *bluetooth*. *Exit App* berfungsi untuk keluar dari aplikasi. '*Enter Password*' berfungsi untuk memasukkan *password* di kolom disampingnya. '*Open gate*' berfungsi untuk membuka pagar dan '*Close Gate*' berfungsi untuk menutup pagar.

3.2.1 Perancangan Program Pada Arduino

Perancangan program pada Arduino ini berkaitan dengan pembuatan program untuk alat serta untuk pengiriman data oleh Modul *Bluetooth* dari alat untuk ditampilkan pada aplikasi *Android*. Langkah awal dalam suatu pemrograman adalah deklarasi variabel dan tipe data yang digunakan. Variabel dan tipe data ini digunakan untuk memudahkan dalam memprogram perangkat apa saja yang ingin diprogram.



Gambar 3.10 Tampilan *Software Arduino IDE*

Pada Gambar 3.10 merupakan Tampilan awal *Software Arduino IDE* ketika akan membuat program. Tiga baris instruksi pada Gambar 3.10 adalah instruksi *default* pada *software Arduino IDE*.

3.3 Perancangan Sistem *Back Up*

Berikut ini ada dua sistem *Back Up* yang diperuntukkan apabila sistem utama ada kesalahan baik dari dalam maupun dari luar.

- 1) Tombol Manual Pada Sisi Bagian Dalam Pagar
Pada sisi bagian dalam pagar diberi saklar atau tombol manual untuk membuka atau menutup pintu pagar dari dalam (sudah masuk ke halaman rumah). Tombol tersebut dimaksudkan agar tidak usah memasukkan *password* melalui aplikasi *smartphone*.
- 2) Generator Untuk Genset Listrik
Genset ini diperlukan apabila listrik padam. Motor tidak akan bergerak apabila tidak ada *supply* tegangan. Maka dari itu generator akan bekerja saat listrik padam sehingga apabila listrik dari PLN padam maka bisa menggunakan generator untuk suplai tegangan.
- 3) Pembuka Manual
Pembuka Manual yang dimaksudkan disini adalah kunci biasa, pembuka biasa. Dapat membuka pagar pintu dengan menggunakan tangan dari dalam pagar di halaman rumah) dengan menggeser pintu pagar secara manual.

BAB IV

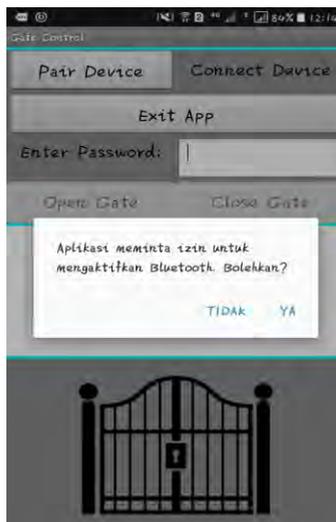
HASIL SIMULASI DAN IMPLEMENTASI

Bab ini dibagi menjadi beberapa bagian. Pada bagian pertama dijelaskan mengenai variabel serta parameter yang digunakan dalam simulasi dan implementasi. Pada bagian kedua akan diberikan penjelasan mengenai hasil simulasi pada Arduino, dan *software* pada Android. Dan bagian yang terakhir akan dijelaskan mengenai hasil implementasi pada *fingerprint*.

4.1 Variabel dan Parameter yang Digunakan

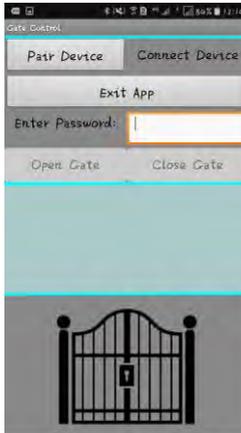
Simulasi dilakukan dengan mengambil kondisi awal untuk posisi pagar pintu yang tertutup. Adapun kondisi awal untuk motor akan ditentukan setelah mengamati hasil simulasi. Pemberian kondisi awal pada simulasi dilakukan dengan memberikan masukan data *password* kedalam *software* pada *smartphone*. Setelah memasukkan data, *smartphone* akan memproses data tersebut dan disalurkan ke Arduino, dari Arduino akan diproses datanya dan motor akan berputar.

Berikut adalah tampilan pada program di Arduino sebelum diberi masukan data:



Gambar 4.1 Tampilan Smartphone

Pada Gambar 4.1 merupakan tampilan pada program di *Smartphone* sebelum dihubungkan dengan *Bluetooth* yang tersambung dengan Arduino. Nyalakan *Bluetooth Smartphone* terlebih dahulu.

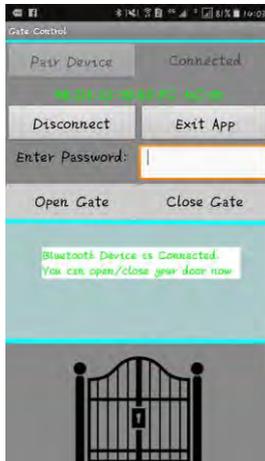


Gambar 4.2 Tampilan *Smartphone*

Pada Gambar 4.2 merupakan tampilan ketika *bluetooth* sudah dinyalakan. Kemudian klik pilihan *connected*. Setelah itu, pada Gambar 4.3 menampilkan pilihan *bluetooth* manakah yang akan disambungkan dengan *smartphone*. Pilih HC-05 dan sambungkan. *Step* ini adalah saat menyambungkan *bluetooth* pada *smartphone* dengan *bluetooth* pada arduino

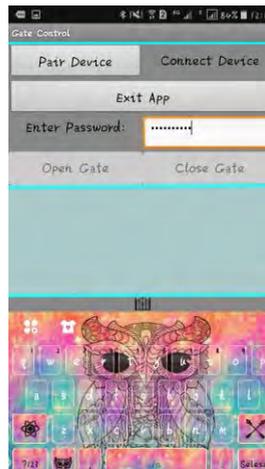


Gambar 4.3 Tampilan *Smartphone*



Gambar 4.4 Tampilan *Smartphone*

Pada Gambar 4.4 merupakan tampilan pada *smartphone* saat *bluetooth* pada *smartphone* sudah terhubung dengan *Bluetooth* pada *Arduino*. Kita dapat mengetahui apakah *bluetooth* sudah tersambung yaitu dengan adanya notifikasi '*Bluetooth Device is Connected. You can open/close your door now*' berwarna hijau pada layar *smartphone*.



Gambar 4.5 Tampilan *Smartphone*

Pada Gambar 4.5 merupakan tampilan saat akan memasukkan *password*. Masukkan *password* kemudian klik ‘*Open gate*’ untuk membuka Pintu pagar atau ‘*Close Gate*’ untuk menutup pintu Pagar. Pada langkah ini, ketika memasukkan *password* akan ada *delay* selama kurang lebih 15 detik untuk menunggu data dan pagar akan bergeser. Saat memasukkan *password* dengan benar maka pagar akan bergeser, namun bila memasukkan *password* tidak benar maka pagar tidak akan bergeser.

