

**PROTOTIPE PAGAR OTOMATIS MENGGUNAKAN *RADIO*
FREQUENCY IDENTIFICATION (RFID) MFRC522**

**DENGAN AKSES E-KTP BERBASIS
ARDUINO UNO**



TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat kelulusan Diploma Tiga (D.III)

FATKU RACHMAN SALEH

NIM : 13160981

GUTOMO MULYA PUTRA

NIM : 13160852

Program Studi Teknologi Komputer

Fakultas Teknologi Informasi

Universitas Bina Sarana Informatika

Jakarta

2019

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Fatku Rachman Saleh
NIM : 13160981
Jenjang : Diploma Tiga (D3)
Program Studi : Teknologi Komputer
Fakultas : Teknologi Informasi
Perguruan Tinggi : Universitas Bina Sarana In

Dengan ini menyatakan bahwa Tugas Akhir yang telah saya buat dengan judul :**“Prototipe Pagar Otomatis Menggunakan Radio Frequency Identification (RFID) MFRC522 Dengan Akses E-KTP Berbasis Arduino Uno”**, adalah asli (orisinal) atau tidak plagiat (menjiplak) dan belum pernah diterbitkan/dipublikasikan dimanapun dan dalam bentuk apapun.

Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya tanpa ada paksaan dari pihak manapun juga. Apabila di kemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa tugas akhir yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari **Universitas Bina Sarana Informatika** dicabut/dibatalkan.

Dibuat di : Jakarta
Pada tanggal : 15 Juli 2019
Yang menyatakan



Fatku Rachman Saleh

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Gutomo Mulya Putra
NIM : 13160852
Jenjang : Diploma Tiga (D3)
Program Studi : Teknologi Komputer
Fakultas : Teknologi Informasi
Perguruan Tinggi : Universitas Bina Sarana Informatika

Dengan ini menyatakan bahwa Tugas Akhir yang telah saya buat dengan judul : "Prototipe Pagar Otomatis Menggunakan *Radio Frequency Identification (RFID)* MFRC522 Dengan Akses E-KTP Berbasis *Arduino Uno*", adalah asli (orisinil) atau tidak plagiat (menjiplak) dan belum pernah diterbitkan/dipublikasikan dimanapun dan dalam bentuk apapun.

Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya tanpa ada paksaan dari pihak manapun juga. Apabila di kemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa tugas akhir yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari Universitas Bina Sarana Informatika dicabut/dibatalkan.

Dibuat di : Jakarta
Pada tanggal : 15 Juli 2019
Yang menyatakan,



Gutomo Mulya Putra

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan di bawah ini, saya:

Nama : Fatku Rachman Saleh
NIM : 13160981
Jenjang : Diploma Tiga (D3)
Program Studi : Teknologi Komputer
Fakultas : Teknologi Informasi
Perguruan Tinggi : Universitas Bina Sarana Informatika

Dengan ini menyetujui untuk memberikan ijin kepada pihak **Universitas Bina Sarana Informatika**, Hak Bebas Royalti Non- Eksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah kami yang berjudul: "**Prototipe Pagar Otomatis Menggunakan Radio Frequency Identification(RFID) MFRC522 Dengan Akses E-KTP Berbasis Arduino Uno**", beserta perangkat yang diperlukan (apabila ada).

Dengan **Hak Bebas Royalti Non-Eksklusif** ini pihak **Universitas Bina Sarana Informatika** berhak menyimpan, mengalih-media atau mem-format-kan, mengelolanya dalam pangkalan data (*database*), mendistribusikannya dan menampilkan atau mempublikasikannya di *internet* atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari kami selama tetap mencantumkan nama kami sebagai penulis/pencipta karya ilmiah tersebut.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak **Universitas Bina Sarana Informatika**, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta
Pada tanggal: 15 Juli 2019
Yang menyatakan,



Fatku Rachman Saleh

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan di bawah ini, saya:

Nama : Gutomo Mulya Putra
NIM : 13160852
Jenjang : Diploma Tiga (D3)
Program Studi : Teknologi Komputer
Fakultas : Teknologi Informasi
Perguruan Tinggi : Universitas Bina Sarana Informatika

Dengan ini menyetujui untuk memberikan ijin kepada pihak **Universitas Bina Sarana Informatika**, Hak Bebas Royalti Non- Eksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah kami yang berjudul: "**Prototipe Pagar Otomatis Menggunakan Radio Frequency Identification(RFID) MFRC522 Dengan Akses E-KTP Berbasis Arduino Uno**", beserta perangkat yang diperlukan (apabila ada).

Dengan **Hak Bebas Royalti Non-Eksklusif** ini pihak **Universitas Bina Sarana Informatika** berhak menyimpan, mengalih-media atau mem-format-kan, mengelolanya dalam pangkalan data (*database*), mendistribusikannya dan menampilkan atau mempublikasikannya di *internet* atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari kami selama tetap mencantumkan nama kami sebagai penulis/pencipta karya ilmiah tersebut.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak **Universitas Bina Sarana Informatika**, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta
Pada tanggal: 15 Juli 2019
Yang menyatakan



PERSETUJUAN DAN PENGESAHAN TUGAS AKHIR

Tugas Akhir ini diajukan oleh:

Nama : Fatku Rachman Saleh
NIM : 13160981
Program Studi : Teknologi Komputer
Jenjang : Diploma Tiga
Judul Tugas Akhir : PROTOTYPE PAGAR OTOMATIS MENGGUNAKAN
RADIO FREQUENCY IDENTIFICATION (RFID)
MFRC522 DENGAN AKSES E-KTP BERBASIS
ARDUINO UNO

Telah dipertahankan pada periode 2019-1 dihadapan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh Ahli Madya Teknik (A.Md.T) pada Program Diploma Tiga Program Studi Teknologi Komputer di Universitas Bina Sarana Informatika.

Jakarta, 29 Juli 2019

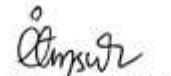
PEMBIMBING TUGAS AKHIR

Dosen Pembimbing : Aziz Setyawan Hidayat, M.Kom



DEWAN PENGUJI

Penguji I : Ita Dewi Sintawati, ST.,M.Kom



Penguji II : Giatika Chrisnawati, ST, M.Kom



PERSETUJUAN DAN PENGESAHAN TUGAS AKHIR

Tugas Akhir ini diajukan oleh:

Nama : Gutomo Mulya Putra
NIM : 13160852
Program Studi : Teknologi Komputer
Jenjang : Diploma Tiga
Judul Tugas Akhir : PROTOTYPE PAGAR OTOMATIS MENGGUNAKAN
RADIO FREQUENCY IDENTIFICATION (RFID)
MFRC522 DENGAN AKSES E-KTP BERBASIS
ARDUINO UNO

Telah dipertahankan pada periode 2019-1 dihadapan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh Ahli Madya Teknik (A.Md.T) pada Program Diploma Tiga Program Studi Teknologi Komputer di Universitas Bina Sarana Informatika.

Jakarta, 29 Juli 2019

PEMBIMBING TUGAS AKHIR

Dosen Pembimbing : Aziz Setyawan Hidayat, M.Kom



DEWAN PENGUJI

Penguji I : Ita Dewi Sintawati, ST.,M.Kom



Penguji II : Giatika Chrisnawati, ST, M.Kom





LEMBAR KONSULTASI TUGAS AKHIR

UNIVERSITAS BINA SARANA INFORMATIKA

NIM : 13160981
Nama Lengkap : Fatku Rachman Saleh
Dosen Pembimbing : Aziz Setyawan Hidayat, M.Kom
Judul Tugas Akhir : Prototipe Pagar Otomatis Menggunakan Radio Frequency Identification (RFID) MFRC522 Dengan Akses E-KTP Berbasis Arduino Uno.

NO	Tanggal Bimbingan	Pokok Bahasan	Paraf Dosen Pembimbing
1	04-04-2019	Bimbingan pertama membahas Bab 1	
2	25-04-2019	Revisi Bab 1 dan Acc Bab 1	
3	20-06-2019	Bimbingan Bab 2	
4	24-06-2019	Revisi Bab 2 dan Acc	
5	15-07-2019	Bimbingan Bab 3 dan Bab 4	
6	18-07-2019	Revisi Bab 3 dan Bab 4 dan acc	
7	22-07-2019	Uji Coba Alat	
8	24-07-2019	ACC Keseluruhan	

Catatan untuk Dosen Pembimbing.

Bimbingan Tugas Akhir

- Dimulai pada tanggal : 04 April 2019
- Diakhiri pada tanggal : 24 Juli 2019
- Jumlah pertemuan bimbingan : 8 Kali Pertemuan

Disetujui oleh,
Dosen Pembimbing

(Aziz Setyawan Hidayat, M.Kom)



LEMBAR KONSULTASI TUGAS AKHIR

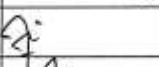
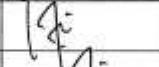
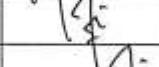
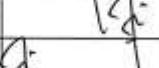
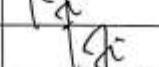
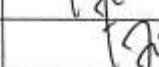
UNIVERSITAS BINA SARANA INFORMATIKA

NIM : 13160852

Nama Lengkap : Gutomo Mulya Putra

Dosen Pembimbing : Aziz Setyawan Hidayat, M.Kom

Judul Tugas Akhir : Prototipe Pagar Otomatis Menggunakan Radio Frequency Identification (RFID) MFRC522 Dengan Akses E-KTP Berbasis Arduino Uno.

NO	Tanggal Bimbingan	Pokok Bahasan	Paraf Dosen Pembimbing
1	04-04-2019	Bimbingan pertama membahas Bab 1	
2	25-04-2019	Revisi Bab 1 dan Acc Bab 1	
3	20-06-2019	Bimbingan Bab 2	
4	24-06-2019	Revisi Bab 2 dan Acc	
5	15-07-2019	Bimbingan Bab 3 dan Bab 4	
6	18-07-2019	Revisi Bab 3 dan Bab 4 dan acc	
7	22-07-2019	Uji Coba Alat	
8	24-07-2019	ACC Keseluruhan	

Catatan untuk Dosen Pembimbing.

Bimbingan Tugas Akhir

- Dimulai pada tanggal : 04 April 2019
- Diakhiri pada tanggal : 24 Juli 2019
- Jumlah pertemuan bimbingan : 8 Kali Pertemuan

Disetujui oleh,
Dosen Pembimbing

(Aziz Setyawan Hidayat, M.Kom)

KATA PENGANTAR

Alhamdulillah, Dengan mengucapkan puji syukur kehadirat Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya, sehingga pada akhirnya penulis dapat menyelesaikan tugas ini dengan baik. Tugas Akhir pada Program Diploma Tiga (D3) ini penulis sajikan dalam bentuk buku yang sederhana. Adapun judul Tugas Akhir, yang penulis ambil sebagai berikut, “**Prototipe Pagar Otomatis menggunakan Radio Frequency Identification (RFID) MFRC522 Dengan Akses E-KTP Berbasis Arduino Uno.**”.

Tujuan penulisan Tugas Akhir pada Program Diploma Tiga (D3) ini dibuat sebagai salah satu syarat kelulusan Program Diploma Universitas Bina Sarana Informatika. Sebagai bahan penulisan diambil berdasarkan hasil penelitian (eksperimen), observasi dan beberapa sumber literatur yang mendukung penulisan ini. Penulis menyadari bahwa tanpa bimbingan dan dorongan dari semua pihak, maka penulisan Tugas Akhir ini tidak akan berjalan lancar. Oleh karena itu pada kesempatan ini, ijinkanlah penulis menyampaikan ucapan terima kasih kepada:

1. Rektor Universitas Bina Sarana Informatika.
2. Dekan Fakultas Teknologi Informasi Universitas Bina Sarana Informatika.
3. Ketua Program Studi Teknologi Komputer Universitas Bina Sarana Informatika.
4. Bapak Aziz Setyawan Hidayat, M.Kom selaku Dosen Pembimbing Tugas Akhir.
5. Staff / karyawan / dosen di lingkungan Universitas Bina Sarana Informatika.
6. Orang tua tercinta yang telah memberikan dukungan moral maupun spiritual.
7. Rekan-rekan mahasiswa kelas TK-13.6A.01.

Serta semua pihak yang terlalu banyak untuk disebut satu persatu sehingga terwujudnya penulisan ini. Penulis menyadari bahwa penulisan Tugas Akhir ini masih jauh sekali dari sempurna, untuk itu penulis mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan di masa yang akan datang.

Akhir kata semoga Tugas Akhir ini dapat berguna bagi penulis khususnya dan bagi para pembaca yang berminat pada umumnya.

Jakarta, 08 Juli 2019

Penulis

ABSTRAK

**Fatku Rachman Saleh (13160981), dan Gutomo Mulya Putra (13160852),
Prototipe Pagar Otomatis Menggunakan RFID (*Radio Frequency Identification*)
MFRC522 Dengan Akses E-KTP Berbasis Arduino Uno.**

Perkembangan teknologi yang sangat cepat semakin terasa dalam kehidupan saat ini. Hal ini ditandai dengan perkembangan pada bidang komunikasi, kehadiran perangkat elektronika pada saat ini sangat besar manfaatnya bagi umat manusia. Seiring dengan perkembangan teknologi kebutuhan untuk mengendalikan berbagai alat elektronika dan keamanan tidak hanya dilakukan secara manual yang mengharuskan kita berada di depan alat elektronik tersebut dan menekan tombol on/off untuk mengaktifkan tetapi bisa juga dilakukan secara otomatis. Untuk itulah penulis mencoba membuat Tugas Akhir mengenai sistem pagar rumah yang bergerak secara otomatis, yang menggunakan rfid sebagai akses buka atau tutupnya. Jadi dengan adanya teknologi ini bisa mempermudah aktivitas kita sehari hari. Pagar rumah otomatis ini dikendalikan oleh mikrokontroller Arduino Uno, dan sebagai penggerak otomatisnya kami menggunakan Motor Stepper yang menggerakkan pagar agar membuka atau menutup secara otomatis. Komponen pendukung seperti sensor HC-SR04 yang sebagai detektor ketika ada objek yang masuk atau berhenti maka sensor ini bekerja untuk memberi intruksi ke motor stepper agar tidak menutup pagar. Adapun komponen penunjang lainnya ada buzzer yang berfungsi sebagai suara ketika proses berjalan, demikian LED sebagai notifikasi ketika proses berjalan juga. Dan dilengkapi dengan monitor LCD sebagai penampil teks ketika akses berjalan.

Kata Kunci: Pagar Otomatis, Arduino Uno, rfid, sensor ultrasonik

ABSTRACT

***Fatku Rachman Saleh (13160981), and Gutomo Mulya Putra (13160852),
Automatic Fence Prototype Using RFID (Radio Frequency Identification)
MFRC522 With E-KTP Access Based on Arduino Uno.***

The rapid development of technology is increasingly felt in today's life. This is marked by developments in the field of communication, the presence of electronic devices at this time is of great benefit to mankind. Along with the development of technology the need to control various electronic devices and security is not only done manually which requires us to be in front of the electronic device and press the on / off button to activate but it can also be done automatically. For this reason, the author tries to make the Final Project concerning a house fence system that moves automatically, which uses rfid as access to open or close it. So the existence of this technology can facilitate our daily activities. This automatic home fence is controlled by an Arduino Uno microcontroller, and as an automatic driver we use a Stepper Motor that moves the fence to open or close automatically. Supporting components such as the HC-SR04 sensor which as a detector when an object enters or stops, this sensor works to give instructions to the stepper motor so that it does not close the fence. The other supporting components have a buzzer that functions as a sound when the process is running, so the LED as a notification when the process is running too. And it is equipped with an LCD monitor as a text viewer when access is running.

Keywords: Automatic Fence, Arduino Uno, rfid, ultrasonic sensor

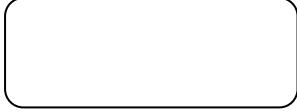
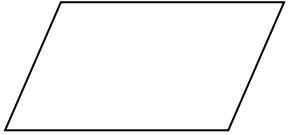
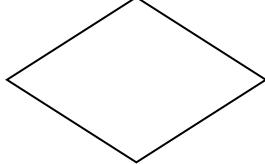
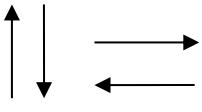
DAFTAR ISI

	Halaman
Lembar Judul Tugas Akhir	i
Lembar Pernyataan Keaslian Tugas Akhir.....	ii
Lembar Pernyataan Persetujuan Publikasi Karya Ilmiah	iii
Lembar Persetujuan dan Pengesahan Tugas Akhir	iv
Lembar Konsultasi Tugas Akhir	v
Kata Pengantar	vi
Abstraksi	viii
Daftar Isi.....	x
Daftar Simbol	xii
Daftar Gambar	xiv
Daftar Tabel.....	xv
Daftar Lampiran	xvi
 BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Maksud dan Tujuan	2
1.3. Metode Penelitian.....	3
1.4. Ruang Lingkup.....	4
 BAB II LANDASAN TEORI	5
2.1. Perangkat Keras	5
2.1.1. Teori IC Digital.....	5
2.1.2. Sumber Tegangan	6
2.1.3. Komponen Elektronika.....	7
2.1.4. Sensor Ultrasonik	12
2.1.5. Komponen Penggerak(<i>Output</i>).....	16
2.1.6. Mikrokontroler	24
2.2. Perangkat Lunak	30
2.2.1. Bahasa Pemrograman	35
2.2.2. <i>Software Editor</i>	44
 BAB III PEMBAHASAN	49
3.1. Tinjauan Umum Alat	49
3.2. Blok Rangkaian Alat	50
3.3. Skema Rangkaian Alat.....	52
3.4. Cara Kerja Alat	55
3.5. <i>Flowchart</i> Program	61
3.6. Konstruksi Sistem (<i>Coding</i>)	62
3.6.1. Initialisasi	62
3.6.2. <i>Input</i>	63
3.6.3. Main Program	64
3.6.4. <i>Output</i>	68
3.7. Hasil Percobaan	71
3.7.1. Hasil <i>Input</i>	71
3.7.2. Hasil <i>Output</i>	73

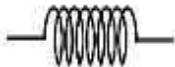
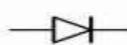
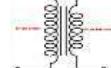
3.7.3. Hasil Keseluruhan	74
BAB IV PENUTUP	75
4.1. Kesimpulan	75
4.2. Saran	76
DAFTAR PUSTAKA.....	77
DAFTAR RIWAYAT HIDUP	79
LAMPIRAN-LAMPIRAN	81

DAFTAR SIMBOL

a. Simbol Flowchart

	TERMINAL Digunakan untuk menggambarkan awal dan akhir dari suatu kegiatan.
	INPUT/OUTPUT Digunakan untuk menggambarkan proses memasukan data yang berupa pembacaan data dan sekaligus proses keluaran yang berupa pencetakan data.
	PROCESS Digunakan untuk menggambarkan suatu proses.
	DECISION Digunakan untuk menggambarkan proses pengujian suatu kondisi yang ada.
	FLOW LINE Digunakan untuk menggambarkan hubungan proses dari satu proses ke proses lainnya.

a. Simbol Komponen Elektronika

Simbol Komponen	Nama Komponen
	Resistor
	Kapasitor
	Induktor
	Dioda
	LED
	Transistor
	Transformator
	Buzzer
	Kristal