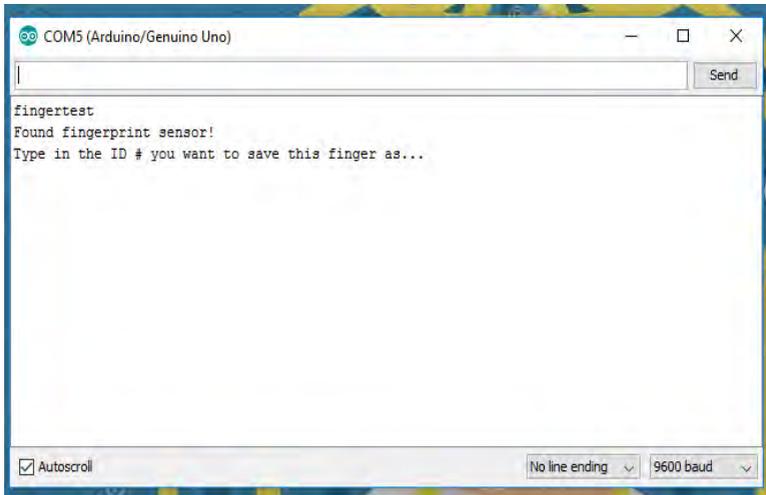
4.2 Hasil Implementasi Pada *Fingerprint*

Simulasi dilakukan dengan mengambil kondisi awal untuk posisi pagar pintu yang tertutup. Adapun kondisi awal untuk motor akan ditentukan setelah mengamati hasil simulasi. Pemberian kondisi awal pada simulasi dilakukan dengan memberikan masukan data sidik jari pada sensor sidik jari. Setelah memasukkan data, *fingerprint* akan memproses data tersebut dan disalurkan ke arduino, dari arduino akan diproses datanya dan motor akan berputar.



Gambar 4.6 Uji Coba *Fingerprint* Sebelum Diberi Masukan Data

Pada Gambar 4.6 merupakan uji coba *fingerprint* dengan arduino sebelum diberi masukan data. Ketika *bluetooth* aktif, LED pada *bluetooth* akan menyala. Dan lampu LED pada alat juga menyala berwarna merah. Berkaitan dengan tampilan pada *software* Arduino IDE pada Gambar 4.7 ketika *fingerprint* belum diberi masukan data.



Gambar 4.7 Tampilan Arduino Diberi Masukan Data Sidik Jari

Pada Gambar 4.7 merupakan tampilan *software* arduino IDE saat diberi masukan data berupa sidik jari dengan cara menempelkan jari ke sensor sidik jari satu kali kemudian menunggu selama beberapa detik tanpa diangkat untuk diidentifikasi karakteristik sidik jarinya. Setelah itu barulah data dapat diterima oleh sensor sidik jari dan disimpan.

Hasil yang didapatkan dari ujicoba pada Gambar 4.6 dan 4.7 adalah *bluetooth* aktif dengan indikator lampu LED pada *bluetooth* yang menyala dan lampu LED pada alat menyala berwarna merah yang berarti sudah tersambung namun belum diberi masukan data.



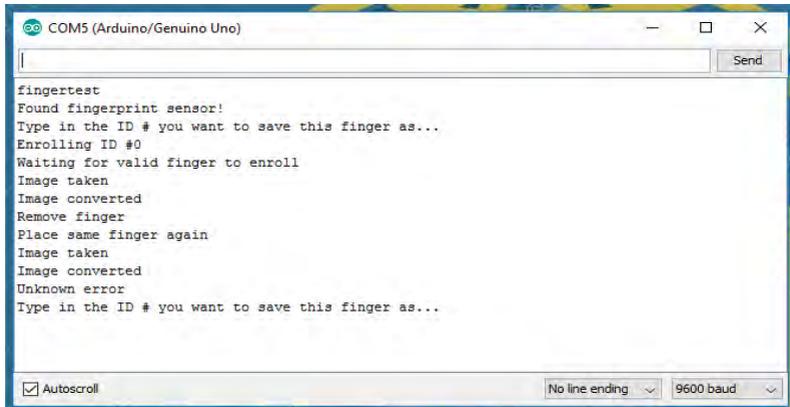
Gambar 4.8 Sketsa Ujicoba Posisi Awal Pagar

Pada Gambar 4.8 merupakan tampilan uji coba ketika memberi masukan data berupa sidik jari pada sensor *fingerprint*. Ketika memasukkan data sidik jari, lampu LED akan berkedip, jika LED berwarna merah padam maka lampu berwarna hijau akan menyala dan itu berarti data telah sukses diterima oleh sensor *fingerprint* dan tersimpan. Jika terjadi eror, maka kedua lampu LED akan padam dan tidak ada yang menyala.



Gambar 4.9 Uji Coba Pada Sensor *Fingerprint*

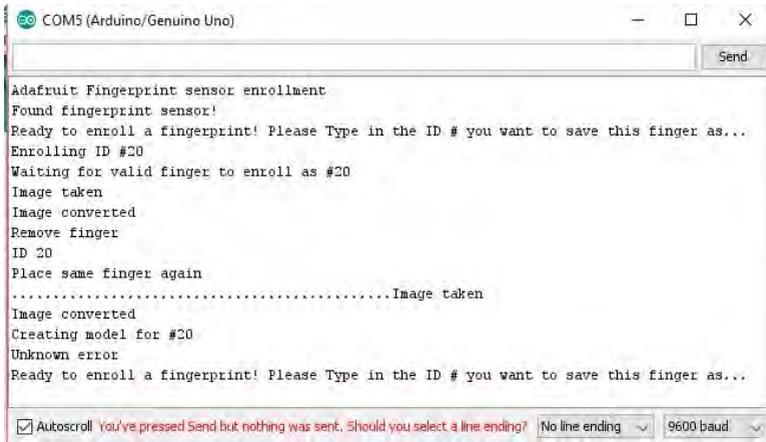
Pada Gambar 4.9 merupakan tampilan uji coba pada sensor *fingerprint* setelah diberi masukan satu kali. Lampu LED berwarna merah padam dan lampu LED berwarna hijau yang menyala. Lampu hijau merupakan indikator apakah data tersebut sudah tersimpan apa belum. Jika lampu LED hijau menyala maka data sudah tersimpan. Perintah selanjutnya dari arduino adalah memasukkan kembali data tersebut sekali lagi. Perintah selanjutnya yang ditampilkan pada software Arduino IDE akan ditampilkan pada Gambar 4.10 yaitu memasukkan kembali data sidik jari untuk kedua kalinya pada sensor *fingerprint*. Kemudian yang dilakukan adalah mengulang kembali menempelkan jari ke sensor sidik jari seperti yang telah dilakukan pada yang ditampilkan pada Gambar 4.8 sebelumnya.



```
fingertest
Found fingerprint sensor!
Type in the ID # you want to save this finger as...
Enrolling ID #0
Waiting for valid finger to enroll
Image taken
Image converted
Remove finger
Place same finger again
Image taken
Image converted
Unknown error
Type in the ID # you want to save this finger as...
```

Gambar 4.10 Tampilan Arduino Diberi Masukan Dua Kali

Pada Gambar 4.10 merupakan tampilan *software* Arduino IDE ketika diberi masukan dua kali, yaitu setelah menempelkan jari ke sensor sidik jari, diangkat kemudian ditempelkan kembali selama beberapa detik untuk menunggu data tersimpan apakah sama dengan masukan sebelumnya. Apabila cocok maka dapat disimpan dengan nomor ID yang diinginkan. Setelah data tersimpan, maka sudah dapat digunakan untuk membuka pintu pagar dengan menempelkan sidik jari. Pada Gambar 4.10 merupakan tampilan uji coba kondisi sensor pada *software* Arduino IDE. Pada saat menempelkan jari, *delay* selama tiga detik motor akan otomatis berputar menggeser pintu pagar. Dipasangkan juga *limitswitch* guna memberhentikan putaran motor secara otomatis. Dapat juga diatur untuk *delay* beberapa saat hingga akhirnya menutup kembali secara otomatis

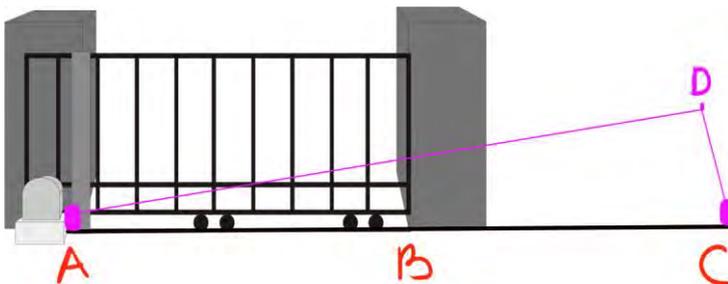


Gambar 4.11 Tampilan Uji Coba Kondisi Sensor

. Pada Gambar 4.11 merupakan tampilan *software* Arduino IDE pada saat ingin menyimpan data ulang dengan nomor ID yang berbeda. Kita dapat mengubah data *fingerprint* yang diinginkan tanpa menghapus data yang sebelumnya. Dengan cara menyimpan data dengan nomor ID yang berbeda. Apabila kita menyimpan dengan nomor ID yang sama, maka data yang lebih awal disimpan akan terhapus dan digantikan oleh data yang baru.

4.3 Hasil Pengambilan Data Pada *Fingerprint*

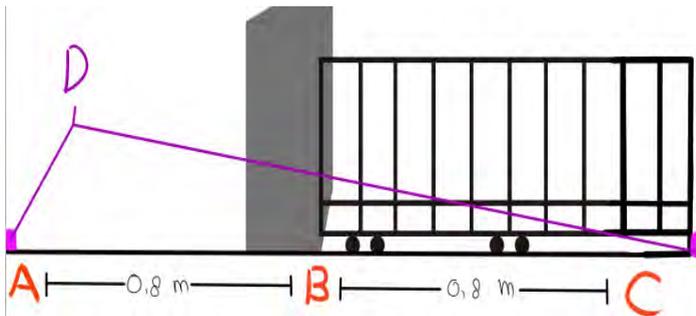
Berikut ini adalah Gambar 4.12 yaitu sketsa cara pengambilan data untuk Tabel 4.1.



Gambar 4.12 Sketsa Ujicoba

Pada Gambar 4.12 merupakan tampilan sketsa ujicoba pada arduino. Untuk tampilan dari arduino pada *fingerpint*. Pada gambar 4.12 yaitu posisi awal pagar yang tertutup. Huruf 'A' menunjukkan posisi awal pagar sebelum dibuka. Huruf 'B' dan 'C' merupakan posisi kedua dan ketiga. Sedangkan huruf 'D' merupakan *limit switch* yang akan bekerja apabila pagar telah menyentuh *limitswitch*. Pada kondisi *real*, huruf 'A' merupakan jarak pada 0 m. Huruf 'B' 0,8 m dan huruf 'C' kurang lebih 1,6 m. Berikut adalah arti dari huruf atau titik yang dipakai pada Gambar 4.12 dan 4.13:

- A. Posisi awal (pagar belum bergeser).
- B. Titik tengah.
- C. Titik akhir (pagar terbuka sepenuhnya)
- D. Titik dimana *limitswitch* dipasang.



Gambar 4.13 Sketsa uji Coba

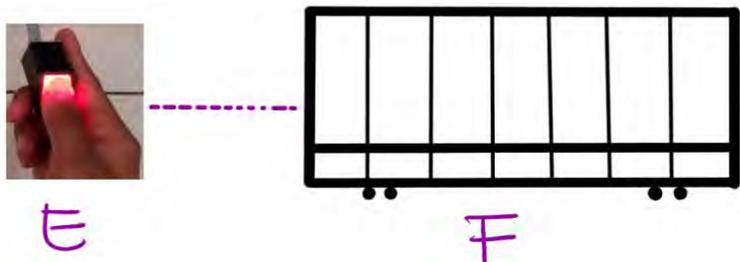
Pada Gambar 4.13 merupakan tampilan sketsa ujicoba pada arduino ketika sudah bergeser untuk membuka pagar. Pada gambar 4.13 ditunjukkan bahwa pagar masih berada pada titik 'A' yang berarti jarak yang ditempuh oleh pagar adalah 0 m. Jarak antara titik 'A' dan 'B' adalah 0,8 m. Dan jarak antara titik 'B' dan 'C' adalah 0,8 m. Sedangkan pada Gambar 4.13 ditunjukkan bahwa pagar telah bergeser sejauh 0,8 m sampai menyentuh *limitswitch* di titik 'C' kemudian berhenti sejenak selama 5 detik. Lalu akan otomatis kembali ke kondisi awal seperti pada Gambar 4.12.



Gambar 4.14 Tampilan Uji Coba Pada *Fingerprint*

Pada Gambar 4.14 merupakan tampilan uji coba pada *fingerprint*. yaitu saat memasukkan data berupa sidik jari terhadap sensor *fingerprint*. hasilnya adalah setelah dimasukkan data sidik jari tersebut, ada *delay* selama kurang lebih 3 detik sampai pagar dapat terbuka secara otomatis kemudian berhenti sejenak selama 5 detik setelah menyentuh *limitswitch*. Lalu kembali bergeser menutup secara otomatis.

Berikut ini adalah Tabel 4.1 yaitu Tabel Percobaan pada *fingerprint* dengan cara pengambilan datanya adalah dengan memberi tegangan terlebih dahulu. Kemudian memastikan bahwa semua rangkaian sudah tersambung dan lampu indikator LED menyala. Kemudian pada percobaan pertama dengan memasukkan data sidik jari dengan cara menempelkan jari yang datanya telah tersimpan di sensor *fingerprint* selama beberapa detik agar dapat terbaca oleh sensor *fingerprint* kemudian disalurkan datanya pada arduino yang akan menjalankan motor dan pagar akan bergerak. Ketika pagar bergeser untuk membuka pagar sampai ke ujung dan menyentuh *limitswitch*, pagar akan berhenti sejenak selama 5 detik dan bergeser menutup kembali secara otomatis kemudian berhenti.



Gambar 4.15 Sketsa Uji Coba

Pada Gambar 4.15 merupakan sketsa uji coba untuk data pada percobaan pada Tabel 4.1. Yaitu percobaan pada *fingerprint*. ‘E’ adalah *fingerprint* dan ‘F’ adalah pagar otomatis.

Tabel 4.1 Percobaan Pada *Fingerprint*

No	Percobaan	Bergeser	Tidak Bergeser	Limit Switch Aktif	Limit Switch Tidak Aktif
1	Percobaan 1	V	-	V	-
2	Percobaan 2	-	V	-	V
3	Percobaan 3	V	-	-	V
4	Percobaan 4	-	V	-	V
5	Percobaan 5	-	V	-	V
6	Percobaan 6	V	-	V	-
7	Percobaan 7	-	V	-	V
8	Percobaan 8	V	-	-	V
9	Percobaan 9	V	-	V	-

Berikut ini adalah pengertian pada Tabel 4.1

- ‘Bergeser’ yaitu pagar bergeser dari titik ‘A’ ke ‘B’ atau sebaliknya. Yaitu dari titik ‘B’ ke ‘A’. Pagar akan bergeser apabila motor berputar. Seperti pada Gambar 4.12 dan Gambar 4.13.
- Tanda ‘V’ merupakan ‘ya’. Permisalan di kolom ‘bergeser’ diberi tanda ‘V’, maka artinya adalah pagar

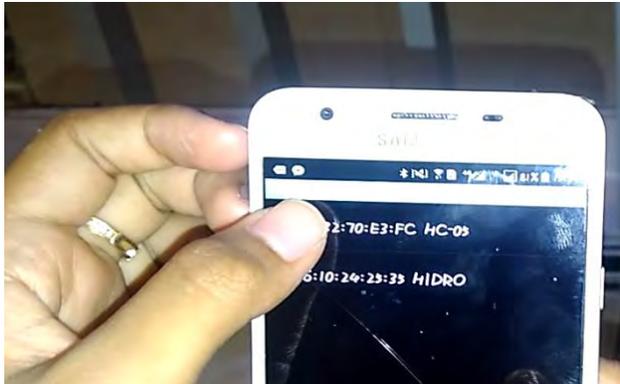
bergeser. Di kolom ‘*Limit Switch* Aktif’ diberi tanda ‘V’ maka artinya ya, *limitswitch* aktif.