DAFTAR GAMBAR

	Halaman
1. Gambar II.1 Catu Daya	7
2. Gambar II.2 LED	8
3. Gambar II.3 Resistor	8
4. Gambar II.4 Kapasitor	9
5. Gambar II.5 Dioda	9
6. Gambar II.6 Konektor	10
7. Gambar II.7 Kabel Jumper	11
8. Gambar II.8 Push Button	11
9. Gambar II.9 Sensor Ultrasonik HC-SR04	13
10. Gambar II.10 Timing Diagram HC-SR04.....	13
11. Gambar II.11 Mifare 522	14
12. Gambar II.12 Tag RFID	15
13. Gambar II.13 Motor Stepper (28BYJ-48)	16
14. Gambar II.14 Prinsip Kerja Motor Stepper	18
15. Gambar II.15 Motor Stepper Dengan Lilitan Unipolar	19
16. Gambar II.16 Motor Stepper Dengan Lilitan Bipolar	19
17. Gambar II.17 Motor driver ULN2003	20
18. Gambar II.18 I2C.....	21
19. Gambar II.19 LCD (<i>Liquid Crystal Display</i>)	22
20. Gambar II.20 Buzzer.....	23
21. Gambar II.21 Arduino Uno	25
22. Gambar II.22 Kaki Pin Atmega328P	26
23. Gambar II.23 Bagian-bagian Arduino Uno	27
24. Gambar II.24 Skema Bagian-bagian Arduino Uno	27
25. Gambar II.25 Tampilan Skematik Proteus	32
26. Gambar II.26 Mengambil Komponen	34
27. Gambar II.27 Arduino IDE	45
28. Gambar II.28 Verifikasi <i>Sketch</i>	47
29. Gambar II.29 Memilih Board	47
30. Gambar II.30 Memilih Port	48
31. Gambar II.31 Sukses Mengunggah <i>Sketch</i>	48
32. Gambar III.1 Blok Rangkaian Alat	50
33. Gambar III.2 Skema Rangkaian Alat	52
34. Gambar III.3 Skema Rangkaian Adaptor	55
35. Gambar III.4 Arduino Pada Rangkaian	56
36. Gambar III.5 Skema Rangkaian Sensor Ultrasonik HC-SR04	57
37. Gambar III.6 Skema Rangkaian RFID MFRC522	58
38. Gambar III.7 Skema Rangkaian Motor Stepper	58
39. Gambar III.8 Skema Rangkaian Buzzer	59
40. Gambar III.9 Skema Rangkaian LCD	59
41. Gambar III.10 Skema Rangkaian LED	60
42. Gambar III.11 <i>Flowchart</i> Program	61

DAFTAR TABEL

	Halaman
1. Tabel II.1. Pin-pin Pada HC-SR04	13
2. Tabel II.2. Pin Modul Reader RFID MFRC522	15
3. Tabel II.3. Spesifikasi Motor Stepper 28BYJ-48	17
4. Tabel II.4. Tabel Operator <i>Binary</i>	38
5. Tabel II.5. Tabel Operator <i>Unary</i>	39
6. Tabel II.6. Tabel Operator Relasi	39
7. Tabel II.7. Tabel Operator Logika	40
8. Tabel III.1. Posisi Pin dan Arduino Dan Modul	53
9. Tabel III.2. Hasil Percobaan <i>Input</i> Sensor RFID MFRC522	71
10. Tabel III.3. Hasil Percobaan <i>Input</i> Sensor HC-SR 04	72
11. Tabel III.4. Hasil Percobaan Push Button	72
12. Tabel III.5. Hasil Percobaan <i>Output</i> RFID MFRC522	73
13. Tabel III.6. Hasil Percobaan <i>Output</i> Sensor HC-SR04	73
14. Tabel III.7. Hasil Percobaan Keseluruhan Alat	74

DAFTAR LAMPIRAN

Halaman

1. Lampiran A1. Data Sheet	81
2. Lampiran B1. Listing Program Keseluruhan	105
3. Lampiran C1. Skema Rangkaian Keseluruhan	115
4. Lampiran D1. Daftar Komponen dan Daftar Harga.....	116
5. Lampiran E1. Foto Alat	117
6. Lampiran F1. Tampilan Serial Monitor	118

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi yang sangat cepat semakin terasa dalam kehidupan saat ini. Hal ini ditandai dengan perkembangan pada bidang komunikasi, kehadiran perangkat elektronika pada saat ini sangat besar manfaatnya bagi umat manusia. Hal ini dikarenakan hampir seluruh aktifitas manusia memanfaatkan sebagai sarana pendukung utama sehingga membuat kita merasakan manfaat dalam berbagai bidang kehidupan.

Pada awalnya, bentuk dan pengoprasian perangkat elektronika masih sangat sederhana, yakni dengan sistem manual. Pengendalian on/off berbagai alat elektronika saat ini kebanyakan masih dikendalikan secara manual dengan menekan tombol on/off. Seiring dengan perkembangan teknologi kebutuhan untuk mengendalikan berbagai alat elektronika dan keamanan tidak hanya dilakukan secara manual yang mengharuskan kita berada di depan alat elektronik tersebut dan menekan tombol on/off untuk mengaktifkan tetapi bisa juga dilakukan secara otomatis.

Kebanyakan dari masyarakat tidak memiliki kedipulian tinggi atau kedisiplinan yang baik terhadap lingkungan disekitar utamanya masyarakat di perkotaan sehingga tingkat keamanan dari tempat tinggal perlu di tingkatkan. Buka tutup pagar rumah menjadi salah satu sistem keamanan utama pada suatu hunian atau rumah tinggal, seiring dengan perkembangan keamanan dan kemudahan dalam mengontrol buka tutup pagar rumah maka diperlukan suatu alat yang dapat memberikan keamanan pada rumah dan kemudahan dalam pengontrolan.

Dalam kerja suatu kontrol, dapat memberikan kemudahan kepada seseorang untuk mengontrol keamanan dan kemudahan pagar rumah, tentunya dengan menggunakan alat pengontrol, salah satunya dengan menggunakan mikrokontroller. Mikrokontroller atau spesifikasinya yakni menggunakan Arduino Uno sebagai suatu terobosan teknologi dan mikrokomputer, hadir memenuhi kebutuhan pasar (market need) dan teknologi baru. Suatu controller digunakan untuk mengontrol suatu proses atau aspek – aspek dari lingkungan. Salah satu aplikasi dari Arduino Uno adalah sebagai pengatur buka tutup pagar rumah, dengan adanya alat pengontrol ini dapat memberikan kemudahan dan keamanan pada setiap orang yang memiliki kesibukan atau sering bepergian meninggalkan rumah dimana mereka dapat mengontrol keamanan dan kemudahan untuk membuka dan menutup pagar rumah saat bepergian secara otomatis. Berdasarkan latar belakang tersebut, maka penulis tertarik untuk mengangkat suatu judul yaitu ***“Prototipe Pagar Rumah Otomatis Menggunakan RFID (Radio Frequency Identification) MFRC522 Dengan Akses E-KTP Berbasis Arduino Uno”*** suatu alat yang dapat di aplikasikan atau digunakan pemilik rumah untuk berinteraksi dengan sistem keamanan dan kemudahan pada pagar rumah dengan menggunakan pengontrolan secara otomatis dengan menggunakan Arduino Uno.

1.2. Maksud dan Tujuan

Maksud dari penyusunan tugas akhir ini adalah:

1. Merancang suatu alat yang dapat bekerja secara otomatis dan efisien
2. Mengaplikasikan pelajaran elektronika dasar, rangkaian digital, dan mikrokontroler yang telah dipelajari selama kuliah.
3. Merancang suatu robot yang dapat bekerja tanpa harus dikendalikan manusia.

Sedangkan tujuan dari penulisan Tugas Akhir ini adalah sebagai salah satu syarat kelulusan Program Diploma Tiga (D.III) program studi Teknologi Komputer Fakultas Teknologi Informasi di Universitas Bina Sarana Informatika.

1.3. Metode Penelitian

Dalam penyusunan Laporan Tugas Akhir ini, penulis menggunakan metode penelitian dan pengumpulan data. Data tersebut diperoleh melalui :

1. Metode Observasi

Observasi adalah sebagai aktivitas mencatat suatu gejala dengan bantuan instrumen-instrumen dan merekamnya dengan tujuan ilmiah atau tujuan lain. Disini penulis melakukan metode observasi dengan melihat apa saja yang akan dibutuhkan pada pembuatan alat tempat sampah otomatis ini serta bertanya dan mendiskusikannya kepada seseorang yang lebih tahu tentang elektronika maupun mikrokontroler seperti dosen atau senior.

2. Studi Pustaka

Studi Pustaka adalah untuk mendapatkan landasan teori, data-data atau informasi sebagai bahan acuan dalam melakukan perencanaan, percobaan, pembuatan dan penyusunan penelitian. Dalam melakukan studi pustaka penulis mempelajari dan mengunjungi Buku-buku, jurnal-jurnal, serta website-websit yang membahas tentang mikrokontroler arduino uno dan komponen yang akan digunakan pada alat pagar otomatis.

1.4. Ruang Lingkup

Pada pembahasan laporan ini, penulis melakukan pembatasan permasalahan yang akan dibahas yaitu, Alat yang akan dirancang dapat mengontrol buka tutup pagar rumah, sistem keamanan dalam rancangan ini menggunakan **RFID** (*Radio Frequency Identification*). Adapun ruang lingkup dari permasalahan yang penulis gunakan adalah Suatu alat yang dapat di aplikasikan atau digunakan pemilik rumah untuk berinteraksi dengan sistem keamanan dan kemudahan pada pagar rumah dengan menggunakan pengontrolan secara otomatis dengan menggunakan *Arduino Uno*.

BAB II

LANDASAN TEORI

2.1. Perangkat Keras

Dalam bab ini penulis akan menjelaskan tentang teori – teori yang berhubungan dengan alat yang dibuat, meliputi perangkat keras dan perangkat lunak.

Berikut adalah komponen - komponen perangkat keras untuk membuat alat Pagar Otomatis, yang berisikan seperti Arduino UNO sebagai otak dari sistem kerja semua komponen, sensor untuk mendekripsi objek/benda, motor Stepper sebagai penggerak pagar otomatis, LCD untuk menampilkan notifikasi akses dari Module RFID Kemudian Baterai sebagai sumber tenaga.

2.1.1 Teori IC Digital

Menurut (Maulana, 2017) mengatakan bahwa, “*Integrated Circuit (IC)* adalah Sirkuit terintegrasi atau yang biasa juga disebut sebagai IC merupakan komponen elektronika yang terbuat dari kumpulan puluhan, ratusan, hingga ribuan transistor, resistor, dioda dan komponen elektronika lainnya”.

IC mempunyai fungsi dan tugas khusus yang telah ditentukan oleh pembuatnya, tugas khusus ini ditentukan dengan bagaimana cara merangkai komponen yang ada karena pada dasarnya adalah sebuah rangkaian tertentu dengan fungsi tertentu pula. *IC* Digital beroperasi pada tegangan 0 volt (*low*) dan 5 volt (*high*). *IC* digital tersusun dari beberapa rangkaian logika *AND*, *OR*, *NOT*, *NAND*, *NOR*, dan *XOR*.

2.1.2. Sumber Tegangan

Menurut (Sithohang et al., 2018) memberikan batasan bahwa, “Catu daya (*Power Supply*) adalah sebuah perangkat yang memasok listrik energi untuk satu atau lebih beban listrik”.

Catu daya menjadi bagian yang penting dalam elektronika yang berfungsi sebagai sumber tenaga listrik misalnya pada baterai atau accu. Pada dasarnya power supply ini mempunyai konstruksi rangkaian yang hampir sama yaitu terdiri dari trafo, penyearah, dan penghalus tegangan. Istilah ini paling sering diterapkan ke perangkat yang mengubah satu bentuk energi listrik yang lain, meskipun juga dapat merujuk ke perangkat yang mengkonversi bentuk energi lain (misalnya, mekanik, kimia, solar) menjadi energi listrik.

Secara umum prinsip rangkaian catu daya terdiri atas komponen utama yaitu ; transformator, dioda dan kondensator. Dalam pembuatan rangkaian catu daya selain menggunakan komponen utama juga diperlukan komponen pendukung agar rangkaian berfungsi dengan baik Ada dua sumber catu daya yaitu sumber AC dan sumber DC. Sumber AC yaitu sumber tegangan bolak – balik, sedangkan sumber tegangan DC merupakan sumber tegangan searah.

Dibawah ini adalah jenih-jenis catu daya:

1. Sumber Arus Bolak Balik (*Alternating Current/AC*)

Arus listrik bolak balik adalah arus listrik dengan besar dan arah yang berubah-ubah secara bolak balik. Arus AC mengalir bolak balik dari potensial tinggi (+) ke potensial rendah (-) dan dari potensial rendah (-) ke potensial tinggi (+), Gelombang listrik pada arus AC berbentuk *sinusoidal*, gelombang segi empat atau gelombang segitiga, contoh penggunaan arus listrik bolak-balik adalah pada

jaringan PLN dan generator AC, jika menggunakan tegangan listrik PLN, besar arusnya berkisar antara 110 *volt* hingga 220 *volt* dengan frekuensi 50 *Hertz*.

2. Sumber Arus Searah (*Direct Current/ DC*)

Arus listrik searah adalah arus listrik yang bernilai konstan dan mengalir dari potensial tinggi (+) ke potensial rendah (-). Besar arus listrik yang sering ditemukan berkisar antara 1,5 *volt* hingga 24 *volt*. Arus listrik searah biasa digunakan pada baterai, dinamo arus searah atau aki, sumber tegangan ini tidak mengalami perubahan terhadap waktu.



Sumber : <http://corelita.com/pengertian-adaptor-atau-catu-daya/>

Gambar II.1.

Catu Daya

2.1.3. Komponen Elektronika

Menurut (Maulana, 2017) disimpulkan bahwa, “Rangkaian elektronika, komponen-komponen elektronika dibagi dalam jenis komponen pasif dan komponen aktif”

1. ***Light Emitting Diode (LED)***

Menurut (Suhardi, 2014) LED adalah semikonduktor yang dapat mengubah energi listrik lebih banyak menjadi cahaya, merupakan perangkat keras dan padat (solid-state component) sehingga lebih unggul dalam ketahanan (durability).



Sumber : https://www.123rf.com/photo_19706682_set-of-color-3-mm-led-diodes-isolated-on-white-background.html

Gambar II.2

LED

1. **Resistor**

Menurut (Listiyarini, 2018) memberikan batasan bahwa “Resistor disebut juga dengan tahanan atau hambatan, merupakan komponen pasif elektronika yang berfungsi untuk membatasi arus listrik yang mengalir pada suatu rangkaian listrik”.



Sumber : <http://www.instructables.com/lesson/Resistors/>

Gambar II.3

Resistor

2. Kapasitor

Menurut (Maulana, 2017) “Kapasitor merupakan komponen pasif yang sering digunakan pada sistem yang berfungsi sebagai filter dan penyimpan energi listrik”.



Sumber : <http://www.uniksharianja.com/2016/02/cara-menentukan-nilai-kapasitas-kapasitor.html>

Gambar II.4.

Kapasitor

3. Dioda

Menurut (Maulana, 2017) memberikan batasan bahwa, “Dioda adalah komponen elektronika yang membuat arus listrik mengalir hanya dalam satu arah, sehingga biasa disebut juga sebagai penyearah”.



Sumber : <http://lapantech.com/Dioda-1n4002>

Gambar II.5

Dioda

4. Konektor

Menurut (Maulana dan Rachmat Adi Purnama, 2017) memberikan batasan bahwa, “Konektor yang digunakan dalam rangkaian elektronika, untuk memudahkan melepas pasang rangkaian konektor ini memiliki lubang pin beragam, dan disesuaikan dengan kebutuhan”.

Header atau biasa dikenal dengan nama pin *head* berguna sebagai soket tempat menghubungkan kabel-kabel konektor, ada dua jenis pin¹⁴ *header* yaitu pin *header male* dan pin *header female*, walaupun berbeda namun memiliki fungsi yang sama yaitu sebagai konektor.



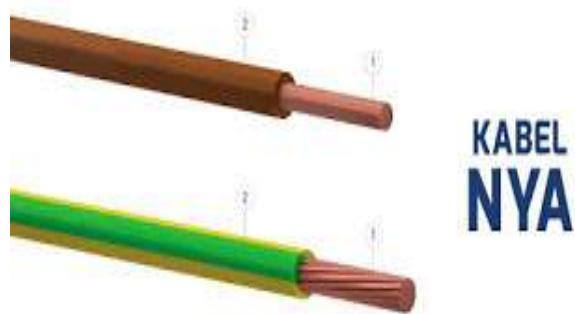
Sumber : <https://electronics.stackexchange.com/questions/37783/how-can-i-create-a-female-jumper-wire-connector>

Gambar II.6.

Konektor

5. Kabel Jumper

Menurut (Maulana dan Rachmat Adi Purnama, 2017) memberikan batasan bahwa, “Kabel Jumper digunakan untuk menyalurkan energi listrik. Sebuah kabel listrik terdiri dari isolator dan konduktor isolator adalah bahan pembungkus kabel yang biasa terbuat dari plastik atau karet, sedangkan konduktor terbuat dari tembaga”.



Sumber : https://klinikkonstruksi.jogjaprov.go.id/artikel_detail.php?id=99

Gambar II. 7

Kabel Jumper

6. Push Button Switch

Menurut (Sokop, Jendri Steven, Dringhuzen J. Mamahit, 2016) memberikan batasan bahwa, “Push button switch (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan unlock (tidak mengunci)”.

Sistem kerja unlock disini berarti saklar akan bekerja sebagai device penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal.



Sumber : <https://www.sparkfun.com/products/97>

Gambar II. 8

Push Button Switch

2.1.4. Sensor Ultrasonik

Menurut (Arasada, 2017) memberikan batasan bahwa, “Sensor Ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek atau benda tertentu didepan frekuensi kerja pada daerah diatas gelombang suara dari 20 kHz hingga 2 MHz”.

1. Sensor Ultrasonik HCSR04

Prinsip kerja sensor ini adalah transmitter mengirimkan sebuah gelombang ultrasonik lalu diukur dengan waktu yang dibutuhkan hingga datangnya pantulan dari objek. Lamanya waktu ini sebanding dengan Aplikasi Sensor Ultrasonik Untuk Deteksi Posisi Jarak Pada Ruang Menggunakan Arduino Uno. Dua kali jarak sensor dengan objek, sehingga jarak sensor dengan objek dapat ditentukan persamaan 1: $s = v \times t$ 2 Keterangan: s = jarak (meter) v = kecepatan suara (344 m/detik) t = waktu tempuh (detik) HCSR04 dapat mengukur jarak dalam rentang antara 3cm–3m dengan output panjang pulsa yang sebanding dengan jarak objek. Sensor ini hanya memerlukan 2 pin I/O untuk berkomunikasi dengan mikrokontroler, yaitu TRIGGER dan ECHO. Untuk mengaktifkan HCSR04 mikrokontroler mengirimkan pulsa positif melalui pin TRIGGER minimal 10 μ s, selanjutnya HCSR04 mengirimkan pulsa positif melalui pin ECHO selama 100 μ s hingga 18 ms, yang sebanding dengan jarak objek. Spesifikasi dari sensor ultrasonik HCSR04 adalah sebagai berikut: a. Dimensi: 24mm (P) x 20mm (L) x 17mm (T). b. Konsumsi Arus: 30 mA (rata-rata), 50 mA (max). c. Jangkauan: 3 cm–3 m. d. Sensitifitas: Mampu mendeteksi objek dengan diameter 3 cm pada jarak > 1m.



Sumber ; <https://www.tokopedia.com/arduinomakassarr/sensor-ultrasonic-hc-sr04>

Gambar II.9

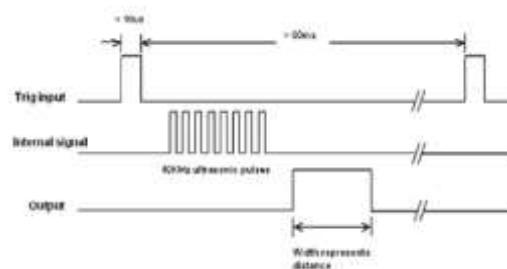
. Sensor ultrasonik HC-SR04

Tabel II. 1
Pin-pin pada HC-SR04

Pin	Keterangan
Pin 1	Vcc (dihubungkan ke tegangan +5V)
Pin 2	Trig (untuk mengirimkan gelombang suara)
Pin 3	Echo (untuk menerima pantulan gelombang suara)
Pin 4	Gnd (dihubungkan ke ground)

2. Prinsip Kerja Sensor Ultrasonik HCSR04

Prinsip kerja HCSR04 adalah transmitter memancarkan seberkas sinyal ultrasonik (20 KHz) yang bebentuk pulsa, kemudian jika didepan HCSR04 ada objek padat maka receiver akan menerima pantulan sinyal ultrasonik tersebut. Receiver akan membaca lebar pulsa (dalam bentuk PWM) yang dipantulkan objek dan selisih waktu pemancaran.



Sumber :<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

Gambar II.10

Timing diagram HC-SR04

3. ***Radio Frequency Identification (RFID)***

Menurut (Mubarok, Sofyan, Rismayadi, & Naiyah, 2018) memberikan batasan bahwa, “RFID adalah teknologi identifikasi berbasis gelombang radio. Teknologi ini mampu mengidentifikasi berbagai objek secara simultan tanpa diperlukan kontak langsung (atau dalam jarak pendek)”.

1. ***Mifare RC522***

RFID Reader Module adalah sebuah modul berbasis IC *Philips* MFRC522 yang dapat membaca RFID dengan penggunaan yang mudah dan harga yang murah, karena modul ini sudah berisi komponen-komponen yang diperlukan oleh MFRC522 untuk dapat bekerja. Modul ini dapat digunakan langsung oleh MCU dengan menggunakan interface SPI, dengan suplai tegangan sebesar 3,3 volt. MFRC522 merupakan produk dari NXP yang menggunakan *fully integrated 13.56MHz non-contact communication card chip* untuk melakukan pembacaan maupun penulisan. MFRC522 support dengan semua varian MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE *Ultralight*, MIFARE DESFire EV1 and MIFARE Plus RF *identification Protocols*.



Sumber : <https://www.indo-ware.com/produk-2865-rfid-reader-writer-module-mifare-rc522-1356-mhz-.html>

Gambar II.11.

Mifare RC522

Tabel II. 2
Pin Modul Reader RFID MFRC522

NO	Pin
1.	SDA
2.	SCK
3.	MOSI
4.	MISO
5.	IRQ
6.	GND
7.	RST
8.	3.3V

2. Tag RFID

Tag ini bekerja saat antena mendapatkan sinyal dari *reader* RFID dan sinyal tersebut akan dipantulkan lagi, sinyal pantul ini biasanya sudah ditambahkan dengan data yang dimiliki tag tersebut. RFID tag ukurannya dapat berbeda-beda, pada umumnya kecil. Tag yang ditanamkan ke dalam metal, kunci ,dan sebagainya.



Sumber : <https://www.indiamart.com/proddetail/rfid-tag-card-17920917273.html>

Gambar II.12

Tag RFID

2.1.5. Output

Sebuah alat tidak dapat bekerja apabila tidak ada *output*. Begitu juga dengan alat yang kami buat yaitu alat *prototype* pagar otomatis. Berikut adalah komponen penggerak (*output*) yang kami gunakan dalam pembuatan alat pagar otomatis yaitu *motor stepper*, *motor driver*, I2C, dan LCD. Uraian dari masing – masing hal tersebut akan dijelaskan sebagai berikut.

1. *Motor Stepper (28BYJ-48)*

Motor Stepper yang digunakan pada alat ini yaitu *motor stepper* 28BYJ48.

Motor stepper ini merupakan jenis motor stepper Unipolar. Motor ini memiliki rasio gear sebesar 64, dan putaran per stepnya sebesar 5.625° jadi motor ini memiliki step sebanyak 4096. Untuk menghitung jumlah step dalam satu putarannya dapat menggunakan rumus dibawah ini :

steps = Jumlah step per satu Putaran * Rasio Gear

$$\text{steps} = (360^\circ / 5.625^\circ) * 64 \text{ steps} = 64 * 64 \text{ steps} =$$

4096



Sumber : <https://www.instructables.com/id/Supercharge-and-Convert-a-Unipolar-Stepper-to-Bipo/>

Gambar II. 13

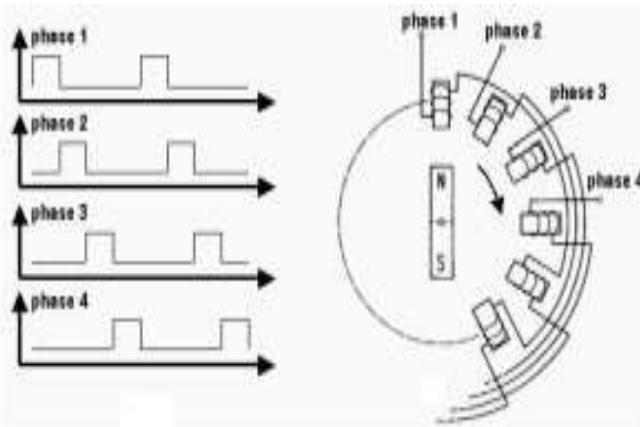
Motor Stepper

Tabel II. 3
Spesifikasi Motor Stepper 28BYJ-48

Related Voltage	5 VDC
Number of Phase	4
Speed Variation Ratio	6-20 Volt
Stride Angle	5.625°/64
Frequency	100Hz
DC Resistance	50Ω±7%(25°C)
Idle In-traction Frequency	> 600Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3mN.m(120Hz)
Self-positioning Torque	>34.3mN.m
Friction Torque	600-1200 gf.cm
Pull in Torque	300 gf.cm
Insulated Resistance	>10MΩ(500V)
Insulated Electricity	600VAC/1mA/1s
Insulated Grade	A
Rise in Temperature	<40K(120Hz)
Noise	<35dB(120Hz,No load,10cm)
Model	28BYJ-48 – 5V

a. Prinsip Kerja Motor Stepper

Prinsip kerja motor stepper adalah mengubah pulsa-pulsa input menjadi gerakan mekanis diskrit. Oleh karena itu untuk menggerakkan motor stepper diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik. Berikut ini adalah ilustrasi struktur *motor stepper* sederhana dan pulsa yang dibutuhkan untuk menggerakkannya



Sumber : <http://zonaelektronik.net/motor-stepper/>

Gambar II. 14

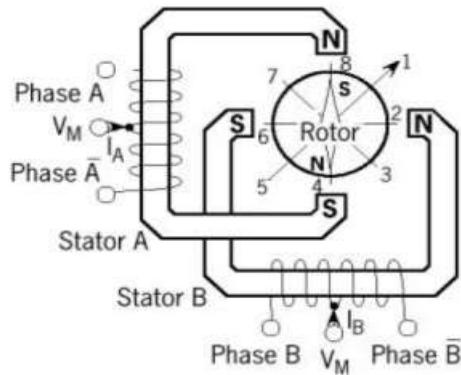
Prinsip Kerja *Motor Stepper*

Gambar diatas memberikan ilustrasi dari pulsa keluaran pengendali motor stepper dan penerapan pulsa tersebut pada motor stepper untuk menghasilkan arah putaran yang bersesuaian dengan pulsa kendali.

b. Jenis – Jenis *Motor Stepper*

1) *Motor Stepper Unipolar*

Rangkaian pengendali motor stepper unipolar lebih mudah dirancang karena hanya memerlukan satu switch / transistor setiap lilitannya. Untuk menjalankan dan menghentikan motor ini cukup dengan menerapkan pulsa digital yang hanya terdiri atas tegangan positif dan nol (ground) pada salah satu terminal lilitan (wound) motor sementara terminal lainnya dicatut dengan tegangan positif konstan (VM) pada bagian tengah (center tap) dari lilitan seperti pada gambar berikut.



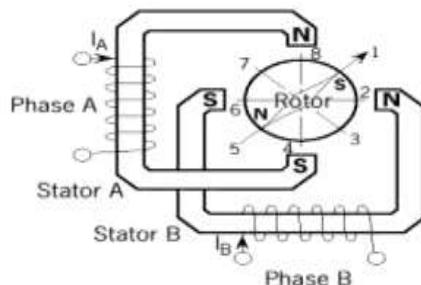
Sumber : <http://zonaelektronik.net/motor-stepper/>

Gambar II. 15

Motor stepper dengan lilitan unipolar

2) Motor Stepper Bipolar

Untuk *motor stepper* dengan lilitan bipolar, diperlukan sinyal pulsa yang berubah-ubah dari positif ke negatif dan sebaliknya. Jadi pada setiap terminal lilitan (A & B) harus dihubungkan dengan sinyal yang mengayun dari positif ke negatif dan sebaliknya. Karena itu dibutuhkan rangkaian pengendali yang agak lebih kompleks daripada rangkaian pengendali untuk motor unipolar. *Motor stepper bipolar* memiliki keunggulan dibandingkan dengan motor stepper unipolar dalam hal torsi yang lebih besar untuk ukuran yang sama.



Sumber:<http://zonaelektronik.net/motor-stepper/>

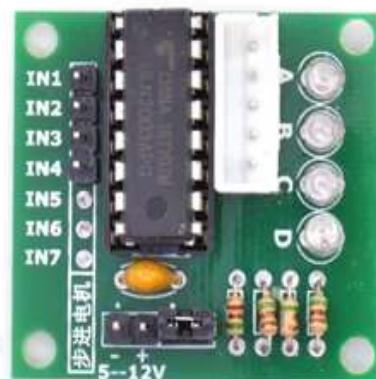
Gambar II. 16

Motor stepper dengan lilitan bipolar

2. ***Motor Driver ULN2003***

Motor Driver ULN2003 merupakan perangkat yang mempermudah mengontrol Motor Stepper 28BYJ-48 dari mikrokontroler, seperti Arduino.

Pada motor driver ULN2003 ini terdapat 5 *socket* kabel untuk menyambungkan kabel dari motor stepper 28BYJ-48 dan 4 LED untuk menunjukkan coil yang sedang diberi tegangan. Untuk tegangannya, dianjurkan menggunakan power supply 5-12Volt 1Amp. Dan pada motor driver ini juga terdapat 4 input kontrol yang harus dihubungkan ke empat Arduino pin digital.



Sumber:<https://www.barerobotics.com/product/uln2003-unipolar-stepper-motor-driver-12v-05a>

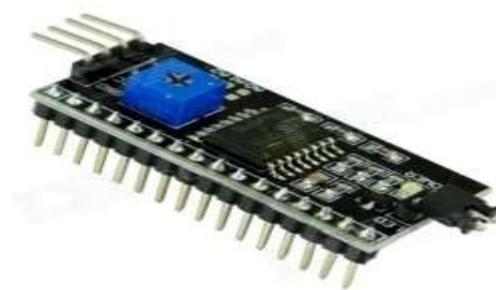
Gambar II. 17

Motor Driver ULN2003

3. I2C (*Inter Integrated Circuit*)

Menurut (Sari, Kartika, Cucu Suhery, 2015) memberi batasan bahwa, “I2C merupakan singkatan dari *Inter Integrated Circuit*, yaitu sebuah protokol untuk komunikasi serial antar IC dan sering disebut juga *Two Wire Interface* (TWI). Pada penelitian ini, komunikasi I2C digunakan untuk komunikasi antara mikrokontroler dan *RTC*”.

Komunikasi I2C merupakan standar komunikasi dengan menggunakan 2 kabel dengan kecepatan yang cukup tinggi sehingga antara IC yang satu dengan yang lainnya dapat berkomunikasi dengan baik. Pada penelitian ini, komunikasi I2C digunakan untuk meng-hubungkan antara RTC dengan Arduino.



Sumber : <http://www.dx.com/p/lcd1602-adapter-board-w-iic-i2c-interface-black-works-with-official-arduino-boards-216865>

Gambar II.18

I2C

4. LCD (*Liquid Crystal Display*)

Menurut (Dinata dan Wahri Sunanda, 2015) memberikan batasan bahwa, “LCD berfungsi untuk menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan menu pada aplikasi mikrokontroler”.

Modul LCD matrix tersedia dengan konfigurasi 16 karakter dan 2 baris dengan setiap karakternya dibentuk oleh baris pixel.

Adapun fitur yang disajikan dalam LCD ini adalah:

- a. Terdiri dari 16 karakter dan 2 baris.
- b. Mempunyai 192 karakter tersimpan.
- c. Terdapat karakter generator terprogram.
- d. Dapat dialamati dengan mode 4-bit dan 8-bit.
- e. Dilengkapi dengan *back light*.



Sumber : <https://www.indiamart.com/proddetail/adraxx-hd44780-16-x-2-lcd-module-diy-general-microcontroller-16393243033.html>

Gambar II.19

LCD

5. *Buzzer*

Menurut (Mubarok et al., 2018) memberikan batasan bahwa, “Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara.”.

Prinsip kerja buzzer hampir sama dengan *loud speaker*, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara.



Sumber: <https://www.indiamart.com/proddetail/horn-buzzer-15460445088.html>

Gambar II. 20

Buzzer

2.1.6. Mikrokontroler

Menurut (Sujarwata, 2018) mengatakan bahwa, “Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran (*I/O*) serta pengendali (kontrol) dengan suatu program yang dapat ditulis dan dihapus dengan cara khusus”.

Mikrokontroler berfungsi sebagai pengontrol rangkaian elektronika dan pada umumnya dapat menyimpan program yang pada dasarnya menggunakan bahasa *assembler*. Saat ini mikrokontroler dapat diprogram dengan menggunakan bahasa tingkat tinggi seperti *BASIC*, *PASCAL* atau *C*. Agar semua mikrokontroler dapat berfungsi, maka mikrokontroler tersebut memerlukan komponen eksternal yang kemudian disebut dengan *system minimum*. Untuk membuat sistem minimal paling tidak dibutuhkan sistem *clock* dan *reset*, walaupun pada beberapa mikrokontroler sudah menyediakan sistem *clock* internal, sehingga tanpa rangkaian eksternal pun mikrokontroler sudah beroperasi. Mikrokontroler berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah *personal computer (PC)*, karena sebuah mikrokontroler pada umumnya telah berisi komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka *I/O*. Mikrokontroler cenderung beroperasi pada manipulasi bit, sedangkan mikroprosesor cenderung beroperasi pada operasi *byte* (8bit).

Pada alat pagar otomatis ini menggunakan mikrokontroler ATmega328P, dan implementasi yang digunakan dari ATmega328P adalah *platform* pengembangan Arduino yaitu model Arduino Uno.

Sedangkan Arduino menurut (Ahyadi, 2018) mengatakan bahwa “Arduino adalah suatu *open-source platform* elektronik yang berbasis kemudahan penggunaan (*easy to use*) baik *hardware* maupun *software*”.

1. Arduino Uno

Menurut (Kadir, 2017) mengatakan bahwa, “Arduino Uno adalah papan yang mengandung sebuah mikrokontroler buatan *Atmel* (Atmega328p) yang menjadi pusat pengendali perangkat keras dan sejumlah pin untuk kepentingan operasi masukan (*input*) dan keluaran (*output*)”.

Hardware mikrokontroler Arduino diprogram dengan menggunakan bahasa pemrograman *wiring-based* yang berbasiskan *syntax* dan *library*. Pemrograman *wiring-based* ini tidak jauh berbeda dengan C/C++, tetapi dengan beberapa penyederhanaan dan modifikasi. Untuk memudahkan dalam pengembangan aplikasinya, mikrokontroler Arduino juga menggunakan *Integeated Development Environment (IDE)* berbasis *processing*. Mikrokontroler Arduino dapat dipasangkan dengan bermacam-macam sensor. Adapun sensor yang dapat dipasangkan pada Arduino seperti sensor gerak, ultrasonik, panas, suara, asap, gas, dan lain-lain.



Sumber : Akhiruddin (2017:36)

Gambar II.21

Arduino Uno

Konfigurasi yang terdapat pada Arduino Uno menurut (Sokop, Jendri Steven, Dringhuzen J. Mamahit, 2016) adalah:

- a. Mikrokontroler : ATmega328
- b. Tegangan pengoperasian : 5V
- c. Tegangan *input* yang disarankan: 7-12V.
- d. Batas tegangan *input* : 6-20V.
- e. Jumlah pin *I/O* digital : 14.
- f. Jumlah pin *input* analog : 6.
- g. Arus *DC* tiap pin *I/O* : 40 mA.
- h. Arus *DC* untuk pin 3.3V : 50 mA.
- i. Memori : 32 KB (ATmega328).

sekitar 0.5 KB digunakan oleh *bootloader*.

- j. SRAM : 2 KB (ATmega328).
- k. EEPROM : 1 KB (ATmega328).
- l. Clock Speed : 16 MHz.

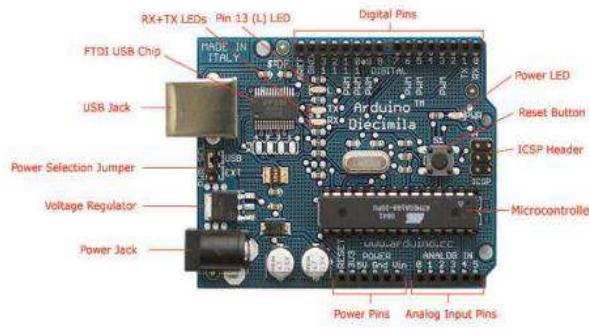


Sumber:<http://www.labelektronika.com/2017/02/arduino-uno-mikrokontroler-atmega-328.html>

Gambar II.22

Kaki Pin Atmega328

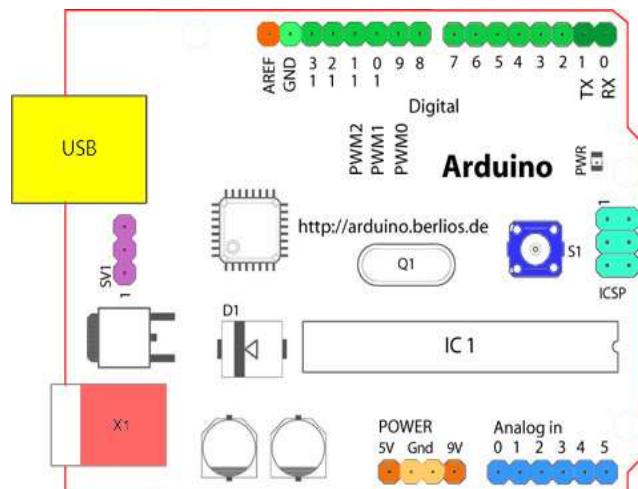
2. Berikut ini bagian-bagian Arduino Uno yang terdapat pada gambar:



Sumber : Dinata dan Wahri Sunanda (2015:85)

Gambar II.23

Bagian-bagian Arduino Uno



Sumber : <https://www.arduino.cc/en/reference/board>

Gambar II.24

Skema Bagian-bagian Arduino Uno

- a. *USB*

USB digunakan untuk mengupload program dari komputer ke dalam *Board* Arduino dan digunakan untuk memberikan catu daya ke Papan Arduino menggunakan kabel *USB* dari komputer.

b. Kristal (*quartz crystal oscillator*)

Kristal merupakan jantung dari *Board Arduino* itu sendiri, dikarenakan kristal menghasilkan detak-detak yang dikirim kepada mikrokontroler agar melakukan sebuah operasi untuk setiap detaknya, yang biasanya berdetak 16 juta kali per detik ($16MHz$).

c. Tombol *Reset*

Tombol *reset* digunakan untuk mereset *Board Arduino* sehingga program akan memulai dari awal kembali. Tombol ini tidak untuk menghapus program atau mengosongkan mikrokontroler.

d. Pin Analog *Input* (0 – 5)

Pin analog digunakan untuk membaca tegangan yang dihasilkan oleh sensor analog. Nilai pin *output* analog dapat diprogram dari 0 – 255 dan dapat membaca nilai analog dari 0 – 1023, yang keduanya mewakili nilai tegangan 0-5 *Volt*.

e. Pin Digital *I/O* (0 – 13)

14 pin digital ini berfungsi sebagai *input* atau *output* dan dapat diatur oleh program. Khusus untuk pin 3, 5, 6, 9, 10, dan 11 dapat juga berfungsi sebagai pin analog *output* yang tegangan *outputnya* dapat diatur.

f. Eksternal *Power Supply*

Sumber tenaga eksternal adalah penyuplai tegangan *DC* sebesar 9–12 *Volt*.

g. *In Circuit Serial Programming (ICSP)*

Port *ICSP* digunakan untuk memprogram mikrokontroler tanpa melalui *bootloader*. *ICSP* menggunakan jalur *Serial Peripheral Interface (SPI)* untuk transfer data. Namun pada umumnya para pengguna tidak melakukan hal ini.

h. *IC* – mikrokontroler atmega

IC merupakan komponen utama dari *Board* Arduino yang didalamnya terdapat *CPU*, *ROM*, dan *RAM*.

i. Pin *Power*

Pin *power* yang terdapat pada Arduino Uno adalah sebagai berikut:

- 1) *Vin* = pin yang digunakan jika ingin memberikan *power* langsung ke *board* Arduino dengan tegangan 7V – 12V.
- 2) *5V* = pin *output 5V* yang telah diatur oleh regulator *board* Arduino. *Board* dapat diaktifkan dengan daya dari *Power Supply* (7 - 12V), *USB* (5V), atau pin *Vin board* (7-12V). Jika tegangan di *input* melalui pin *5V* atau *3.3V* secara langsung tanpa melewati regulator, maka dapat merusak *board* Arduino.
- 3) *3.3V* = dihasilkan oleh regulator *on board*. Menyediakan arus maksimum 50 mA.
- 4) *GND* = Pin *Ground*.
- 5) *IOREF* = memberikan tegangan referensi ketika mikrokontroler beroperasi. Biasa digunakan pada *board shield* untuk dapat memilih sumber daya yang tepat agar dapat bekerja dengan 5V atau 3.3V.