- Tanda ‘-’ merupakan ‘tidak’. Permisalan di kolom ‘Bergeser’ diberi tanda ‘-’ maka artinya adalah tidak, pagar tidak bergeser.
- ‘*Limit Switch* Aktif’ merupakan kolom yang menunjukkan apakah *limitswitch* aktif, yaitu saat pagar menyentuh *limitswitch*, *limitswitch* akan aktif dan memerintahkan motor untuk berhenti berputar.
- ‘*Limit Switch* Tidak Aktif’ merupakan kolom yang menunjukkan apakah *limitswitch* tidak aktif jika diberi tanda ‘V’.

Pada Tabel 4.1 diketahui bahwa tanda ‘V’ berarti ya, dan tanda ‘-’ berarti tidak. Pengambilan data percobaan pertama sukses, motor berputar menggerakkan pagar sampai ke ujung dan kemudian berhenti sejenak selama lima detik dan otomatis kembali menutup pagar. Pada percobaan kedua ada kesalahan pada data *fingerprnt* sehingga pagar motor tidak berputar dan pagar tidak bergeser. Pada percobaan ketiga diketahui motor berputar dan menggerakkan pagar sampai ke ujung namun limit switch tidak berfungsi sehingga pagar bergeser menabrak *limitswitch* dan merusak lintasan dan harus mematikan dengan manual menggunakan tombol reset. Pada percobaan keempat dan kelima diketahui data yang diterima leh fingerprint tidak sesuai sehingga motor tidak berputar dan *limitswitch* tidak berfungsi. Pada percobaan keenam diketahui bahwa motor berputar dan menggerakkan pagar dan *limitswitch* berfungsi sehingga pagar dapat kembali menutup setelah delay 5 detik. Pada percobaan ketujuh ada kesalahan pada *fingerprnt* sehingga motor tidak berputar dan *limitswitch* tidak aktif. Pada percobaan ke delapan motor berputar dan menggerakkan pagar namun *limitswitch* tidak berfungsi. Sedangkan pada percobaan ke sembilan motor tidak bergerak sehingga pagar tidak bergerak dan *limitswitch* tidak berfungsi.

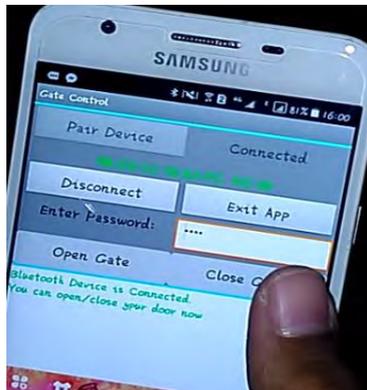
4.4 Hasil Pengambilan Data Pada Aplikasi Smartphone

Berikut ini adalah Gambar 4.15 yang menampilkan pemilihan *bluetooth* manakah yang akan disambungkan dengan *bluetooth* pada *smartphone*.



Gambar 4.16 Uji Coba Pada *Smartphone*

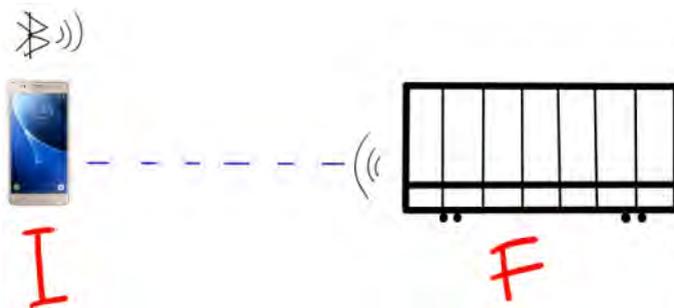
Pada Gambar 4.15 merupakan tampilan uji coba pada *smartphone* saat memilih *bluetooth* yang akan disambungkan dengan *bluetooth* pada *smartphone*. Perangkat *bluetooth* yang harus dipilih adalah HC-05.



Gambar 4.17 Uji Coba Pada *Smartphone*

Pada Gambar 4.16 merupakan tampilan uji coba pada *smartphone*. Setelah tahap menyambungkan *bluetooth* pada *smartphone* dengan *bluetooth* pada arduino, akan tersambung. Setelah tersambung maka sudah dapat memasukkan *password*. Setelah memasukkan *password* dengan benar maka dapat menyentuh pilihan 'Open gate' untuk membuka pagar.

Berikut ini adalah Tabel 4.2 yaitu Tabel Percobaan pada *Smartphone* dengan cara pengambilan datanya adalah dengan menyambungkan *bluetooth* terlebih dahulu. Pastikan Alat sudah terisi dengan tegangan yang dibutuhkan. Kemudian memastikan bahwa semua rangkaian sudah tersambung dan lampu indikator LED menyala. Lalu pada percobaan pertama dengan menyambungkan *bluetooth* pada *smartphone* dengan *bluetooth* pada Arduino. Pilih HC-05 yang tertera pada pilihan untuk sambungan *bluetooth*. Setelah tersambung, akan ada notifikasi bahwa *bluetooth* telah tersambung. Setelah itu, masukkan *password* yang sesuai dengan yang telah diatur. Kemudian dapat memilih pilihan 'Open gate' untuk membuka pagar dan 'Close Gate' untuk menutup pagar. Untuk membuka pagar menggunakan *smartphone*, dapat diatur kapan menutup atau membukanya. Setelah memilih 'Open gate', pagar akan terbuka secara otomatis dan ketika menyentuh *limitswitch* akan berhenti total. Tidak kembali secara otomatis seperti cara membuka dengan *fingerprint*. Maka harus memilih pilihan 'Close Gate' untuk menutup kembali pintu pagar.



Gambar 4.18 Uji Coba Pada *Smartphone*

- F. Pagar Otomatis
- I. *Smartphone*

Pada Gambar 4.18 merupakan sketsa uji coba pada *smartphone*. Yaitu menyambungkan *bluetooth* pada *smartphone* dengan *bluetooth* pada arduino. Apakah tersambung atau tidak, setelah tersambung maka dapat memasukkan *password*, apabila *password* yang dimasukkan benar maka pagar dapat membuka atau menutup.

Tabel 4.2 Data Pada Aplikasi *Smartphone*

No	Percobaan	Menyambungkan <i>Bluetooth</i>	<i>Password</i>	Membuka	Menutup
1	Percobaan 1	-	-	-	-
2	Percobaan 2	V	V	V	V
3	Percobaan 3	V	V	V	-
4	Percobaan 4	V	-	-	-
5	Percobaan 5	-	-	-	-
6	Percobaan 6	V	V	V	-
7	Percobaan 7	V	V	V	V
8	Percobaan 8	V	V	-	V
9	Percobaan 9	V	V	V	V
10	Percobaan 10	V	V	V	V

Berikut ini adalah pengertian pada Tabel 4.2

- ‘Menyambungkan *Bluetooth*’ merupakan kolom untuk mengetahui apakah *bluetooth* aktif atau tidak. Apabila diberi tanda ‘V’ maka *bluetooth* aktif. Dan jika tidak maka diberi tanda ‘-’
- Tanda ‘V’ merupakan ‘ya’. Permisalan di kolom ‘Menyambungkan *bluetooth*’ diberi tanda ‘V’, maka artinya adalah *bluetooth* aktif dan tersambung. Di kolom ‘*Password*’ diberi tanda ‘V’ maka artinya ya, *password* yang dimasukkan benar dan pagar dapat bergeser.
- Tanda ‘-’ merupakan ‘tidak’. Permisalan di kolom ‘membuka’ diberi tanda ‘-’ maka artinya adalah tidak, pagar tidak bergeser.
- ‘*Password*’ merupakan kolom yang menunjukkan apakah *password* yang dimasukkan benar atau salah. Apabila diberi tanda ‘V’ maka *password* benar, dan ‘-’ maka *password* tidak benar.
- ‘Membuka’ merupakan kolom yang menunjukkan apakah pagar dapat bergeser membuka atau tidak. Bila diberi

tanda ‘V’ maka pagar bergeser dan ‘-‘ tidak bergeser. ‘Membuka’ berarti bergeser dari titik ‘A’ ke titik ‘B’. Seperti pada Gambar 4.12 dan Gambar 4.13.

- ‘Menutup’ merupakan kolom yang menunjukkan apakah pagar dapat bergeser menutup atau tidak. Bila diberi tanda ‘V’ maka pagar bergeser dan ‘-‘ tidak bergeser. ‘Menutup’ berarti bergeser dari titik ‘B’ ke titik ‘A’ seperti pada Gambar 4.12 dan Gambar 4.13.

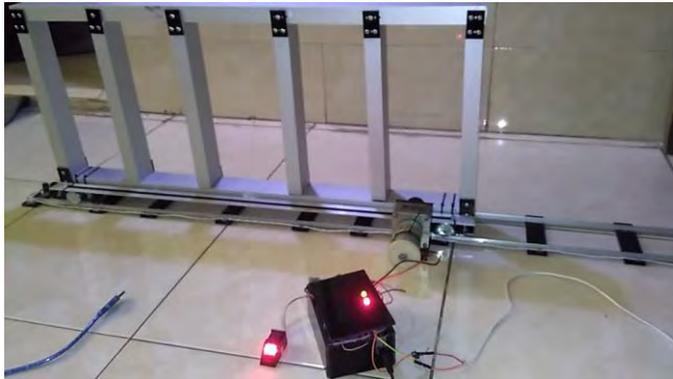
Pada Tabel 4.2 diketahui bahwa tanda ‘V’ berarti ya, dan tanda ‘-‘ berarti tidak. Tabel 4.2 menampilkan bahwa apabila *bluetooth* pada *smartphone* dan *bluetooth* pada Arduino telah tersambung maka selanjutnya akan ada perintah untuk memasukkan *password*, setelah memasukkan *password* dengan benar kita langsung bisa memerintahkan sistem untuk membuka pagar. Data akan diterima oleh *bluetooth* dan memerintahkan Arduino untuk menjalankan motor dan pagar akan bergeser. Pada percobaan pertama, diketahui bahwa penyambungan *bluetooth* gagal sehingga tidak bisa memasukkan *password* maupun membuka atau menutup pagar. Pada percobaan kedua, *bluetooth* dapat tersambung kemudian saat memasukkan *password* sudah tepat sehingga dapat membuka atau menutup pagar. Pada percobaan ketiga diketahui *bluetooth* sudah tersambung dan *password* yang dimasukkan benar, sehingga pagar dapat membuka, namun saat menutup terdapat error dan tidak dapat menutup. Pada percobaan keempat *bluetooth* dapat tersambung namun *password* yang dimasukkan tidak benar, maka pagar tidak bisa membuka atau menutup. Pada percobaan kelima, terdapat kesalahan pada *bluetooth* sehingga tidak dapat tersambung dan pagar tidak dapat bergeser membuka atau menutup. Pada percobaan keenam telah sukses menyambungkan *bluetooth* dan *password* yang dimasukkan benar sehingga pagar dapat bergeser membuka atau menutup. Pada percobaan ke delapan *bluetooth* tersambung dan *password* yang dimasukkan benar, namun pagar tidak bisa bergeser membuka dan hanya dapat menutup saja. Pada percobaan ke sembilan dan ke sepuluh diketahui bahwa sukses menyambungkan *bluetooth* dan memasukkan *password* dengan benar sehingga pagar dapat bergeser membuka atau menutup.

Perbedaan antara sistem pada *fingerprint* dengan sistem pada *Bluetooth* adalah, jika menggunakan aplikasi *smartphone*, kita bisa mengatur kapan kita akan membuka maupun menutup dan kita bisa

memerintah sistem. Karena pada saat pagar menyentuh *limitswitch*, motor akan otomatis berhenti dan menunggu perintah selanjutnya dari aplikasi *smartphone*.

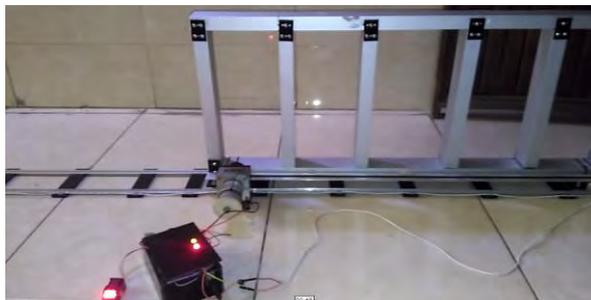
4.5 Data Percobaan Jarak Aplikasi *Smartphone*

Berikut ini adalah Gambar 4.19 yaitu tampilan saat *bluetooth* sudah tersambung namun belum diberi masukan data dari *smartphone*.



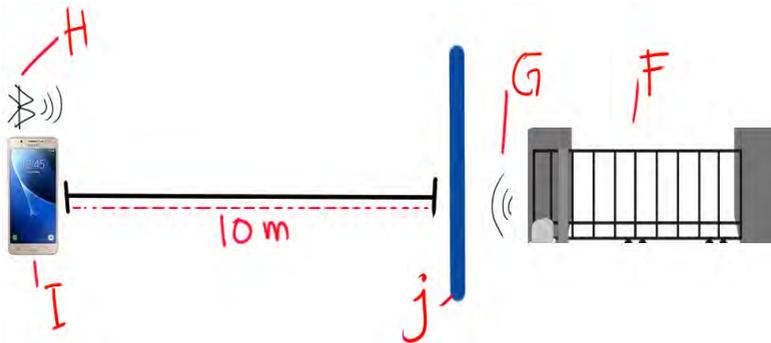
Gambar 4.19 Tampilan Uji Coba pada *Smartphone*

Pada Gambar 4.19 merupakan tampilan uji coba pada *smartphone*. Ketika belum diberi masukan data, lampu LED yang menyala adalah yang berwarna merah, dan pagar berada di sisi kiri. Ketika pagar sudah bergeser, maka kedua lampu LED akan menyala dan berkedip secara bersamaan.



Gambar 4.20 Tampilan Uji Coba pada *Smartphone*

Pada Gambar 4.20 merupakan tampilan uji coba pada *smartphone* ketika sudah diperintahkan untuk 'Open gate' sehingga mencapai ujung lintasan dan menyentuh *limitswitch* untuk berhenti. Kemudian agar dapat bergeser menutup kembali maka harus memilih pilihan 'Close Gate' pada aplikasi di *smartphone*.