2.2. Perangkat Lunak

Menurut (Swara, Kom, & Pebriadi, 2016) memberi batasan bahwa, “Perangkat Lunak merupakan seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program maupun prosedur yang didalamnya merupakan kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi”.

Perangkat lunak atau *software* yang digunakan dalam pembuatan alat pagar otomatis menggunakan sensor sonar, rfid mfrc522 ini adalah *software proteus*, *Flowchart*, *Software Arduino IDE (Integrated Development Environment)* yang menggunakan bahasa C.

1. *Flowchart*

Menurut (Nuraini, 2015) menyimpulkan bahwa:

“*Flowchart* dapat diartikan sebagai suatu alat atau sarana yang menunjukkan langkah-langkah yang harus dilaksanakan dalam menyelesaikan suatu permasalahan untuk komputasi dengan cara mengekspresikannya ke dalam serangkaian simbol-simbol grafis khusus, *flowchart* sangat berguna khususnya untuk menjelaskan urutan-urutan proses yang pelaksanaannya mempunyai banyak pilihan atau percabangan”.

Bentuk diagram alur (*flowchart*) yang sering digunakan dalam proses pembuatan suatu program komputer adalah sebagai berikut:

a. Program *Flowchart*

Simbol-simbol yang menggambarkan suatu urutan prosedur secara rinci dan detail antara instruksi yang satu dengan yang lainnya di dalam suatu program komputer yang bersifat *logic*.

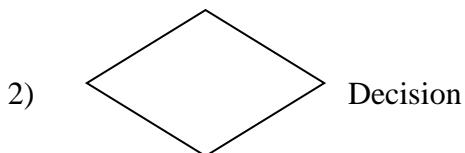
b. Sistem *Flowchart*

Simbol-simbol yang menggambarkan urutan prosedur secara *detail* di dalam suatu sistem komputer, dan bersifat fisik.

Simbol-simbol yang digunakan adalah sebagai berikut.



Menyatakan awal dan akhir dari suatu kegiatan.



Simbol ini menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya/tidak.



Simbol proses menggambarkan proses yang ada dalam program.



Simbol yang menggambarkan masukan dan keluaran dalam program

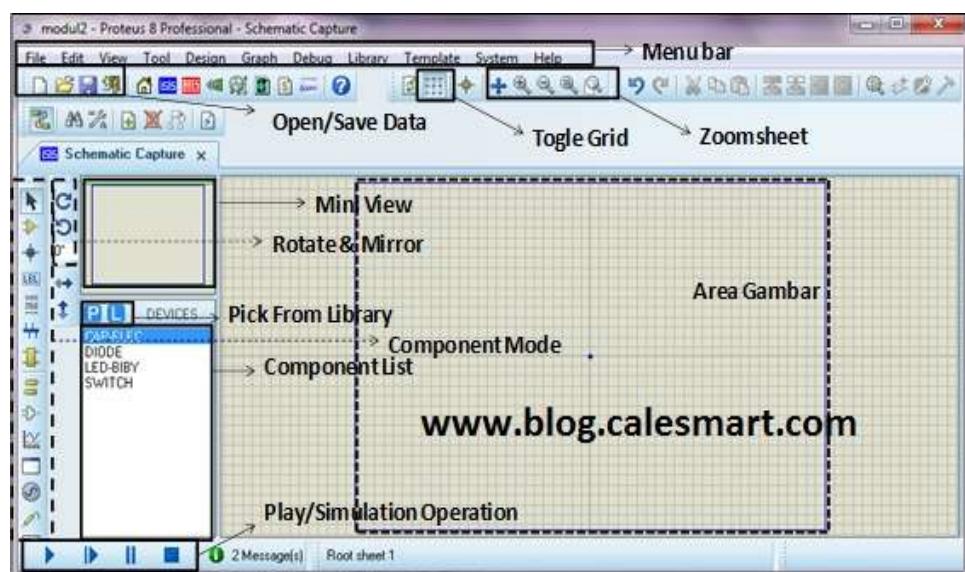


Simbol yang digunakan untuk menggambarkan hubungan proses dari satu proses ke proses lainnya.

2. Software Proteus

Menurut (Falani, 2015) memberikan batasan bahwa, “*Proteus* adalah sebuah software untuk mendesain *PCB* yang juga dilengkapi dengan simulasi pspice pada level skematik sebelum rangkaian skematik diupgrade ke *PCB* sehingga sebelum *PCB*nya di cetak, kita akan tahu apakah *PCB* yang akan kita cetak sudah benar atau tidak”.

Proteus mengkombinasikan program *ISIS* untuk membuat skematik desain rangkaian dengan program *ARES* untuk membuat layout *PCB* dari skematik yang kita buat. Software ini bagus digunakan untuk desain rangkaian mikrokontroller. *Proteus* juga bagus untuk belajar elektronika seperti dasar-dasar elektronika sampai pada aplikasi mikrokontroller. Software ini jika di install menyediakan banyak contoh aplikasi desain yang disertakan sehingga kita bisa belajar dari contoh-contoh yang sudah ada.



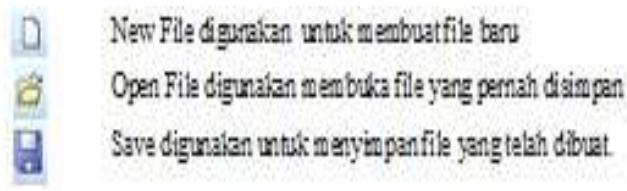
Sumber:https://calesmart.com/artikel/Pengenalan-simulasi-elektronika-dengan-Software-Proteus-8-Profesional_132.html

Gambar II. 25

Tampilan Skematik Proteus

Keterangan *Menu Bar* pada *Proteus* :

- a. Menu Bar Merupakan List menu yang dapat digunakan dalam perancangan atau pengolahan rangkaian.
- b. *Open, save*



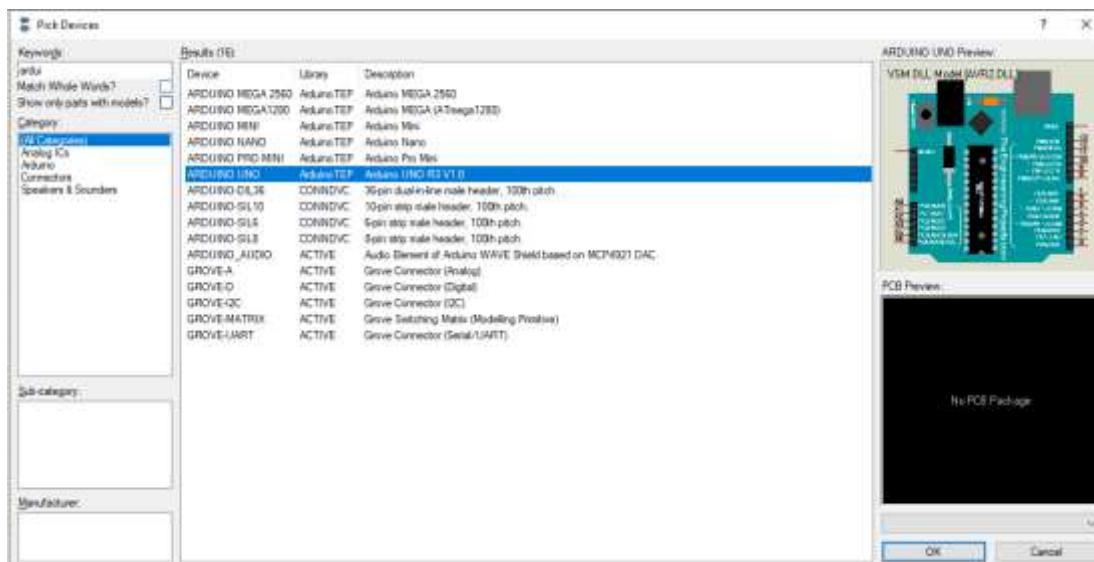
- c. *Toggle grid* digunakan untuk menampilkan bantuan titik-titik panduan pada area gambar.
- d. *Zoom sheet* untuk menentukan area tengah gambar dengan betumpu pada kursor, memperbesar dan memperkecil gambar.
- e. Menu View Digunakan untuk menampilkan gambar dalam bentuk tampilan kecil seluruh area gambar.
- f. *Componen list* merupakan daftar komponen yang telah diambil dari library
- g. *Componen Mode*



- h. *Rotate* dan *Mirror* digunakan untuk merotasi obyek.
- i. *Play* dan *Simulation Operation* digunakan untuk menjalankan simulasi rangkaian yang telah dibuat.

Langkah – langkah pembuatan rangkaian simulasi di dalam proteus dapat diuraikan sebagai berikut :

- a. Jalankan Program Proteus
- b. Langkah selanjutnya adalah mengambil komponen yang dibutuhkan, dengan cara pilih mode komponen atau klik symbol P (pick from library) untuk Menuju library.
- c. Akan muncul window baru, selanjutnya mengambil komponen yang dibutuhkan dengan mengetikkan keyword pada kolom keyword.
- d. Menambil komponen yang dibutuhkan dilakukan dengan cara mengetikkan keyword komponen atau dobel klik pada komponen yang dipilih atau sampai muncul pada list komponen pada window utama.



Sumber : Dokumen Pribadi

Gambar II. 26

Mengambil Komponen

2.2.1. Bahasa Pemprograman

Menurut (Iswanto dan Nia Maharani Raharja, 2015) menyimpulkan bahwa:

“Bahasa C pertama kali digunakan di komputer Digital Equipment Corporation PDP-11 yang menggunakan sistem operasi UNIX C adalah bahasa yang standar, artinya suatu program yang ditulis dengan bahasa C tertentu akan dapat dikonversi dengan bahasa C yang lain dengan sedikit modifikasi, Standar bahasa C yang asli adalah standar dari UNIX”.

Berikut ini adalah fungsi-fungsi dari bahasa C :

1. Struktur

Struktur dalam penulisan program bahasa C mencakup tiga hal yaitu File *Header*, Fungsi Utama, Fungsi Lain. Pemrograman bahasa C biasa menggunakan fungsi utama Main() yang akan dijalankan pertama kali pada saat eksekusi program, meskipun didalam fungsi utama ini terdapat statement yang memanggil fungsi lain. Sedangkan didalam program Arduino (*Sketch*) memiliki dua fungsi yang diperlukan yang setiap fungsinya melampirkan blok-blok pernyataan.

void setup()

{

statements;

}

void loop()

{

statements;

}

a. *Void setup() { }*

Fungsi *setup()* akan dipanggil ketika *sketch* dijalankan pertama kali dan hanya akan berjalan sekali, yaitu setiap *power-up* atau *restart board* Arduino. Struktur fungsi ini untuk menginisialisasi mode pin, *variabel*, memulai penggunaan *library*, dan lain-lain.

b. *Void Loop() { }*

Setelah fungsi *setup()* selesai, fungsi *loop()* akan dijalankan secara terus menerus sampai catu daya (*power supply*) dilepas. Fungsi ini secara aktif mengontrol *board* Arduino baik secara *input* maupun *output*.

2. *Further Syntax*

- a. ; (titik koma) digunakan pada akhir setiap baris kode.
- b. {} (kurung kurawal) untuk mendefinisikan awal fungsi dan akhir fungsi.
- c. // (baris komentar) digunakan untuk memberi satu baris catatan dari arti kode-kode yang dituliskan.
- d. /**/ (blok komentar) digunakan untuk memberi dua atau lebih baris catatan dari arti kode-kode yang dituliskan.
- e. #include yaitu satu jenis pengarah praprosessor yang dipakai untuk membaca file yang dinamakan file judul (*header file*).

3. *Variabel*

Variabel adalah cara penamaan dan menyimpan nilai numerik yang nantinya digunakan oleh program. Sebuah *variabel* perlu dinyatakan dan diberikan secara opsional ke nilai yang perlu disimpan. *int inputVariable*

```
= 0; // menjelaskan sebuah variabel dan memberi nilai 0 inputVariable =  
analogRead(2); // menetapkan variabel ke nilai pin analog 2 input  
Variable adalah variabel itu sendiri.
```

4. Tipe Data

- a. *Char* : Memiliki ukuran memori 1 *Byte* dengan jangkauan nilai -128 s/d 127 dan menyimpan 1 karakter menggunakan kode *ASCII*.
- b. *Byte* : Menyimpan nilai numerik 8-bit tanpa titik desimal dan memiliki jangkauan nilai 0-255.
- c. *Int* : Menyimpan bilangan tanpa titik desimal dan memiliki ukuran memori 2 *Byte*(16-bit) dengan jangkauan nilai -32,768 s/d 32,767.
- d. *Long* : Tipe data *extended* untuk bilangan *long integer* dan memiliki ukuran memori 4 *Byte* dengan jangkauan nilai -2,147,435,648 s/d 2,147,435,647.
- e. *Float* : Tipe data untuk angka yang memiliki titik desimal dan memiliki ukuran memori 4 *Byte* dengan jangkauan nilai -3.4028235E+38 s/d 3.4028235E+38.
- f. *Double* : Memiliki ukuran memori 8 *Byte* dengan jangkauan nilai -1,7x10⁻³⁰⁸ s/d 1,7x10⁽⁺³⁰⁸⁾.
- g. *Boolean* : Digunakan untuk menyimpan nilai *TRUE* (benar) atau *FALSE* (salah) dan hanya menggunakan 1 bit dari *RAM*.

5. Array

Adalah tipe terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe yang sama. Banyaknya komponen dalam array ditunjuk oleh suatu indeks untuk membedakan antar variabel.

Format: *TipeData NamaVariabel [ukuran]*

- a. *Type* data, untuk menyatakan *type* data yang digunakan.
- b. Nama variabel, untuk menyatakan nama variabel yang digunakan.
- c. Ukuran, untuk menyatakan jumlah maksimum elemen array.

Contoh: *Float Jumlah[8]*

6. Operator Bahasa

Merupakan simbol atau kata yang digunakan dalam program untuk melakukan suatu operasi atau manipulasi, seperti menjumlahkan dua buah nilai, memberikan nilai suatu variabel, membandingkan dua buah nilai, dan lain-lain. Berikut ini operator-operator yang biasa digunakan dalam bahasa C, yaitu:

a. Operator Aritmatika

Operator ini digunakan untuk memanipulasi angka yang bekerja seperti matematika sederhana. Operator Aritmatika digolongkan dalam Operator *binary* dan operator *unary*.

Tabel II.4

Tabel Operator *Binary*

Operator	Keterangan	Contoh
+	Penjumlahan	$4 + 5$
-	Pengurangan	$3 - 1$
*	Perkalian	$4 * 2$
/	Pembagian	$8 / 4$
%	Sisa Pembagian (mod)	$5 \% 2$

Sumber : www.duniailkom.com

Tabel II.5
Tabel Operator *Unary*

Operator	Keterangan	Contoh
+	Tanda Plus	+5
-	Tanda Minus	-5

Sumber : www.duniailkom.com

b. Operator Relasi / Perbandingan

Operator ini digunakan untuk membandingkan dua buah nilai dengan hasil berupa nilai numerik 1 (*True*) atau 0 (*False*).

Tabel II.6
Tabel Operator Relasi

Operator	Keterangan
==	Sama dengan (bukan pemberi nilai)
!=	Tidak Sama dengan
>	Lebih dari
<	Kurang dari
>=	Lebih dari atau Sama dengan
<=	Kurang dari atau Sama dengan

Sumber : www.duniailkom.com

c. Operator Logika / Boolean

Digunakan untuk mengekspresikan satu atau lebih data atau ekspresi logika (boolean) dengan menghasilkan data logika (boolean) baru.

Tabel II.7.

Operator Logika

Operator	Keterangan
&&	Operator Logika AND
	Operator Logika OR
!	Operator Logika NOT

Sumber : www.duniailkom.com

Operator Logika AND digunakan untuk menghubungkan dua atau lebih ekspresi relasi, akan bernilai benar jika semua ekspresi relasi yang dihubungkan bernilai benar. Operator Logika OR digunakan untuk menghubungkan dua atau lebih ekspresi relasi, akan bernilai benar jika salah satu ekspresi relasi yang dihubungkan bernilai benar dan akan bernilai salah jika semua ekspresi relasi yang dihubungkan bernilai salah. Operator Logika NOT akan memberikan nilai kebalikan dari ekspresi yang dihubungkan, jika ekspresi bernilai benar maka akan menghasilkan nilai salah, begitu dengan sebaliknya.

7. Konstanta

Adalah nilai yang telah ditentukan yang digunakan untuk membuat program lebih mudah dibaca. Konstanta dikelompokkan menjadi sebagai berikut:

a. *TRUE/FALSE*

Adalah konstanta boolean yang mendefinisikan level logika.

FALSE didefinisikan sebagai 0 (nol), sedangkan *TRUE* didefinisikan sebagai 1 (satu).

```
If (b == TRUE);  
{ pernyataan; }
```

b. *HIGH/LOW*

Konstanta ini menentukan level tinggi atau rendah dan digunakan saat membaca atau menulis ke pin digital. *HIGH* didefinisikan sebagai logika 1, *on*, atau 5 volt. Sedangkan *LOW* didefinisikan sebagai logika 0, *off*, atau 0 volt.

```
digitalwrite(13, HIGH);
```

c. *INPUT/OUTPUT*

Konstanta ini digunakan dengan fungsi pinMode() untuk menentukan mode pin digital sebagai *input* atau *output*.

```
pinMode(13, OUTPUT);
```

8. Struktur Pengaturan

Sebuah program mempunyai suatu struktur dan pengaturan untuk menjalankan perintah programnya, yang biasa disebut juga dengan seleksi kondisi. Beberapa seleksi kondisi yang biasa digunakan dalam bahasa C sebagai berikut:

a. Perintah *If*

Digunakan untuk memilih satu dari dua atau lebih pernyataan.

Format: *if (kondisi)*

```
{ pernyataan }
```

b. Perintah *If...Else*

Digunakan untuk memilih satu dari dua atau lebih pernyataan dengan menjalankan pernyataan *if* terlebih dahulu, jika kondisinya *TRUE* maka akan diteruskan perintah programnya, jika kondisinya *FALSE* maka pernyataan *else* yang akan dijalankan.

Format: *if (kondisi) { Pernyataan }*

Else { pernyataan }

c. Perintah *If* dalam *If*

Perintah *if* dalam *if* sering disebut *nasted-if*, dengan menjalankan kondisi 1 *if* terlebih dahulu kemudian meneruskannya ke kondisi 2 *if* yang jika pernyataan 1 benar/*TRUE* maka diteruskan perintah programnya, jika kondisinya *FALSE* maka pernyataan 2 yang akan dijalankan.

Format: *if (kondisi1)*

{ if (kondisi2) { pernyataan1 } }

Else { pernyataan2 } }

d. Perintah *For*

Digunakan untuk melakukan perulangan pada suatu kondisi menggunakan perhitungan (*counter*) yang pasti.

Format: *for (inisialisasi; syarat perulangan; pengubah nilai pencacahan)*

{ pernyataan; }

Inisialisasi merupakan bagian untuk memberikan nilai awal untuk variabel-variabel tertentu. Syarat Perulangan yaitu memegang kontrol terhadap perulangan, karena bagian ini yang akan menentukan suatu

perulangan diteruskan atau dihentikan. Pengubah Nilai Pencacah; mengatur kenaikan atau penurunan nilai pencacah.