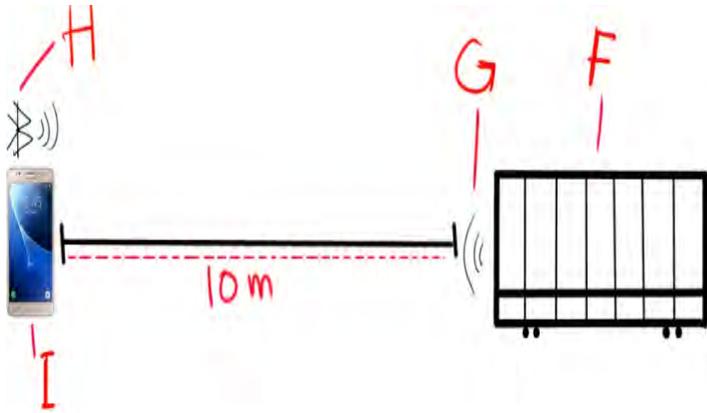


Gambar 4.21 Sketsa Uji Coba Jarak Pada Aplikasi *Smartphone*

Pada Gambar 4.21 merupakan sketsa uji coba jarak pada aplikasi *smartphone*. Berikut adalah pengertian angka yang tertera pada sketsa di Gambar 4.21

- F. Pagar Otomatis
- G. *Bluetooth* pada *Arduino*
- H. *Bluetooth* pada *smartphone*
- I. *Smartphone*
- J. Penghalang berupa tembok atau dinding

Pada Gambar 4.21 menampilkan sketsa uji coba jarak pada aplikasi di *smartphone* yang akan digunakan untuk tabel 4.3. Huruf I merupakan *smartphone* yang artinya *bluetooth* yang digunakan pada *smartphone* akan disambungkan dengan *bluetooth* yang ada pada *arduino* yang tersambung dengan pagar. Huruf E yaitu dinding penghalang. Uji coba ini ditujukan untuk mengetahui seberapa panjang jarak yang dapat ditempuh untuk penyambungan antar *bluetooth* dimana bila ada penghalang berupa dinding.



Gambar 4.22 Sketsa Uji Coba Jarak Pada Aplikasi *Smartphone*

Pada Gambar 4.22 merupakan sketsa uji coba jarak pada aplikasi *smartphone*. Berikut adalah pengertian huruf yang tertera pada sketsa di Gambar 4.22.

- F. Pagar Otomatis
- G. *Bluetooth* pada Arduino
- H. *Bluetooth* pada *smartphone*
- I. *Smartphone*

Pada Gambar 4.22 menampilkan sketsa uji coba jarak pada aplikasi *smartphone* yang akan digunakan untuk Tabel 4.3. Berbeda dengan sketsa pada Gambar 4.21. Pada Gambar 4.22 tidak memiliki penghalang ditengahnya sehingga *bluetooth* pada *smartphone* dapat langsung terhubung dengan *bluetooth* yang ada pada arduino tanpa halangan.

Berikut ini adalah Tabel 4.3 yaitu Tabel Percobaan pada *Smartphone* dengan cara pengambilan datanya adalah dengan menyambungkan *bluetooth* terlebih dahulu. Pastikan Alat sudah terisi dengan tegangan yang dibutuhkan. Kemudian memastikan bahwa semua rangkaian sudah tersambung dan lampu indikator LED menyala. Lalu pada percobaan pertama dengan menyambungkan *bluetooth* pada *smartphone* dengan *bluetooth* pada arduino. Pilih HC-05 yang tertera pada pilihan untuk sambungan *bluetooth*. Setelah tersambung, akan ada notifikasi bahwa *bluetooth* telah tersambung. Setelah itu, masukkan

password yang sesuai dengan yang telah diatur. Kemudian dapat memilih pilihan ‘*Open gate*’ untuk membuka pagar dan ‘*Close Gate*’ untuk menutup pagar. Untuk membuka pagar menggunakan *smartphone*, dapat diatur kapan menutup atau membukanya. Setelah memilih ‘*Open gate*’, pagar akan terbuka secara otomatis dan ketika menyentuh *limitswitch* akan berhenti total. Tidak kembali secara otomatis seperti cara membuka dengan *fingerpint*. Maka harus memilih pilihan ‘*Close Gate*’ untuk menutup kembali pintu pagar.

. Percobaan ini bertujuan untuk mengukur jarak yang dapat digunakan oleh *bluetooth* pada *smartphone* dengan *bluetooth* pada arduino apakah tersambung atau tidak dapat tersambung. Dengan menggunakan penghalang berupa tembok maupun tanpa penghalang.

Tabel 4.3 Data Percobaan Aplikasi *Smartphone*

No	Jarak (Meter)	Bergeser	Tanpa Penghalang	Dengan Penghalang
1	1	V	V	V
2	2	V	V	V
3	3	V	V	V
4	4	V	V	V
5	5	V	V	V
6	6	V	V	-
7	7	V	V	-
8	8	V	V	-
9	9	-	-	-
10	10	-	-	-

Pada Tabel 4.3 merupakan data percobaan pada aplikasi *smartphone*. Berikut adalah pengertiannya:

- ‘Jarak’ merupakan kolom untuk mengetahui seberapa jauh jarak agar *bluetooth* pada *smartphone* tersambung dengan *bluetooth* pada arduino.
- ‘Bergeser’ merupakan kolom untuk mengetahui apakah motor dapat berputar dan menggerakkan pagar sehingga pagar bergeser dari titik ‘A’ ke ‘B’ apabila membuka dan sebaliknya apabila menutup yaitu ‘B’ ke ‘A’ seperti pada Gambar 4.12 dan Gambar 4.13.
- ‘Tanpa Penghalang’ merupakan kolom untuk mengetahui apakah pagar dapat bergeser dengan jarak yang sudah ditentukan tanpa penghalang berupa dinding atau tembok antara *bluetooth* pada *smartphone* dengan *bluetooth* pada arduino. Pagar bergeser dari titik ‘A’ ke ‘B’ apabila membuka dan sebaliknya apabila menutup yaitu ‘B’ ke ‘A’ seperti pada Gambar 4.12 dan Gambar 4.13.
- ‘Dengan Penghalang’ merupakan kolom untuk mengetahui apakah pagar dapat bergeser dengan jarak yang sudah ditentukan dengan penghalang berupa dinding atau tembok antara *bluetooth* pada *smartphone* dengan *bluetooth* pada arduino. Pagar bergeser dari titik ‘A’ ke ‘B’ apabila membuka dan sebaliknya apabila menutup yaitu ‘B’ ke ‘A’ seperti pada Gambar 4.12 dan Gambar 4.13.

Pada Tabel 4.3 diketahui bahwa tanda ‘V’ berarti ya dan tanda ‘-’ berarti tidak. Tabel 4.3 menampilkan percobaan pada aplikasi *smartphone* apakah *bluetooth* dapat berfungsi dan menyalurkan data ke Arduino sehingga motor dapat bergerak. Pada percobaan pertama diketahui pagar dapat bergeser pada jarak 1 meter dengan penghalang maupun tanpa penghalang. Pada percobaan kedua dapat tersambung *bluetooth* pada *smartphone* dengan *bluetooth* pada arduino sehingga dapat bergeser dengan penghalang dan tanpa penghalang. Begitu juga untuk percobaan ketiga sampai kelima, *bluetooth* dapat tersambung dengan baik dan pagar bergeser ketika tanpa penghalang maupun dengan penghalang. Untuk percobaan kelima sampai ketujuh, *bluetooth* dapat aktif dan tersambung saat tanpa penghalang namun tidak dapat tersambung saat tidak dengan penghalang.

Pada percobaan ini *bluetooth* diketahui dapat digunakan tanpa penghalang sampai dengan kurang lebih delapan meter jaraknya. Sedangkan untuk dengan penghalang berupa tembok hanya dapat

digunakan sejauh lima meter saja. Kesimpulan yang dapat diambil pada percobaan ini diketahui bahwa *bluetooth* dapat digunakan dan aktif saat jarak sampai dengan 8 meter untuk tanpa penghalang. Dan *bluetooth* dapat digunakan sampai dengan 5 meter dengan penghalang berupa tembok.

BAB V

PENUTUP

4.1 Hasil Pengambilan Data Pada Aplikasi *Smartphone*

Dari hasil pengujian sistem kontrol hasil desain dengan simulasi maupun implementasi, dapat diambil kesimpulan bahwa *remote control* pembuka pintu pagar otomatis menggunakan program Android memberikan respon positif terhadap pintu pagar tersebut. Cara kerja pertamanya adalah motor dan Arduino sudah harus terpasang dengan *power supply*, yaitu stop kontak. Pastikan Arduino dan *Bluetooth* sudah tersambung dengan benar dan bekerja dengan baik. Kemudian nyalakan *smartphone* dan buka *software* yang ada pada *smartphone*, hidupkan *Bluetooth* pada *smartphone* dan sambungkan dengan *Bluetooth* yang tersambung dengan Arduino. Setelah terhubung akan muncul layar masukkan *password*, setelah memasukkan *password* akan menampilkan fitur lanjutan yaitu membuka atau menutup pagar. Jika menyentuh pilihan '*open gate*' maka pintu pagar akan terbuka dan jika menekan '*close gate*' maka pintu pagar akan menutup. Yang kedua adalah sensor *fingerprint* harus sudah tersambung dengan Arduino dan kabel usb. Ketika kita memasukkan data sidik jari, *fingerprint* akan membaca datanya kemudian disalurkan ke Arduino, dari Arduino akan memerintah motor untuk bergerak. Alat dapat bekerja dengan jarak kurang lebih 0,5 sampai 1 meter sesuai dengan uji coba yang telah dilakukan. Pagar akan membuka dengan *delay* kurang lebih 1 detik apabila menggunakan *Fingerprint* dan akan menutup otomatis setelah *delay* 5 detik. Sedangkan apabila menggunakan *Smartphone* akan ada *delay* kurang lebih 15 detik setelah menyentuh pilihan '*Open gate*' maupun '*Close Gate*'.

a. Saran

Saran untuk penelitian selanjutnya adalah diharapkan ada *software* yang langsung dari Android. Menggunakan Android yang mempunyai *fingerprint*, dan juga perlu ditambahkan beberapa variasi pengujian lain pada simulasi maupun implementasi guna menguji tingkat kekokohan kontroler hasil desain.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Austin, Bill. *Electric Motors and Drivers, Fourth Edition*. Oxford : Elsevier Ltd. 2014
- [2] Andrianto, Heri. *Pemrograman Mikrokontroler AVR ATmega16 Menggunakan Bahasa C (CodeVisionAVR)*. Bandung : Informatika. 2013
- [3] Feri, Djuandi. *Pengenalan Arduino*. E-book. www.tobuku.com. Diakses pada 25 Juni 2017
- [4] McLeod Jr, Raymond. *Sistem Informasi Manajemen, Edisi Ketujuh*. Jakarta : PT Prenhallindo. 2001
- [5] Safaat, Nazarudin. *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Jakarta: Informatika. 2011
- [6] Sugiri. *Elektronika Dasar Dan Peripheral Komputer*, Yogyakarta : Andi Publisher. 2014
- [7] Syahrul. *Pemrograman Mikrokontroller Avr*. Bandung: Informatika. 2014.
- [8] Syahwil, Muhammad. *Panduan Mudah Simulasi dan Praktek Mikrokontroler Arduino*. Yogyakarta : Andi Publisher. 2014.

-----Halaman ini sengaja dikosongkan-----

G. LAMPIRAN

A) Listing program pada Arduino

```
#include <Adafruit_Fingerprint.h>
#if ARDUINO >= 100
  #include <SoftwareSerial.h>
#else
  #include <NewSoftSerial.h>
#endif

#include <Streaming.h>

int getFingerprintIDez();
int pwm_ena=10;           //pin untuk mengatur kecepatan motor
int in1=8;                //direction motor
int in2=9;                //direction motor
int led_r=11;             //pin led merah
int led_g=12;             //pin led hijau
int flag=0;               //indeks untuk perulangan
int ls=6;                 //pin limit switch
int state=0;              //kondisi penekanan limit switch
int data1,data2;          //penyimpanan data serial dari Android
String dataIn;            //data yang diterima dari Android sebelum di
  parsing
String dt[10];            //jumlah maksimal data yang diterima dari
  Android
int k,i;                  //variable counter
boolean parsing=false;    //kondisi pengiriman serial

#if ARDUINO >= 100
  SoftwareSerial mySerial(2, 3);// tx, rx
#else
  NewSoftSerial mySerial(2, 3);
#endif

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
```

```

void parsingData() { //berisi program untuk pemisahan
data yang dikirim dari Android ke Arduino
  int j=0;
  //kirim data yang telah diterima sebelumnya
  //Serial.print("data masuk : ");
  //Serial.print(dataIn);
  //Serial.print("\n");

  //inisialisasi variabel, (reset isi variabel)
  dt[j]="";
  //proses parsing data
  for(k=1;k<dataIn.length();k++){
    //pengecekan tiap karakter dengan karakter (#) dan (,.)
    if ((dataIn[k] == '#') || (dataIn[k] == ',')){
      //increment variabel j, digunakan untuk merubah index array
penampung
      j++;
      dt[j]=""; //inisialisasi variabel array dt[j]
    }
    else{
      //proses tampung data saat pengecekan karakter selesai.
      dt[j] = dt[j] + dataIn[k];
    }
  }
  data1=dt[0].toInt(); //penyimpanan data pertama
  data2=dt[1].toInt(); //penyimpanan data kedua
  Serial.print(data1); //tampilkan data pertama pada serial monitor
  Serial.print(", ");
  Serial.println(data2); //tampilkan data kedua pada serial monitor
  //kirim data hasil parsing
  /*Serial.print("data 1 : ");
  Serial.print(dt[0].toInt());
  Serial.print("\n");
  Serial.print("data 2 : ");
  Serial.print(dt[1].toInt());
  Serial.print("\n");
  Serial.print("data 3 : ");
  Serial.print(dt[2].toInt());
  Serial.print("\n");
*/
}

```

```

    Serial.print("data 4 : ");
    Serial.print(dt[3].toInt());
    Serial.print("\n\n");*/
    delay(50);
}
void doorOpen() {           //fungsi untuk membuka pintu
    analogWrite(pwm_ena,255);
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
}
void doorClose() {        //fungsi untuk mnutup pintu
    analogWrite(pwm_ena,255);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
}
void doorStopped() {      //fungsi untuk memberhentikan pergerakan
    pntu
    analogWrite(pwm_ena,0);
    digitalWrite(in1,HIGH);
    digitalWrite(in2,HIGH);
}

void setup() {
    Serial.begin(9600);      //inisialisasi komunikasi serial
    //Serial.println("fingertest");
    pinMode(pwm_ena,OUTPUT); //inisialisasi pwm motor
    pinMode(in1,OUTPUT);    //inisialisasi direction motor
    pinMode(in2,OUTPUT);    //inisialisasi direction motor
    pinMode(led_r,OUTPUT);  //inisialisasi led merah
    pinMode(led_g,OUTPUT);  //inisialisasi led hijau
    pinMode(ls,INPUT_PULLUP); //inisialisasi limit switch
    finger.begin(57600);    //inisialisasi finger print

    if (finger.verifyPassword()) { //pengecekan kondisi fingerprint
        //Serial.println("Found fingerprint sensor!");
    } else { //fingerprint not ready
        //Serial.println("Did not find fingerprint sensor :(");
        while (1);
    }
}

```

```

//Serial.println("Waiting for valid finger...");
}

void loop(){
  start:
  if(Serial.available(>0) {          //nunggu data masuk dari Android
    char inChar = (char)Serial.read(); //simpan data yang masuk
    dataIn += inChar;
    if (inChar == '\n') {          //jika karakter yang terkirim merupakan
karakter enter
      parsing = true;              //mulai parsing
    }
  }
  if(parsing){
    parsingData();                //panggil fungsi parsing data
    //servo1.write(dt[0].toInt());
    parsing=false;
    dataIn="";                     //clear data untuk sementara
  }
  if((data1==0) && (data2==1234)){ //password dari Android untuk
menutup pintu
    doorClose();                  //tutup pintu
    digitalWrite(led_r,HIGH);     //nyalakan lampu led merah
    delay(50);
    state = digitalRead(ls);      //baca pembacaan dari limit switch
    if (state == LOW){            //limit switch ke tekan
      delay(50);
      while(1){
        doorStopped();
        delay(100);
        doorOpen();
        delay(500);
        doorStopped();
        data1=2;
        data2=0;
        goto start;
      }
    }
  }
}
}