9. Digital *I/O*

I/O digital pada Arduino mempunyai fungsi sendiri untuk mengaksesnya. Fungsi ini menyederhanakan perintah yang berhubungan dengan pin *I/O* Arduino.

a. *pinMode* (pin, mode)

Berfungsi untuk mengkonfigurasi pin tertentu dan digunakan sebagai *input* atau *output*. Pin adalah nomor pin yang akan digunakan dari 0-19 (pin analog 0-5 adalah 14-19).

b. *digitalWrite* (pin, value)

Berfungsi untuk memberi nilai *HIGH* atau *LOW* ke pin digital. Jika pin dikonfigurasi sebagai *OUTPUT* dengan *pinMode()*, maka tegangan akan diatur ke nilai yang sesuai, yaitu 5V atau 3.3V untuk *HIGH* dan 0V (*ground*) untuk *LOW*. Jika pin dikonfigurasi sebagai *INPUT* dengan *pinMode()*, maka *digitalWrite()* akan mengaktifkan (*HIGH*) atau menonaktifkan (*LOW*) dengan pullup internal pada pin *input*.

c. *digitalRead*(pin)

Berfungsi untuk membaca nilai dari pin digital tertentu, baik itu *HIGH* atau *LOW*.

10. Analog *I/O*

Berikut ini adalah fungsi analog dari Arduino:

a. *AnalogRead*(pin)

Berfungsi untuk membaca nilai analog melalui pin analog. Pada Arduino Uno sendiri memiliki 6 *channel* analog. Ketika pin analog

ditetapkan sebagai *INPUT*, maka keluaran voltase dapat dibaca, yang keluarannya antara angka 0 (0V) dan 1024 (5V).

b. *AnalogWrite(pin, value)*

Pin pada Arduino ada yang mendukung *PWM* (*pulse width modulation*) yaitu pin 3, 5, 6, 9, 10, 11, yang dapat merubah pin hidup (*on*) atau mati (*off*) dengan cepat sehingga dapat membuatnya berfungsi seperti keluaran analog. *Value* atau nilai pada format kode tersebut adalah antara angka 0.

(0% *duty cycle* – 0V) dan 255 (100% *duty cycle* – 5V).

11. *Delay(time)*

Fungsi ini untuk menghentikan sementara program selama waktu yang ditentukan dalam milidetik. 1000 sama dengan 1 detik

Delay(1000); //tunggu 1 detik

2.2.2. *Software Editor*

1. *Arduino IDE*

Menurut (ANANDA, 2018) mengatakan bahwa, “*Arduino IDE* (*Integrated Development Program*) merupakan *software* yang dipakai untuk membuat *sketch* yang akan di *upload* ke *board arduino sendiri*”.

Arduino IDE juga tergolong bentuk *software* pengembang program yang terintegrasi sehingga berbagai *tools* atau tampilan disediakan dan dinyatakan dalam bentuk antarmuka berbasis menu. Dengan menggunakan *Arduino IDE*, kesalahan penulisan *sketch* atau kebenaran penulisan *sketch* langsung dapat dibuktikan.



Sumber : <https://www.arduino.cc>

Gambar II.27

Arduino *IDE*

2. Bagian-bagian Arduino *IDE*

Berikut adalah fungsi *toolbar* yang terdapat pada Arduino *IDE* :



Verify : Berfungsi untuk melakukan *checking* kode yang dibuat pada kesalahan kompilasi.



Upload : Berfungsi untuk melakukan kompilasi program dan menguploadnya ke *Board* arduino yang sudah dikonfigurasi.



New : Berfungsi untuk membuat *sketch* baru.



Open : Berfungsi untuk membuka *sketch* yang sudah pernah disimpan.



Save : Berfungsi untuk menyimpan *sketch* yang telah dibuat.



Serial Monitor : Berfungsi untuk membuka *serial monitor*. *Serial*

monitor merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara arduino dengan *sketch* pada port *serialnya*. *Serial monitor* ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, dan pesan *error*. Perintah tambahan dapat ditemukan dalam lima menu: ***File, Edit, Sketch, Tool, Help***.

3. Menulis *Sketch*

Jalankan program arduino dari file master arduino yang kita simpan ataupun dari ikon di komputer desktop. Kemudian akan muncul program arduino *IDE* dengan tampilan *sketch* sesuai dengan tanggal dan bulan pada saat program dijalankan. *Sketch* atau kode pemrograman Arduino ditulis dibagian editor *text* dan disimpan dalam file dengan ekstensi .ino.

4. Verifikasi *Sketch*

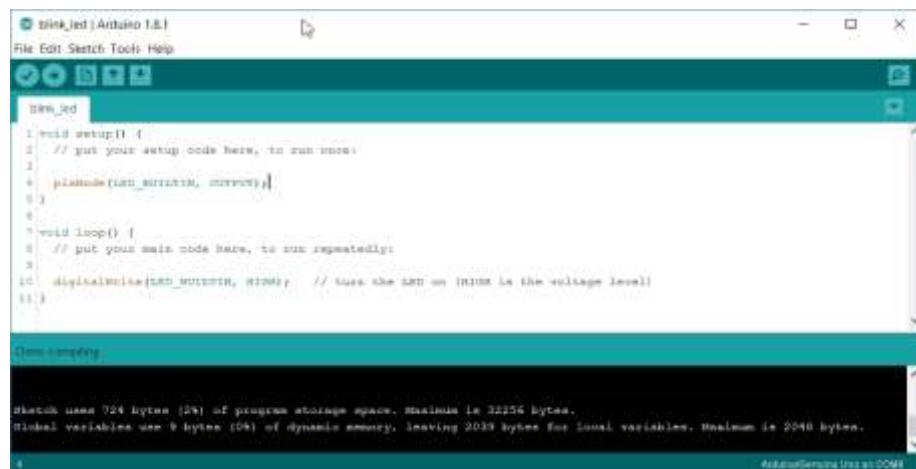
Jika *sketch* telah disimpan selanjutnya adalah verifikasi memastikan bahwa kode program telah benar. Klik tombol verifikasi untuk melakukannya, dan jendela pesan akan muncul “*Done Compiling*” jika verifikasi sukses.

a. Memilih Jenis dan Port *Board* Arduino

Sebelum mengupload *sketch* ke *board* arduino, diharuskan memilih jenis dan port *board* Arduino yang digunakan. Menu ***tools > Board*** untuk memilih *board* yang sesuai, menu ***tools > Port*** untuk memilih port yang digunakan. Pada Windows kemungkinan **COM1** dan **COM2** untuk port *serial* atau COM4, COM5, COM7, dan lebih tinggi untuk port *USB*. Untuk mengetahuinya, dapat dilihat perangkat *USB serial* pada seksi port dari *Windows Device Manager*.

b. Mengunggah Sketch

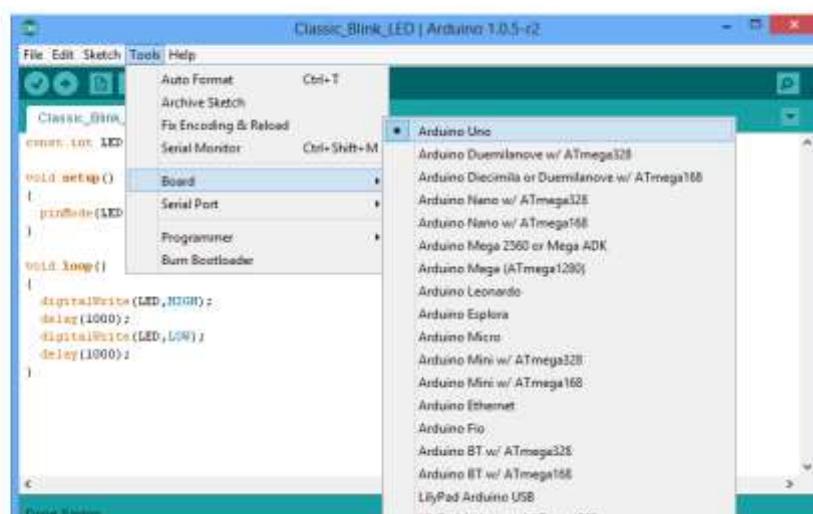
Klik tombol *Upload*, LED RX dan TX di *board* akan berkedip dan jendela Arduino *IDE* akan memberi status bar yang menunjukkan berapa banyak kemajuan yang telah diunggah. Jika unggahan berhasil, pesan “*Done uploading*”, akan muncul di jendela pesan.



Sumber : <http://www.circuitstoday.com/blink-led-with-arduino>

Gambar II.28

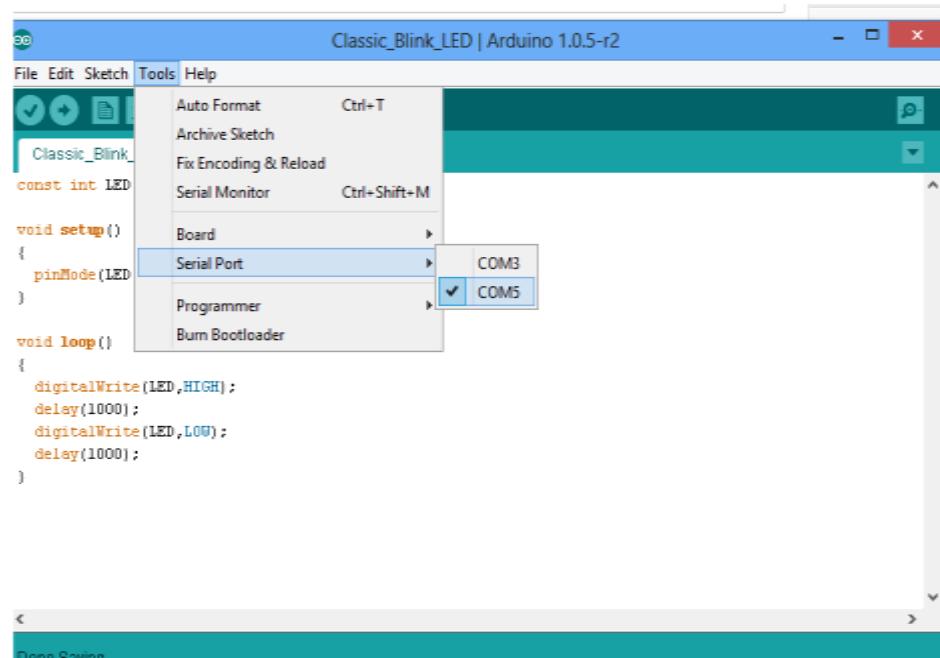
Verifikasi Sketch



Sumber : <http://www.circuitstoday.com/blink-led-with-arduino>

Gambar II.29

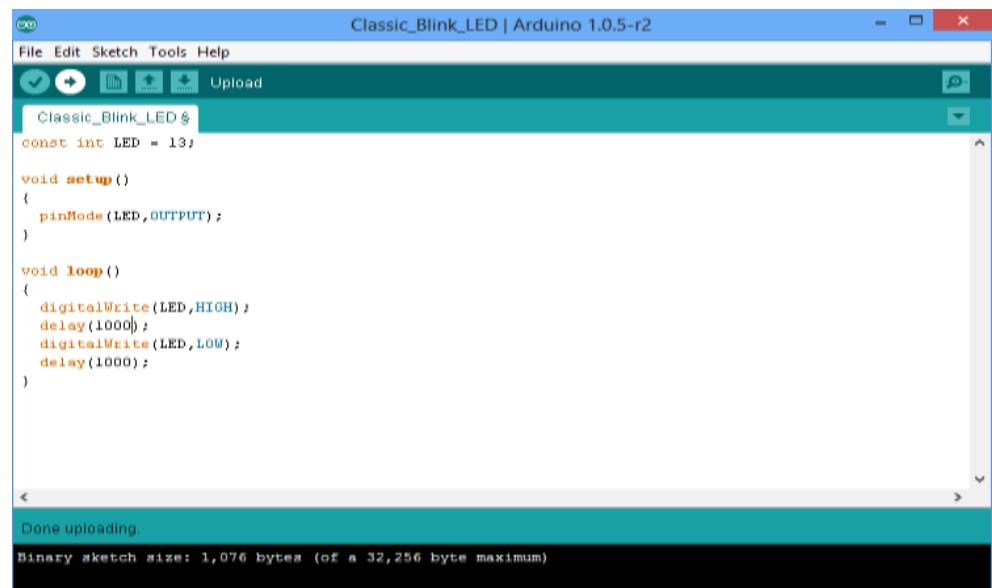
Memilih Board



Sumber : <http://www.circuitstoday.com/blink-led-with-arduino>

Gambar II.30

Memilih Port



Sumber : <http://www.circuitstoday.com/blink-led-with-arduino>

Gambar II.31

Sukses Mengunggah Sketch

BAB III

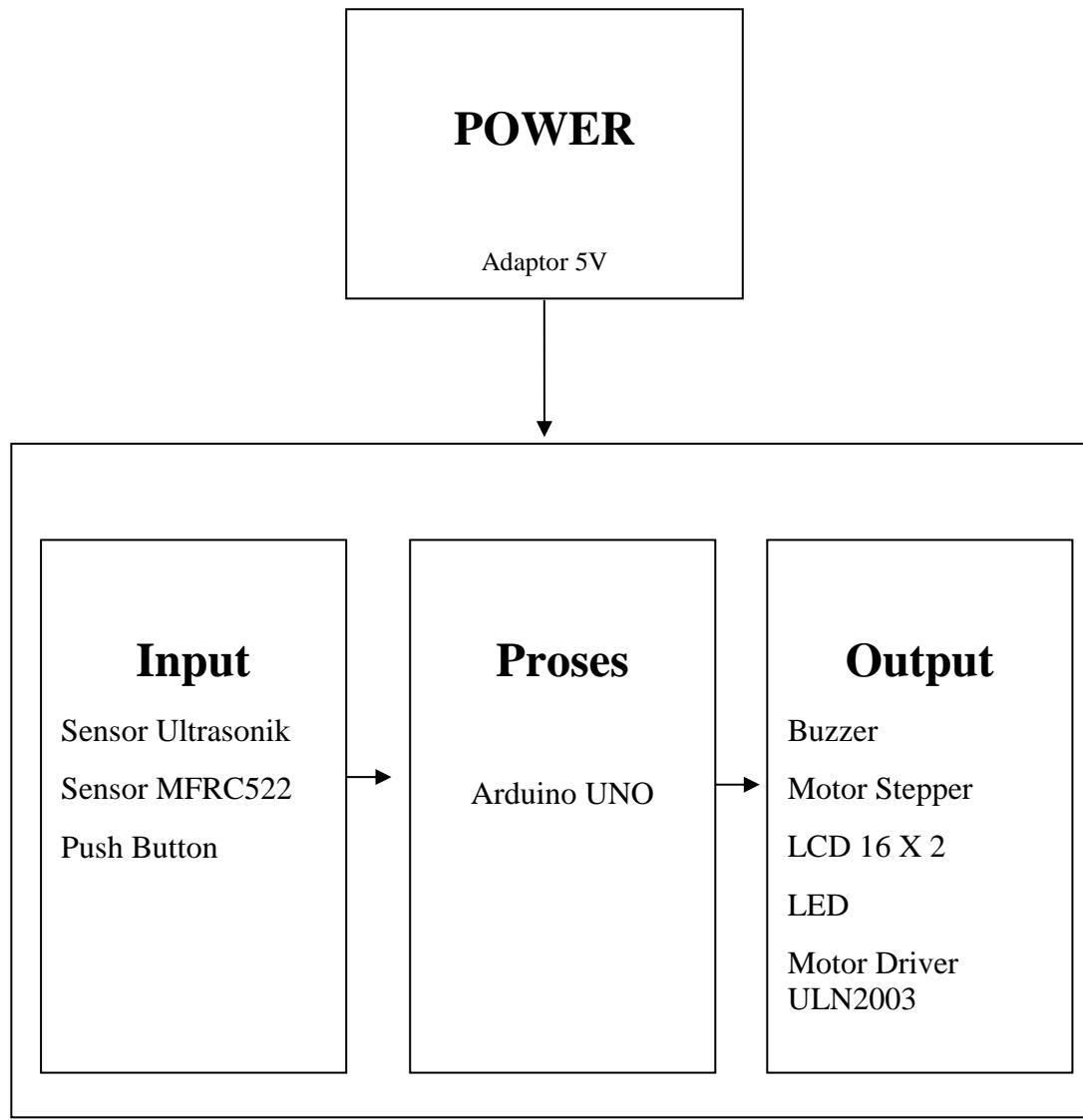
PEMBAHASAN

3.1. Tinjauan Umum Alat

Pada dasarnya, membuat suatu pekerjaan kita menjadi ringan atau otomatis itu sangatlah membantu. Oleh karena itu, dalam project kali ini kami akan membuat alat yang nanti akan membuat ringan pekerjaan kita, yaitu project membuat Pagar otomatis yang proses penggeraknya atau *output* digerakkan oleh *motor dc/stepper* dan sebagai keamanannya kami menggunakan scan yang berbasis sinyal radio, di sini kami menggunakan *Radio Frequency identifications (RFID)*. Dengan demikian, ketika akan membuka atau menutup pagar, kita tidak perlu mendorong atau menarik, cukup memindai kartu dan ketika akses nya di terima pagar tersebut akan membuka atau menutup secara otomatis.

Proses utama dalam perancangan pagar otomatis ini terdapat pada arduino UNO. Arduino UNO mengatur keseluruhan kerja rangkaian termasuk *input* dan *output*. Untuk *Input* kami menggunakan sensor Ultrasonik, *rfid mfrc522, push button*. Lalu untuk *output* kami menggunakan motor stepper dan LCD 16x2. Pada rancangan ini Motor Driver ULN2003 berfungsi sebagai penggerak Motor Stepper untuk pembuka atau penutup pagar.

3.2. Blok Rangkaian Alat



Gambar III.1.

Blok Rangkaian Alat

Penjelasan blok diagram alat sebagai berikut:

1. *Input*

Komponen *input* ini merupakan komponen masukan yang akan diproses.

Komponen *input* ini terdiri dari:

- a. Catu daya digunakan untuk pemberi *supply* tegangan ke alat yaitu tegangan sebesar 5-12 volt.
- b. Sensor rfid mfrc522 berfungsi untuk memindai kartu untuk memberi akses yang nantinya akan menggerakkan pagar membuka atau menutup.
- c. Sensor HC-SR 04 berfungsi untuk mendeteksi adanya objek yang ada di depan sensor.
- d. Push Button Sebagai tombol reset, yang dimana program nya sama seperti sensor rfid, namun penempatannya berada di dalam pagar.

2. Proses

Proses merupakan komponen utama yang berfungsi sebagai pengelola data yang diterima dari masukan yang kemudian akan menghasilkan *output*. Dalam proses ini penulis menggunakan mikrokontroler Arduino Uno.

3. *Output*

Output merupakan keluaran dari semua proses yang telah dijalankan.

Output yang dihasilkan yaitu:

- a. Motor Stepper

Berfungsi untuk mekanisme penggerak dari pagar otomatis.

- b. Buzzer berfungsi sebagai notifikasi hasil *input* yang menghasilkan bunyi suara seperti alarm.