```

```

if((data1==1) && (data2==1234)){
  doorOpen();
  digitalWrite(led_g,HIGH);
  delay(50);
  state = digitalRead(ls);
  if (state == LOW){
    delay(50);
    while(1){
      doorStopped();
      delay(100);
      doorClose();
      delay(500);
      doorStopped();
      data1=2;
      data2=0;
      goto start;
    }
  }
}
}
}
//start:
/*
if(Serial.available()>0) {
  char inChar = (char)Serial.read();
  dataIn += inChar;
  if (inChar == '\n') {
    parsing = true;
  }
}
}
if(parsing){
  parsingData();
  //servo1.write(dt[0].toInt());
  parsing=false;
  dataIn="";
}
while((data1==0) && (data2==1234)){
  flag=2;
  delay(50);
  while(flag==2){
    doorClose();

```

```

    digitalWrite(led_r,HIGH);
    state = digitalRead(ls);
    if (state == LOW){
        flag=3;
        while(flag==3){
            doorStopped();
            delay(200);
            digitalWrite(led_r,LOW);
            digitalWrite(led_g,LOW);
            data1=2;
            flag=0;
            goto start;
        }
        //goto start;
    }
}
}
}
while((data1==1) && (data2==1234)){
    flag=4;
    while(flag==4){
        doorOpen();
        digitalWrite(led_g,HIGH);
        state = digitalRead(ls);
        if (state == LOW){
            flag=5;
            while(flag==5){
                doorStopped();
                delay(200);
                digitalWrite(led_g,LOW);
                digitalWrite(led_r,LOW);
                data1=2;
                flag=0;
                goto start;
            }
            //goto start;
        }
    }
}
}
}*/
digitalWrite(led_r,HIGH);

```

```

delay(500);
digitalWrite(led_r,LOW);
delay(500);
if(getFingerprintIDEz()==1){
  flag=1;
  digitalWrite(led_r,LOW);
  while(flag==1){
    digitalWrite(led_g,HIGH);
    delay(500);
    digitalWrite(led_g,LOW);
    delay(500);
    doorOpen();
    state = digitalRead(ls);
    if (state == LOW) {
      doorClose();
      delay(2000);
      doorStopped();
      delay(5000);
      while(1){
        digitalWrite(led_g,HIGH);
        delay(200);
        digitalWrite(led_g,LOW);
        delay(200);
        doorClose();
        state = digitalRead(ls);
        if (state == LOW){
          delay(50);
          while(1){
            doorStopped();
            delay(100);
            doorOpen();
            delay(500);
            doorStopped();
            delay(200);
            for(i=0;i<5;i++){
              digitalWrite(led_g,LOW);
              digitalWrite(led_r,HIGH);
              delay(500);
              digitalWrite(led_g,HIGH);
            }
          }
        }
      }
    }
  }
}

```



```

    //Serial.println("Image converted");
    break;
case FINGERPRINT_IMAGEMESS:
    //Serial.println("Image too messy");
    return p;
case FINGERPRINT_PACKETRECIEVEERR:
    //Serial.println("Communication error");
    return p;
case FINGERPRINT_FEATUREFAIL:
    //Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    //Serial.println("Could not find fingerprint features");
    return p;
default:
    //Serial.println("Unknown error");
    return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    //Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    //Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    //Serial.println("Did not find a match");
    return p;
} else {
    //Serial.println("Unknown error");
    return p;
}

// found a match!
//Serial.print("Found ID #"); Serial.print(finger.fingerID);
//Serial.print("          with          confIDence          of          ");
Serial.println(finger.confIDence);
}

```

```

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;

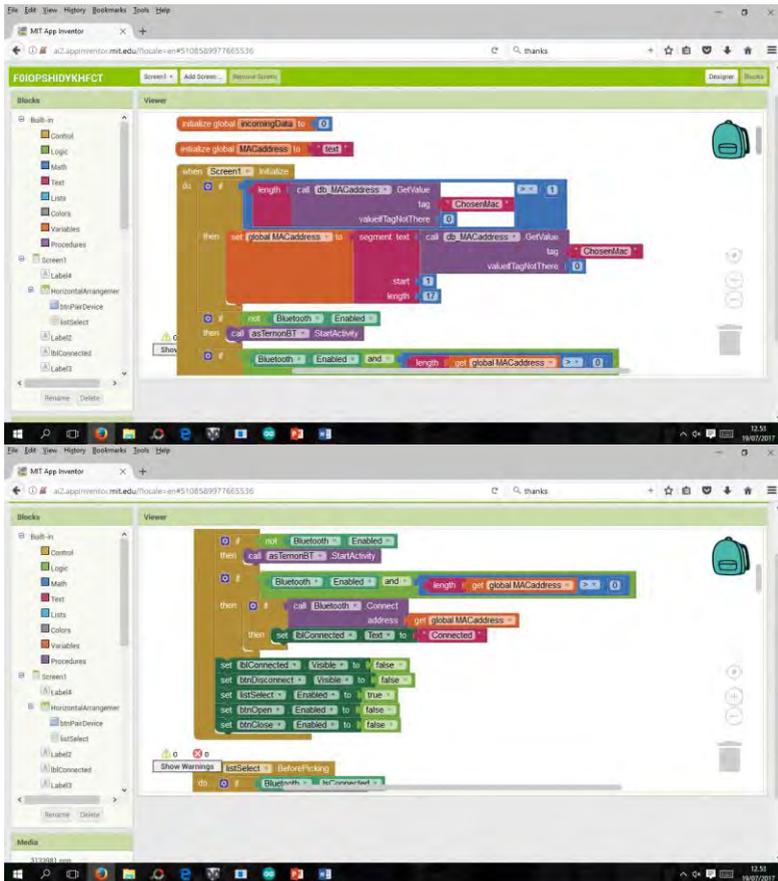
    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;

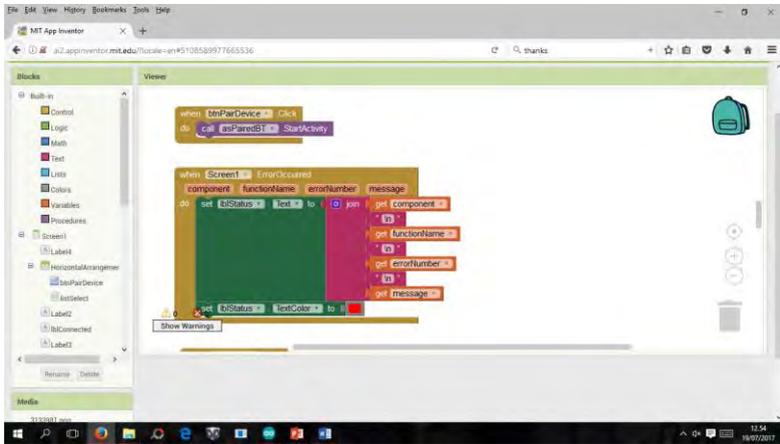
    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) return -1;

    // found a match!
    //Serial.print("Found ID #"); Serial.print(finger.fingerID);
    //Serial.print("      with      confidence      of      ");
    Serial.println(finger.confidence);
    return finger.fingerID;
}

```

b. Listing program pada *App Inventor*





V