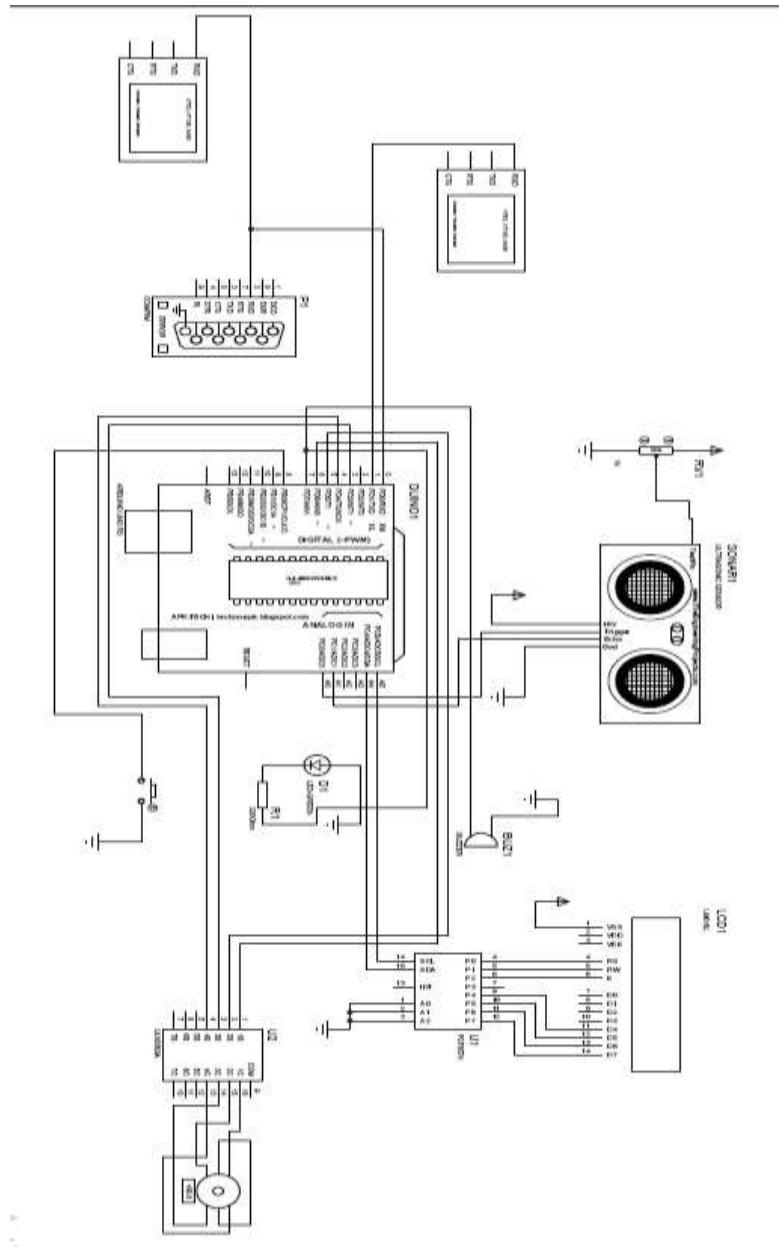
c. LCD

berfungsi untuk menampilkan *text* hasil *input* dari Sensor rfid mfrc522.

d. LED (*Light Emitting Diode*)

Berfungsi sebagai notifikasi untuk proses alat.

3.3. Skema Rangkain Alat



Gambar III.2

Skema Rangkaian Alat

Tabel III.1.
Posisi Pin Arduino dan Modul

Modul Pin	Arduino UNO Pin
RFID MFRC522	
3,3	3,3v pin
RST	Pin 9
GND	GND
IRQ	-
MISO	Pin 12
MOSI	Pin 11
SCK	Pin 13
SDA	Pin 10
Sensor HC-SR04	
Vcc	5v pin
Gnd	Gnd pin
TRIG	Pin A0
ECHO	Pin A1
Push Button	
Positif	GND
Negatif	Pin 8

LCD 16x2	
Vcc	5v pin
Gnd	Gnd pin
SCL	Pin A5
SDA	Pin A4
Motor Penggerak Pagar	
Positif	5V
Negatif	GND
IN1	Pin 6
IN2	Pin 5
IN3	Pin 4
IN4	Pin 3
LED Dan Buzzer	
Positif	Pin 7
Negatif	GND

Pagar otomatis yang dibuat ini menerima perintah dari Mikrokontroller Arduino Atmega328P sebagai pusat pemrosesan data, sensor rfid mfrc522 sebagai pemindai kartu untuk pemeberi akses buka tutup pagar, sensor ultrasonik sebagai pendekksi objek masuk yang melewati pagar dan push button sebagai perintah untuk membuka atau menutup pagar yang posisi penempatannya berada di dalam. Serta komponen elektronika lain sebagai pendukung sistem. Untuk pengaktifan sistem, hubungkan sistem dengan *power supply* atau catu daya 5-12V DC, jika indikator *LED* pada Arduino menyala maka alat tersebut mulai siap dioperasikan. Namun jika *LED* pada Arduino mati, periksalah kembali power atau catu daya. Untuk mensimulasikan rangkain pagar otomatis ini, aktifkan dengan cara memindai kartu rfid mfrc522, jika kartu tersebut sudah terverifikasi maka motor stepper akan berjalan sesuai program yang telah dimasukkan di mikrokontroller Arduino. Sensor Ultrasonik HC-SR04

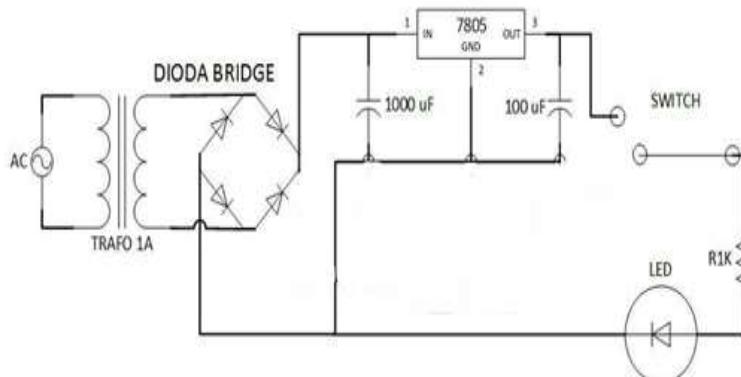
sebagai pendeksi objek yang masuk melewati pagar. Dan Push button sebagai pembuka atau penutup pagar yang diletakkan di dalam pagar. Dari sensor yang telah disebutkan tadi, *lcd* akan menampilkan *text* dari proses pemindaian dari sensor rfid mfrc522.

3.4. Cara Kerja Alat

Cara kerja alat dari tiap-tiap blok pada rangkaian alat adalah sebagai berikut:

1. Catu Daya

Catu Daya yang digunakan untuk pada rangkaian alat pagar otomatis di sini adalah Adaptor.

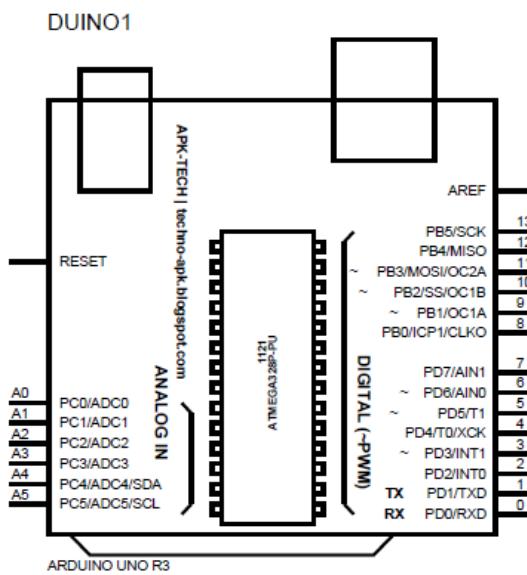


Gambar III.3

Skema Rangkaian Adaptor

2. Arduino Uno

Arduino Uno adalah salah satu produk berlabel arduino yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler Atmega328P, yang memiliki 14 pin I/O digital dan 6 pin *input* analog dan bersifat *open source*.



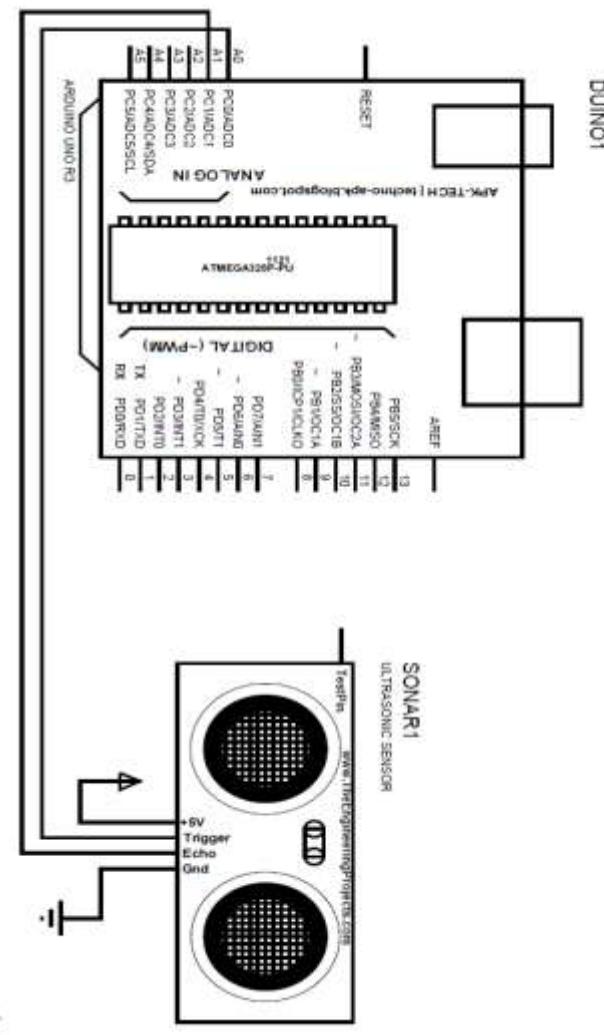
Gambar III. 4

Arduino Pada Rangkaian

Pada rangkaian pagar otomatis ini, pin keluaran Arduino yang digunakan adalah pin Analog *Output* dan Digital *Output*. Pin Digital pada Arduino dihubungkan dengan masing masing module yang kami pakai dalam project ini. Diantaranya adalah rfid mfrc522 pin 3,3v akan dihubungkan dengan port power Arduino 3,3V, pin RST rfid akan dihubungkan dengan port Digital 9 pada Arduino, GND dengan GND, IRQ rfid tdk dihubungkan, MISO dengan pin digital 12 arduino, MOSI dengan pin digital 11 arduino, SCK dengan pin digital 13, SDA dengan pin D10. Sensor ultrasonic HC-SR04 pin TRIG dihubungkan dengan port A0, ECHO dengan port A1, VCC dengan port power 5v, GND dengan GND. LCD i2C pin VCC dengan port power 5v, pin SDA dengan port A4, SCL dengan port A5. Motor Stepper pin IN1-4 masing masing dihubungkan dengan port D6,D5,D4,D3 pin power ke 5v dan GND ke GND. Push Button dihubungkan dengan port D8 dan LED dan Buzzer dihubungkan dengan pin D7.

3. Sensor ultrasonic HC-SR04

Sensor ultrasonik digunakan sebagai pendekripsi objek pada saat objek melewati pagar. Misalnya pada pagar mulai menutup, lalu ada objek melewati pagar, maka sensor akan mendekripsi nya dan akan mengirim perintah agar penngerak tidak menutup. Ketika objek sudah melewati pagar, maka penggerak akan segera menutup pagar.

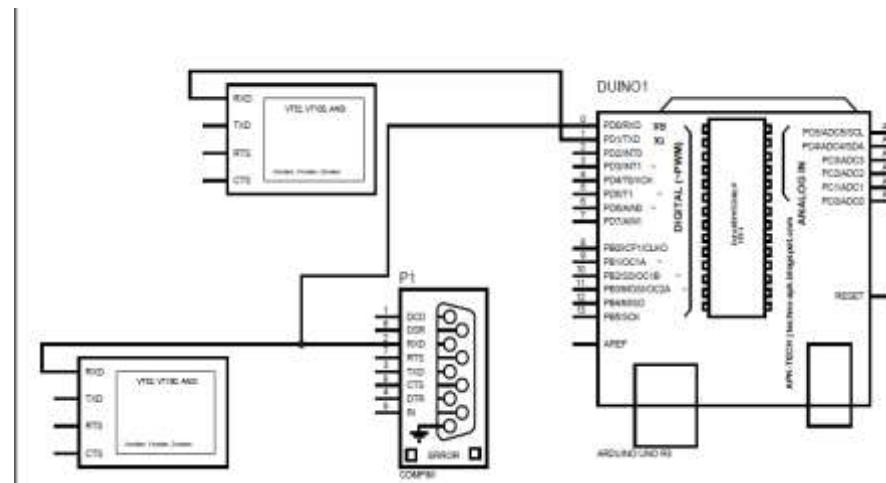


Gambar III.5

Skema Rangkain Sensor Ultrasonik HC-SR04

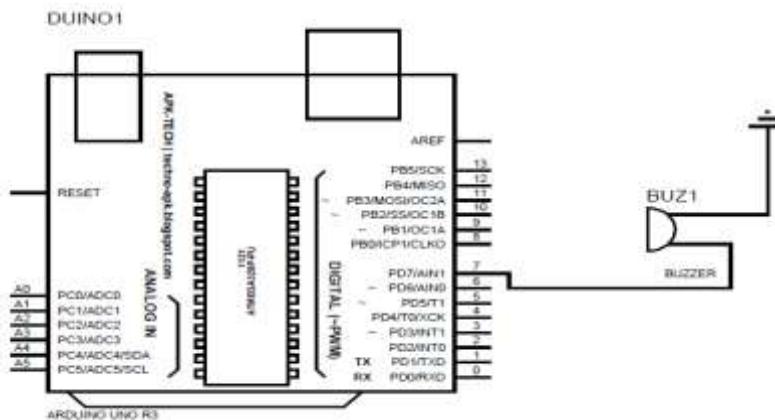
4. RFID MFRC522

Merupakan sensor utama dalam projek ini yang digunakan untuk memindai kartu untuk bisa mengakses dan untuk memberikan perintah ke mikrokontroller Arduino UNO. Jika kartu telah mendapat akses masuk oleh rfid, maka selanjutnya mikrokontroller Arduino akan mengirim perintah ke penggerak motor stepper sebagai pembuka atau penutup pagar secara otomatis.



6. Buzzer

Buzzer adalah komponen elektronika yang berfungsi mengeluarkan suara, prinsip kerja buzzer yaitu merubah listrik menjadi getaran suara. Buzzer biasa digunakan sebagai indikator (*alarm*) bahwa proses sedang bekerja atau proses sudah selesai pada sebuah alat.

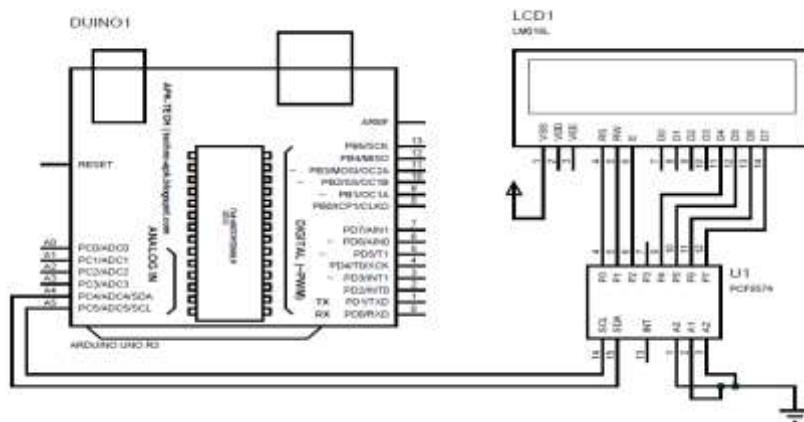


Gambar III. 8

Skema Rangkain Buzzer

7. LCD (*Liquid Crystal Display*)

LCD berfungsi sebagai penampil text dari program yang di kirim dari mikrokontroller Arduino.

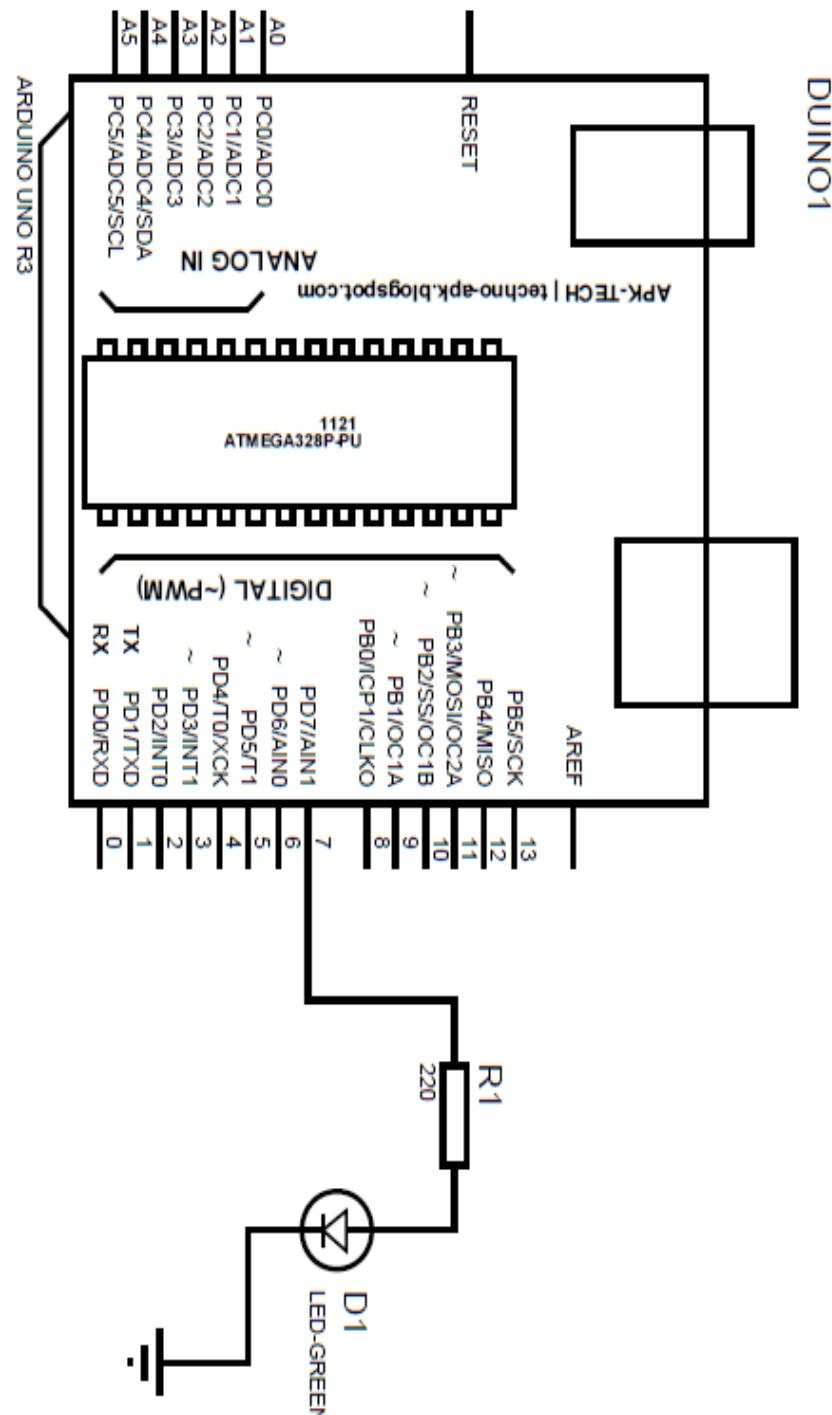


Gambar III. 9

Skema Rangkain LCD

8. LED (Light Emitting Diode)

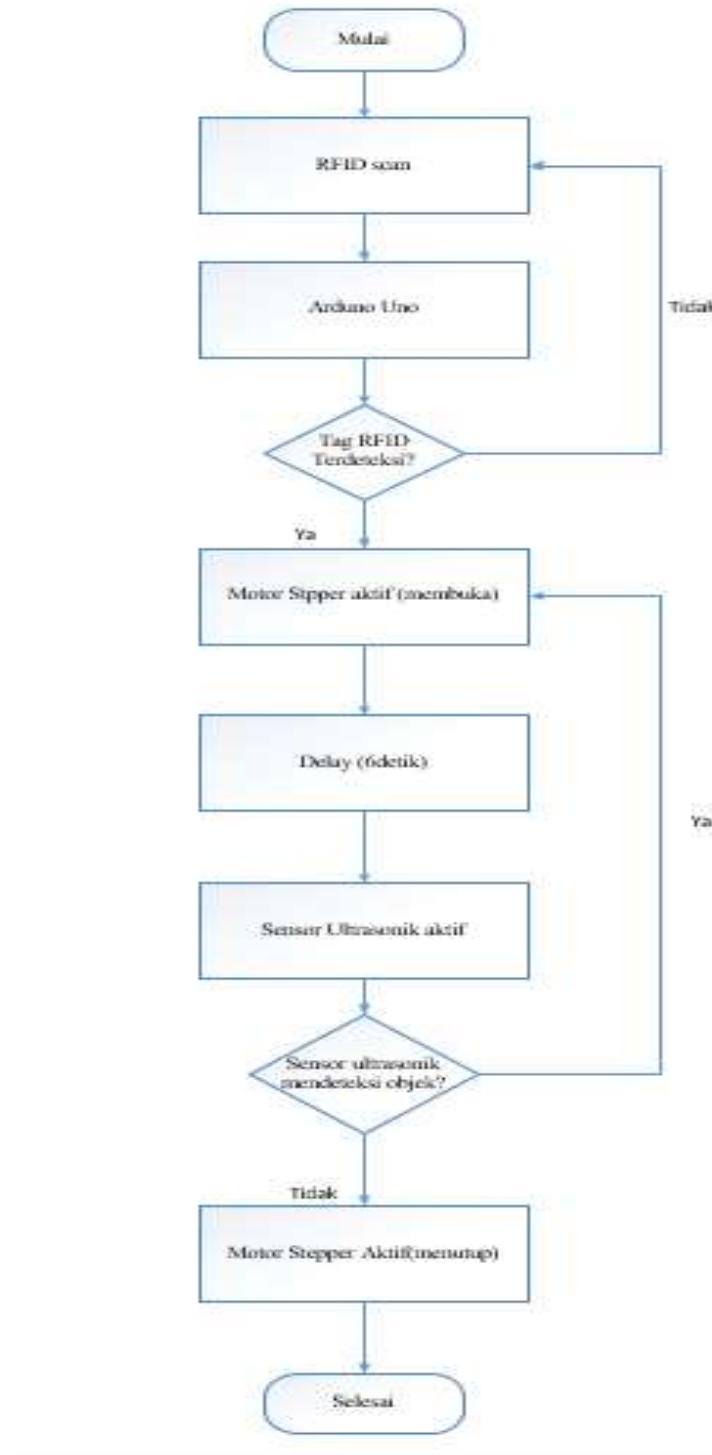
LED digunakan sebagai indikator Proses pada saat alat bekerja.



Gambar III. 10

Skema rangkain LED

3.5. Flowchart Program



Gambar III. 11

Flowchart Program

Pada flowchart diatas proses diawali dengan Mulai atau menyalakan alat, Saat alat aktif, akan melakukan pemrosesan data, jika rfid men scan. Ketika akses dari kartu mendapat kan ijin maka proses akan menggerakkan motor stepper untuk membuka pagar dan menutup pagar. Ketika pagar akan menutup dan ada objek yang berhenti saat melewati pagar, maka sensor HC-SR04 akan mendeteksi objek tersebut dan memberikan perintah ke motor stepper untuk tidak menutup pagar. Saat objek sudah melewati pagar, dan sensor HC-SR04 sudah tidak mendeteksi objek, maka pagar segera akan menutup karena sudah tidak ada penghalang lagi.

3.6. Konstruksi Sistem (Coding)

Konstruksi sistem yang akan dijelaskan pada bab ini meliputi: initialisasi program, Pembacaan *input*, *main program* dan pengendalian *output* program.

3.6.1. Intialisasi

Initialisasi adalah sebuah proses pengisian nilai awal (nilai *default*) kedalam sebuah variabel. Berikut adalah initialisasi yang digunakan pada coding pagar otomatis:

```
#define SS_PIN 10 // pin D10 pembaca kartu pada rfid
#define RST_PIN 9 /// pin D9 ( reset) pembaca kartu pada rfid
#define LED_ACCESS_PIN 7 // led yang dihubungkan pada port D7 arduino
#define motorSteps 64 // putaran motor stepper
#define motorPin1 6 // pin 1 motor stepper di hubungkan ke port D6 arduino
#define motorPin2 5 // pin 1 motor stepper di hubungkan ke port D5 arduino
#define motorPin3 4 // pin 1 motor stepper di hubungkan ke port D4 arduino
#define motorPin4 3 // pin 1 motor stepper di hubungkan ke port D3 arduino
#define button 8 // push button di hubungkan ke port Arduino D8
```

```
const int step_ = 1600; // kecepatan motor stepper
int trigPin = A0; // pin sensor HC-SR04 ke port A0
int echoPin = A1; // pin sensor HC-SR04 ke port A0
int button1 = 0; // variable push button
```

3.6.2. *Input*

Input adalah fungsi pustaka yang digunakan untuk meng atau membaca data. Codingan *input* yang digunakan pada program ini adalah:

```
void setup() // fungsi ini hanya dijalankan sekali ketika proses dijalankan
{
    myStepper.setSpeed(100); // akselerasi dari motor stepper
    Serial.begin(9600); // komunikasi serial
    lcd.begin(); // lcd 16x2 // ukuran lcd 16 kolom dan 2 baris
    lcd.setCursor(1,0); // posisi dari text di lcd
    lcd.clear(); // menghapus teks
    SPI.begin(); // Initiate SPI bus
    mfrc522.PCD_Init(); // Initialisasi MFRC522
    Serial.println("PLEASE TAG YOUR E-KTP CARD to the READER TAG...");

    // tampilan teks pada seial monitor di Arduino IDE
    Serial.println();
    pinMode(LED_ACCESS_PIN, OUTPUT); // led sebagai output
    pinMode(echoPin, INPUT); // pin echo sensor HC-SR04 sebagai input
    pinMode(trigPin, OUTPUT); // pin trig sensor HC-SR04 sebagai input
    pinMode(button, INPUT_PULLUP); // push button sebagai input
}
```

3.6.3. Main Program

```

// Look for new cards

if ( ! mfrc522.PICC_IsNewCardPresent()) // mulai pengenalan kartu akses baru

{
    return;
}

if ( ! mfrc522.PICC_ReadCardSerial() ) // memilih kartu yang akan digunakan sebagai akses

{
    return;
}

//Show UID on serial monitor

Serial.print("UID tag :"); // menampilkan nomor UID pada tag kartu yang sudah diberi akses

String content= "";

byte letter;

for (byte i = 0; i < mfrc522.uid.size; i++)

{
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");

    Serial.print(mfrc522.uid.uidByte[i], HEX);

    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));

    content.concat(String(mfrc522.uid.uidByte[i], HEX));
}

Serial.println();

Serial.print("Message : ");

```

```
content.toUpperCase();

if (content.substring(1) == "04 28 5D 4A F9 2C 80") // jika kartu mendapat akses,
maka pagar akan membuka

{
    Serial.println("Authorized access");
    Serial.println();
    lcd.setCursor(1, 0);
    lcd.println("ACCESS GRANTED");
    lcd.println();
    lcd.clear();
    lcd.setBacklight(HIGH);
    lcd.setCursor(4, 0);
    lcd.print("OPENING");
    lcd.setCursor(4, 1);
    lcd.print("GARAGE");
    Serial.println("BUKA ");
    delay(3000);
    digitalWrite(7, HIGH);
    myStepper.step(step_);
    lcd.clear();
    lcd.setBacklight(HIGH);
    lcd.setCursor(2, 0);
    lcd.print("PASSED");
    lcd.setCursor(0, 1);
    lcd.print("Welcome to Home");
```

```
Serial.println("PASSED");

digitalWrite(7, LOW);

delay(6000);

distance1();

delay(6000);

lcd.clear();

lcd.setBacklight(HIGH);

lcd.setCursor(4, 0);

lcd.print("CLOSING");

lcd.setCursor(2, 1);

lcd.print("See You ^_^");

Serial.println("CLOSING");

digitalWrite(7, HIGH);

delay(3000);

myStepper.step(-step_);

Serial.println("TUTUP");

digitalWrite(7, LOW);

delay(3000);

lcd.clear();

myStepper.step(-step_);

Serial.println("TUTUP");

digitalWrite(7, LOW);

delay(3000);

lcd.clear();

}
```

Else // jika tidak, maka akses kartu ditolak

```
{  
    Serial.println("ACCESS DENIED");  
    digitalWrite(7, HIGH);  
    delay(100);  
    digitalWrite(7, LOW);  
    delay(100);  
    digitalWrite(7, HIGH);  
    delay(100);  
    digitalWrite(7, LOW);  
    delay(100);  
    digitalWrite(7, HIGH);  
    delay(100);  
    digitalWrite(7, LOW);  
    delay(100);  
    Serial.println(" Access denied");  
    Serial.println();  
    lcd.clear();  
    lcd.setBacklight(HIGH);  
    lcd.setCursor(1, 0);  
    lcd.print("Access Denied");  
    delay(3000);  
    lcd.clear();  
}
```

3.6.4. *Output*

Output adalah proses untuk mengoutput atau menampilkan data program.

Berikut codingan *output* yang digunakan pada alat:

```
if (content.substring(1) == "04 28 5D 4A F9 2C 80") // jika kartu mendapat akses,  
maka pagar akan membuka  
{  
    Serial.println("Authorized access"); // tampilan teks di serial monitor  
    Serial.println();  
    lcd.setCursor(1, 0); // letak posisi teks dari lcd  
    lcd.println("ACCESS GRANTED"); // teks di lcd ketika akses diijinkan  
    lcd.println();  
    lcd.clear();  
    lcd.setBacklight(HIGH);  
    lcd.setCursor(4, 0);  
    lcd.print("OPENING"); //  
    lcd.setCursor(4, 1);  
    lcd.print("GARAGE");  
    Serial.println("BUKA");  
    delay(3000);  
    digitalWrite(7, HIGH);  
    myStepper.step(step_);  
    lcd.clear();  
    lcd.setBacklight(HIGH);  
    lcd.setCursor(2, 0);  
    lcd.print("PASSED");
```

```
lcd.setCursor(0, 1);

lcd.print("Welcome to Home");

Serial.println("PASSED");

digitalWrite(7, LOW);

delay(6000); // delay waktu saat pintu menutup

distance1(); // variable sensor ultrasonik

delay(6000);

lcd.clear();

lcd.setBacklight(HIGH);

lcd.setCursor(4, 0);

lcd.print("CLOSING");

lcd.setCursor(2, 1);

lcd.print("See You ^_^");

Serial.println("CLOSING");

digitalWrite(7, HIGH);

delay(3000);

myStepper.step(-step_); //posisi motor stepper menutup

Serial.println("TUTUP");

digitalWrite(7, LOW);

delay(3000);

lcd.clear();

myStepper.step(-step_);

Serial.println("TUTUP");

digitalWrite(7, LOW);

delay(3000);
```

```
lcd.clear();  
}  
  
Else // jika tidak, maka akses kartu ditolak  
{  
    Serial.println("ACCESS DENIED");  
    digitalWrite(7, HIGH);  
    delay(100);  
    digitalWrite(7, LOW);  
    delay(100);  
    digitalWrite(7, HIGH);  
    delay(100);  
    digitalWrite(7, LOW);  
    delay(100);  
    digitalWrite(7, HIGH);  
    delay(100);  
    digitalWrite(7, LOW);  
    delay(100);  
    Serial.println(" Access denied");  
    Serial.println();  
    lcd.clear();  
    lcd.setBacklight(HIGH);  
    lcd.setCursor(1, 0);  
    lcd.print("Access Denied");  
    delay(3000);  
    lcd.clear(); }
```

/ keterangan dari setiap symbol, sintaks, fungsi, variabel, pendefinisian variabel atau hardware.*

Define= yaitu mendefinisikan hardware maupun variabel

Include = penyertaan library yang akan kita gunakan dalam program,

Void setup = fungsi ini dijalankan hanya sekali ketika program mulai dijalankan

Void loop = fungsi yang berjalan terus menerus ketika program dijalankan

Void distance, void button = adalah fungsi tambahan yang kita buat sendiri. Fungsi ini bias dipanggil kalau akan digunakan, pembuatan fungsi tambahan dilakukan biasanya untuk mempermudah dalam penulisan program, serta penulisan akan lebih rapi dan mudah saat menganalisa ketika terjadi error.

Simbol “//” digunakan sebagai komentar satu baris. Simbol ini tidak akan di baca pada saat proses kompilasi.

Simbol “/* dan */” berfungsi sebagai kolom komentar juga, namun kita bias memberikan keterangan lebih dari satu baris (komentar banyak). */

3.7. Hasil Percobaan

Hasil yang dilakukan pada pagar otomatis ini adalah dengan menguji coba apakah sensor dapat melakukan pembacaan dari program atau tidak, serta *output* akan bekerja dengan baik atau tidak jika sensor dalam keadaan aktif.

3.7.1. Hasil Input

Tabel III. 2
Hasil Percobaan *input* Sensor RFID MFRC522

RFID MFRC522			
No	Percobaan	Kartu Akses	Kartu Non Akses
1	Saat Alat Aktif	Dijinkan. Motor Stepper aktif dan pagar akan segera membuka atau menutup secara otomatis	Ditolak. Motor Stepper tidak melakukan apapun.

Sesuai percobaan di atas, saat alat aktif mulai melakukan pemindaian kartu. Jika kartu telah menerima akses, maka proses akan berlanjut yaitu menggerakkan motor stepper untuk membuka dan menutup pagar secara otomatis, sebaliknya, jika kartu yang digunakan tidak menerima akses, maka tidak ada aktifitas dari motor stepper. Dalam proses tersebut, ada waktu(delay) saat pagar membuka atau menutup

Tabel III. 3

Hasil Percobaan *Input* Sensor HC-SR04

Sensor HC-SR04			
No	Percobaan	Jarak	Status
1	Saat Alat Aktif	<10cm	Sensor bekerja
2	Saat Alat Aktif	>10cm	Sensor tidak bekerja

Pada Percobaan di atas, sensor HC-SR04 akan mendeteksi objek yang melewati pagar. Jika objek yang melewati pagar saat pagar menutup, maka sensor akan bekerja mendeteksi objek tersebut dan akan memberi perintah ke motor stepper agar tidak menggerakkan pagar saat menutup. Jadi sensor ini bekerja saat objek melewati pagar yang sedang menutup. Hal ini di program seperti ini semisal ada objek yang melewati pagar saat menutup akan mencegah terjadinya pagar menabrak objek.

Tabel III. 4

Hasil Percobaan Push Button

Push Button			
No	Percobaan	Kondisi	Hasil
1	Saat alat aktif	ditekan	Alat bekerja
2	Saat alat aktif	Tidak ditekan	Tidak ada hasil

Pada percobaan ini, ketika alat aktif dan push button di tekan, maka secara keseluruhan rangkain alat akan bekerja. Penempatan push button ini berada di dalam pagar. Digunakan pada saat kita mau keluar, karena saat kita mau masuk akses masuk menggunakan rfid.

3.7.2. Hasil *Output*

Tabel III. 5

Hasil Percobaan *Output* RFID MFRC522

Percobaan	Akses	Serial monitor	Motor Stepper	Buzzer dan LED	Tampilan LCD
1	Dijinkan	Authorized access	Aktif	Berbunyi dan Menyala	ACCESS GRANTED
2	Ditolak	-	Tidak aktif	Berbunyi dan menyala	ACCESS DENIED

Pada tabel III. 5 Menunjukkan bahwa pada percobaan pertama atau setelah alat aktif, kartu yang mendapat akses akan meneruskan perintah ke motor stepper dan akan menggerakkan pagar membuka dan menutup. Diikuti dengan bunyi buzzer dan nyala dari LED. LCD akan menampilkan Teks “ACCESS GRANTED”. Di percobaan kedua, menggunakan kartu yang tidak mendapatkan akses, tentunya tidak ada lanjutan perintah, karena dari awal kartu tersebut tidak mendapatkan akses. Buzzer dan LED tetap berbunyi dan menyala, namun motor stepper tidak bekerja.

Tabel III. 6

Hasil Percobaan Output Sensor HC-SR04

Percobaan	Jarak	Serial Monitor	Motor Stepper	Buzzer dan LED	Tampilan LCD
1.	<10cm	TRUE	Tidak Aktif	Berbunyi dan menyala	ADA OBJEK
2.	>10cm	-	Aktif	Berbunyi dan Menyalala	Closing

Pada tabel III. 6 Telah dilakukan percobaan pertama atau saat alat aktif. Ketika sensor mendeteksi objek yang dimana jarak yang diatur <10cm, maka sensor akan bekerja dan menahan motor stepper agar tidak menggerakkan pagar pada posisi menutup. Di percobaan kedua, yang dimana sensor tidak dapat mendeteksi objek yang melewati pagar karena jarak jangkaunya >10cm. Secara otomatis motor stepper bergerak secara normal(menutup) sesuai program.

3.7.3. Hasil Keseluruhan Alat

Tabel III. 7
Hasil Percobaan Keseluruhan Alat

No	RFID MFRC522	Sensor HC-SR04	Push Button	Motor Stepper	Buzzer dan LED	Tampilan LCD	Tampilan Serial Monitor
1	Akses Dijinkan	Aktif	Aktif	Berjalan	Berbunyi dan Menyala	ACCESS GRANTED	BUKA
2	Akses ditolak	-	-	-	Berbunyi dan Menyala	ACCESS DENIED	Access Denied

Pada Tabel III. 7 Menunjukkan hasil dari percobaan keseluruhan alat, saat alat pertama kali di aktifkan, maka semua modul atau sensor aktif. Saat menggunakan kartu rfid yang telah mendapat akses, proses berjalan sesuai yang kami inginkan, ketika pemindaian kartu dan akses diijinkan, LCD akan menampilkan text “ACCESS GRANTED” dan motor stepper menerima perintah untuk membuka pagar dan menutup secara otomatis. Jika waktu pagar menutup, ada objek yang masuk atau melewati pagar, sensor HC-SR04 akan mendeteksi objek tersebut, dan akan memberikan perintah ke motor stepper agar tidak menutup jika ada objek berhenti di tengah tengah pagar saat dalam keadaan pagar akan menutup LCD menampilkan text “ADA OBJEK”. Pagar tidak menutup, dan saat objek nya sudah melewati pagar tadi seketika pagar akan menutup lagi dikarenakan sudah tidak ada objek atau penghalang.

BAB IV

PENUTUP

4.1. Kesimpulan

Dari hasil Pembuatan Prototipe Pagar otomatis ini yang menggunakan *Radio Frequency Indentication* (RFID) Berbasis Mikrokontroller Atmega328P Dapat diambil beberapa kesimpulan sebagai berikut :

1. Memberikan kemudahan bagi kita untuk membuka atau menutup pagar secara otomatis tanpa harus menarik atau mendorong pagar secara manual.
2. Pagar otomatis ini membutuhkan daya sekitar 5-12V agar bias mengoperasikannya.
3. Sensor ultrasonic (HC-SR04) berfungsi sebagai pendekripsi objek yang berhenti atau melewati pagar saat posisi pagar sedang menutup. Agar pagar tidak langsung menutup kalau ada objek yang terdeteksi. Hal ini mencegah agar tidak terjadinya tabrakan ketika pagar sedang menutup.
4. Motor Stepper berfungsi sebagai komponen penggerak dari pagar otomatis ini.
5. Push Button sebagai akses keluar karena akses masuknya telah menggunakan RFID.

4.2. Saran

Kami sangat menyadari dalam projek ini masih banyak kekurangan nya, mengingat projek kami sebuah prototipe jadi masih banyak yang harus di kembangkan sampai bias di aplikasikan dalam kehidupan nyata. Berikut saran yang mungkin bisa kami kembangkan suatu saat nanti :

1. Output dari projek ini mungkin bisa ditambah lagi seperti penambahan sistem keamanannya seperti kunci otomatis setelah pagar menutup. Sebagai alternatif bisa di tambahkan motor servo untuk pengunci atau pengait saat pagar menutup.
2. Penambahan limit switch untuk mengatur jalannya dari motor stepper. Saat pendektsian dari sensor ultrasonik ada objek yang berhenti di tengah tengah pagar saat mau menutup, limit switch ini berfungsi menghentikan pagar yang sedang akan menutup.
3. Untuk lebih diperhatikan lagi *wiring* atau pengkabelan dari alat agar lebih rapi lagi, jadi lebih ringkas.
4. Penambahan akses E-KTP (multi akses) agar tidak bergantung dengan satu E-KTP.
5. Bisa juga mengganti buzzer (mungkin bisa diganti dengan speaker), atau mengatur suara dari buzzer itu sendiri agar tidak menimbulkan kebisingan.
6. Tata letak dari push button lebih diperhatikan lagi agar lebih memudahkan seseorang saat menggunakannaya.
7. Penambahan modul atau sensor agar membuat alat tersebut menjadi lebih efisien.

DAFTAR PUSTAKA

- Ahyadi, Z. (2018). *Belajar Antarmuka Arduino Secara Cepat Dari Contoh* (1st ed.). Yogyakarta: POLIBAN PRESS.
- Akhiruddin. (2017). Perancangan Alat Pemisah dan Pensortir Buah Jeruk Berbasis Arduino. *Journal of Electrical Technology*, 2(3), 36. Retrieved from <https://jurnal.uisu.ac.id/index.php/jet/article/view/237>
- ANANDA, R. (2018). *40 Project Robotic dan Aplikasi Android* (1st ed.). Yogyakarta: Deepublish.
- Arasada, B. dan B. S. (2017). Aplikasi Sensor Ultrasonik Untuk Deteksi Posisi Jarak Pada Ruang Menggunakan Arduino Uno Bakhtiyar Arasada. *Jurnal Teknik Elektro*, 06(02), 2.
- Dinata, I. dan W. S. (2015). IMPLEMENTASI WIRELESS MONITORING ENERGI LISTRIK BERBASIS WEB DATABASE. *Jurnal Teknik Elektro*, 4(1), 86.
- Falani, A. Z. (2015). Robot Line Follower Berbasis Mikrokontroler Atmega 16 Dengan Menampilkan Status Gerak Pada Lcd. *E-NARODROID*, 1(1). <https://doi.org/10.31090/narodroid.v1i1.6>
- Iswanto dan Nia Maharani Raharja. (2015). *Mikrokontroller: Teori dan Praktik Atmega 16 dengan Bahasa C* (1st ed.). Yogyakarta: Deepublish.
- Kadir, A. (2017). *Pemrograman Arduino dan Processing*. Jakarta: PT Elex Media Komputindo.
- Listiyorini, R. (2018). *DASAR LISTRIK DAN ELEKTRONIKA* (1st ed.). Yogyakarta: DEEPUBLISH.
- Maulana, E. dan R. A. P. (2017). Pemanfaatan Layanan SMS Telepon Seluler Berbasis Mikrokontroler Atmega328p Sebagai Sistem Kontrol Lampu Rumah. *JURNAL TEKNIK KOMPUTER AMIK BSI*, III, 93.
- Mubarok, A., Sofyan, I., Rismayadi, A. A., & Najiyah, I. (2018). *Sistem Keamanan Rumah Menggunakan RFID Sensor PIR*. 5(1), 137–144.

- Nuraini, R. (2015). DESAIN ALGORITHM OPERASI PERKALIAN MATRIKS MENGGUNAKAN METODE FLOWCHART. *JURNAL TEKNIK KOMPUTER AMIK BSI*, 1(1), 146. Retrieved from <https://ejurnal.bsi.ac.id/ejurnal/index.php/jtk/article/view/245>
- Sari, Kartika, Cucu Suhery, Y. A. (2015). Implementasi Sistem Pakan Ikan Menggunakan Buzzer Dan Aplikasi Antarmuka Berbasis Mikrokontroler. *Jurnal Coding Sistem Komputer Untan*, 3(2), 113. Retrieved from <http://jurnal.untan.ac.id/index.php/jcskommipa/article/view/10803>
- Sitohang, E. P., Mamahit, D. J., Tulung, N. S., Elektro, T., Sam, U., Manado, R., & Manado, J. K. B. (2018). Rancang Bangun Catu Daya DC Menggunakan Mikrokontroler ATmega 8535. *Jurnal Teknik Elektro Dan Komputer*, 7(2), 135–142.
- Sokop, Jendri Steven, Dringhuzen J. Mamahit, S. R. U. A. S. (2016). Trainer Periferal Antarmuka Berbasis Mikrokontroler Arduino Uno. *E-Journal Teknik Elektro Dan Komputer*, 5(3), 14.
- Suhardi, D. (2014). Prototipe Controller Lampu Penerangan LED (Light Emitting Diode) Independent Bertenaga Surya. *Jurnal GAMMA*, 10(1), 116–122.
- Sujarwata. (2018). *Belajar Mikrokontroler BS2SX Teori, Penerapan dan Contoh pemrograman PBasic* (1st ed.). Yogyakarta: Deepublish.
- Swara, G. Y., Kom, M., & Pebriadi, Y. (2016). Jurnal TEKNOIF ISSN : 2338-2724 REKAYASA PERANGKAT LUNAK PEMESANAN TIKET BIOSKOP Jurnal TEKNOIF ISSN : 2338-2724. *Jurnal TEKNOIF*, 4(2), 27–39.

DAFTAR RIWAYAT HIDUP

I. Biodata Mahasiswa

N.I.M : 13160981
Nama Lengkap : Fatku Rachman Saleh
Tempat & Tanggal Lahir : Bogor, 29 Agustus 1995
Alamat lengkap : Kp.Babakan 003/005 Ds. Tarikolot Kec.Citeureup Kab. Bogor.

II. Riwayat Pendidikan Formal & Non-Formal

a. Formal

1. SD Negeri 13 Cepu, Jawa Tengah. Lulus tahun 2007
2. SMP Negeri 1 Parang, Magetan Jawa Timur. Lulus tahun 2010
3. SMK Yosonegoro Magetan, Jawa Timur. Lulus tahun 2013

b. Tidak Formal

1. -

III. Riwayat Pengalaman Berorganisasi / Pekerjaan

1. Karyawan PT. Novell Pharmaceutical Laboratories tahun 2015 s.d tahun 2016.



Jakarta, 15 Juli 2019

Fatku Rachman Saleh

DAFTAR RIWAYAT HIDUP

I. Biodata Mahasiswa

N.I.M	:	13160852
Nama Lengkap	:	Gutomo Mulya Putra
Tempat & Tanggal Lahir	:	Jakarta, 01 Juli 1998
Alamat lengkap	:	Jl. Nusa 001/003 No. 60 Kel. Kramat Jati Jakarta Timur

II. Riwayat Pendidikan Formal & Non-Formal

a. Formal

1. SD Negeri 26 Petang Jakarta Timur . Lulus tahun 2010
2. SMP Negeri 287 Pinang Ranti Jakarta Timur . Lulus tahun 2013
3. SMK Angkasa 1 Jakarta Timur. Lulus tahun 2016

b. Tidak Formal

1. -

III. Riwayat Pengalaman Berorganisasi / Pekerjaan

1. -



Jakarta, 15 Juli 2019

Gutomo Mulya Putra

LAMPIRAN-LAMPIRAN

A1. Data Sheet

1. Data Sheet IC Atmega328P

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
 - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
 - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot
 - Program True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.75 µA (Including 32 kHz RTC)



®
**8-bit
Microcontroller
with 4/8/16/32K
Bytes In-System
Programmable
Flash**

**ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P**

Summary

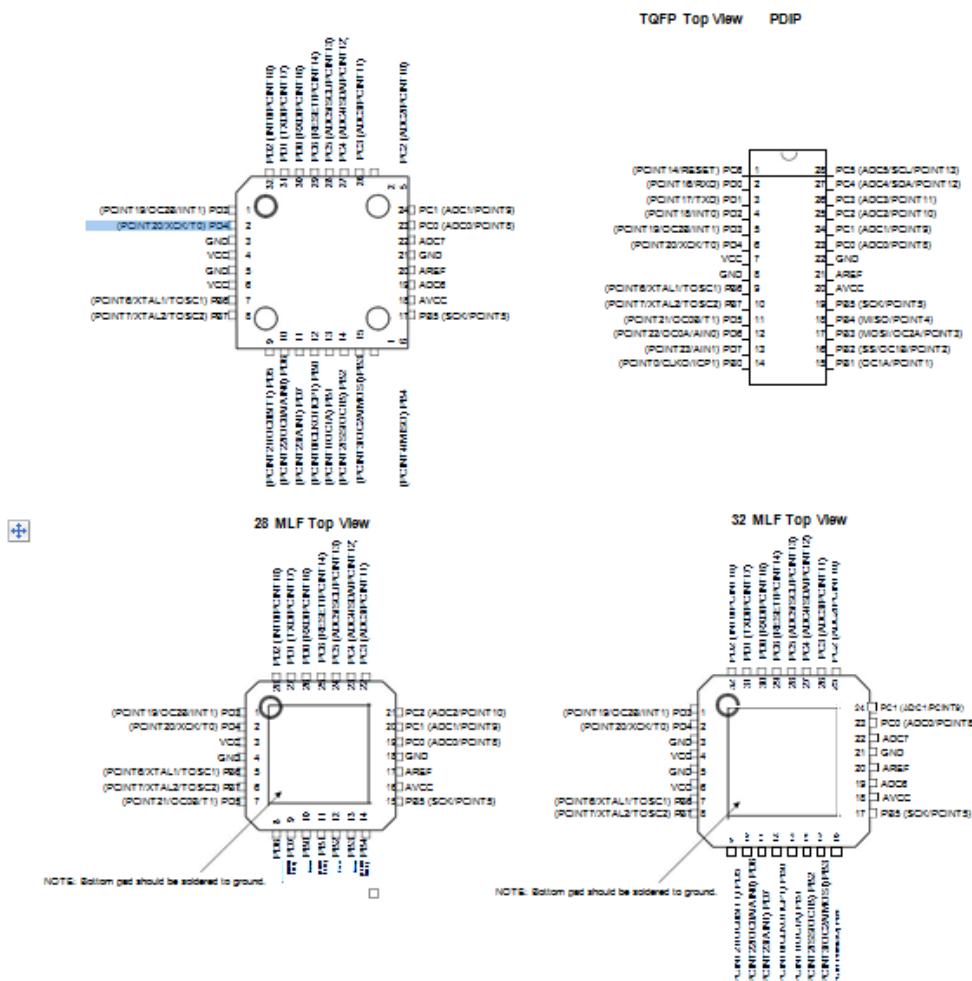
Rev. S161CB-AVR-05/09



ATmega48PA/88PA/168PA/328P

1. Pin Configurations

Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P



1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 76 and "System Clock and Clock Options" on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 28-3 on page 308. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 79.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.



ATmega48PA/88PA/168PA/328P

The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 82.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6..4 use digital supply voltage, V_{CC}.

1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

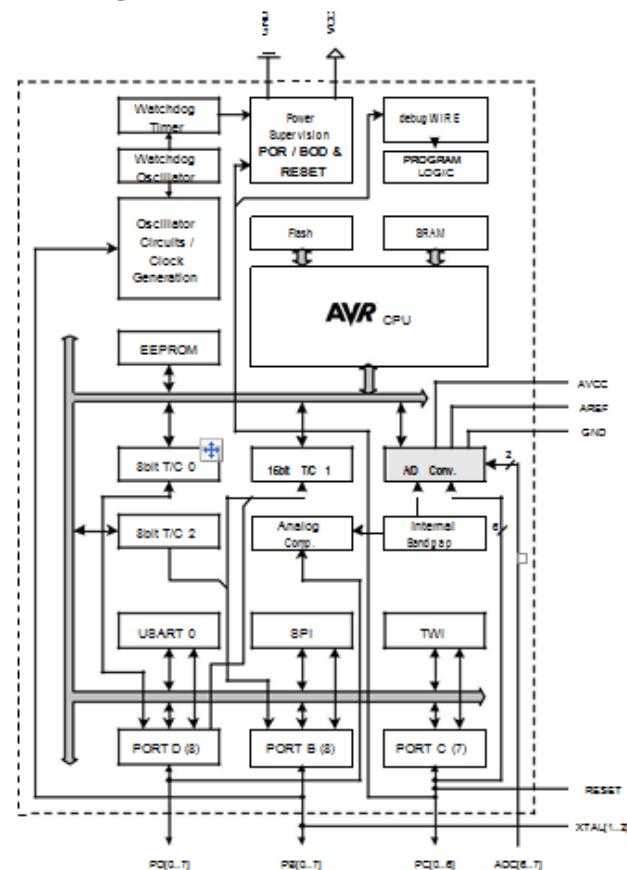
ATmega48PA/88PA/168PA/328P

2. Overview

The ATmega48PA/88PA/168PA/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48PA/88PA/168PA/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting

ATmega48PA/88PA/168PA/328P

architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega48PA/88PA/168PA/328P provides the following features: 4/8/16/32K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1K bytes EEPROM, 512/1K/1K/2K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48PA/88PA/168PA/328P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48PA/88PA/168PA/328P AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

2.2 Comparison Between ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P

The ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P differ only in memory sizes, boot loader support, and interrupt vector sizes. [Table 2-1](#) summarizes the different memory and interrupt vector sizes for the three devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48PA	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88PA	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168PA	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 instruction words/vector

ATmega88PA, ATmega168PA and ATmega328P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48PA, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.



ATmega48PA/88PA/168PA/328P

3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/svr>.

4. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

2. Data Sheet MFRC522



MFRC522

Standard 3V MIFARE reader solution

Rev. 3.8 — 17 September 2014
112138

Product data sheet
COMPANY PUBLIC

1. Introduction

This document describes the functionality and electrical specifications of the contactless reader/writer MFRC522.

Remark: The MFRC522 supports all variants of the MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 and MIFARE Plus RF identification protocols. To aid readability throughout this data sheet, the MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 and MIFARE Plus products and protocols have the generic name MIFARE.

2. General description

The MFRC522 is a highly integrated reader/writer IC for contactless communication at 13.56 MHz. The MFRC522 reader supports ISO/IEC 14443 A/MIFARE mode.

The MFRC522's internal transmitter is able to drive a reader/writer antenna designed to communicate with ISO/IEC 14443 A/MIFARE cards and transponders without additional active circuitry. The receiver module provides a robust and efficient implementation for demodulating and decoding signals from ISO/IEC 14443 A/MIFARE compatible cards and transponders. The digital module manages the complete ISO/IEC 14443 A framing and error detection (parity and CRC) functionality.

The MFRC522 supports MF1xxS20, MF1xxS70 and MF1xxS50 products. The MFRC522 supports contactless communication and uses MIFARE higher transfer speeds up to 848 kBd in both directions.

The following host interfaces are provided:

- Serial Peripheral Interface (SPI)
- Serial UART (similar to RS232 with voltage levels dependant on pin voltage supply)
- I²C-bus interface

2.1 Differences between version 1.0 and 2.0

The MFRC522 is available in two versions:

- MFRC52201HN1, hereafter referred to version 1.0 and
- MFRC52202HN1, hereafter referred to version 2.0.

The MFRC522 version 2.0 is fully compatible to version 1.0 and offers in addition the following features and improvements:

- Increased stability of the reader IC in rough conditions
- An additional timer prescaler, see [Section 8.5](#).
- A corrected CRC handling when RX Multiple is set to 1

This data sheet version covers both versions of the MFRC522 and describes the differences between the versions if applicable.

3. Features and benefits

- Highly integrated analog circuitry to demodulate and decode responses
- Buffered output drivers for connecting an antenna with the minimum number of external components
- Supports ISO/IEC 14443 A/MIFARE
- Typical operating distance in Read/Write mode up to 50 mm depending on the antenna size and tuning
- Supports MF1xxS20, MF1xxS70 and MF1xxS50 encryption in Read/Write mode
- Supports ISO/IEC 14443 A higher transfer speed communication up to 848 kBd
- Supports MFIN/MFOUT
- Additional internal power supply to the smart card IC connected via MFIN/MFOUT
- Supported host interfaces
 - ◆ SPI up to 10 Mbit/s
 - ◆ I²C-bus interface up to 400 kBd in Fast mode, up to 3400 kBd in High-speed mode
 - ◆ RS232 Serial UART up to 1228.8 kBd, with voltage levels dependant on pin voltage supply
- FIFO buffer handles 64 byte send and receive
- Flexible interrupt modes
- Hard reset with low power function
- Power-down by software mode
- Programmable timer
- Internal oscillator for connection to 27.12 MHz quartz crystal
- 2.5 V to 3.3 V power supply
- CRC coprocessor
- Programmable I/O pins
- Internal self-test

4. Quick reference data

Table 1. Quick reference data

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{DDA}	analog supply voltage	$V_{DDA}(PVDD) \leq V_{DDA} = V_{DDD} = V_{DD(PVDD)}$	[1]	2.5	3.3	3.6	V
V_{DDD}	digital supply voltage	$V_{SSA} = V_{SSD} = V_{SS(PVSS)} = V_{SS(TVSS)} = 0\text{ V}$		2.5	3.3	3.6	V
$V_{DD(TVDD)}$	TVDD supply voltage			2.5	3.3	3.6	V
$V_{DD(PVDD)}$	PVDD supply voltage		[2]	1.6	1.8	3.6	V
$V_{DD(SVDD)}$	SVDD supply voltage	$V_{SSA} = V_{SSD} = V_{SS(PVSS)} = V_{SS(TVSS)} = 0\text{ V}$		1.6	-	3.6	V

MFRC522
Product data sheet
Rev. 3.0 — 17 September 2014
COMPANY PUBLIC
112138
©2014 NXP Semiconductors N.V. All rights reserved.
2 of 95

NXP Semiconductors

MFRC522

Standard 3V MIFARE reader solution

Table 1. Quick reference data – continued

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
I_{sd}	power-down current	$V_{DDA} = V_{DDD} = V_{DD(TVDD)} = V_{DD(PVDD)} = 3\text{ V}$					
		hard power-down; pin NRSTPD set LOW	[3]	-	-	5	μA
		soft power-down; RF level detector on	[4]	-	-	10	μA
I_{DDO}	digital supply current	pin DVDD; $V_{DDD} = 3\text{ V}$		-	6.5	9	mA
I_{DDA}	analog supply current	pin AVDD; $V_{DDA} = 3\text{ V}$, CommandReg register's RcvOff bit = 0		-	7	10	mA
		pin AVDD; receiver switched off; $V_{DDA} = 3\text{ V}$, CommandReg register's RcvOff bit = 1		-	3	5	mA
$I_{DD(PVDD)}$	PVDD supply current	pin PVDD	[5]	-	-	40	mA
$I_{DD(TVDD)}$	TVDD supply current	pin TVDD; continuous wave	[6][7]	-	60	100	mA
T_{amb}	ambient temperature	HVQFN32		-25	-	+85	$^{\circ}\text{C}$

[1] Supply voltages below 3 V reduce the performance in, for example, the achievable operating distance.

[2] V_{DDA} , V_{DDD} and $V_{DD(TVDD)}$ must always be the same voltage.

[3] $V_{DD(PVDD)}$ must always be the same or lower voltage than V_{DDA} .

[4] I_{sd} is the total current for all supplies.

[5] $I_{DD(PVDD)}$ depends on the overall load at the digital pins.

[6] $I_{DD(TVDD)}$ depends on $V_{DD(TVDD)}$ and the external circuit connected to pins TX1 and TX2.

[7] During typical circuit operation, the overall current is below 100 mA.

[8] Typical value using a complementary driver configuration and an antenna matched to 40 Ω between pins TX1 and TX2 at 13.56 MHz.

5 Ordering information

5. Ordering information

Table 2. Ordering information

Type number	Package		
Name	Description		Version
MFR52201HN1/TRAYB ^[1]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOTB17-1
MFR52201HN1/TRAYBM ^[2]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOTB17-1
MFR52202HN1/TRAYB ^[1]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOTB17-1
MFR52202HN1/TRAYBM ^[2]	HVQFN32	plastic thermal enhanced very thin quad flat package; no leads; 32 terminal; body 5 × 5 × 0.85 mm	SOTB17-1

[1] Delivered in one tray.

[2] Delivered in five trays.

MFRC522

Product data sheet
COMPANY PUBLIC

All information provided in this document is subject to legal disclosure.

Rev. 3.8 — 17 September 2014

© NXP Semiconductors NV 2014. All rights reserved.

3 of 95

NXP Semiconductors

MFRC522

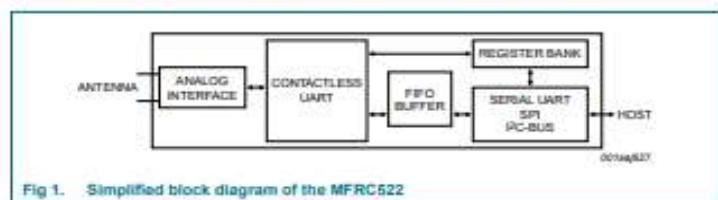
Standard 3V MIFARE reader solution

6. Block diagram

The analog interface handles the modulation and demodulation of the analog signals.

The contactless UART manages the protocol requirements for the communication protocols in cooperation with the host. The FIFO buffer ensures fast and convenient data transfer to and from the host and the contactless UART and vice versa.

Various host interfaces are implemented to meet different customer requirements.



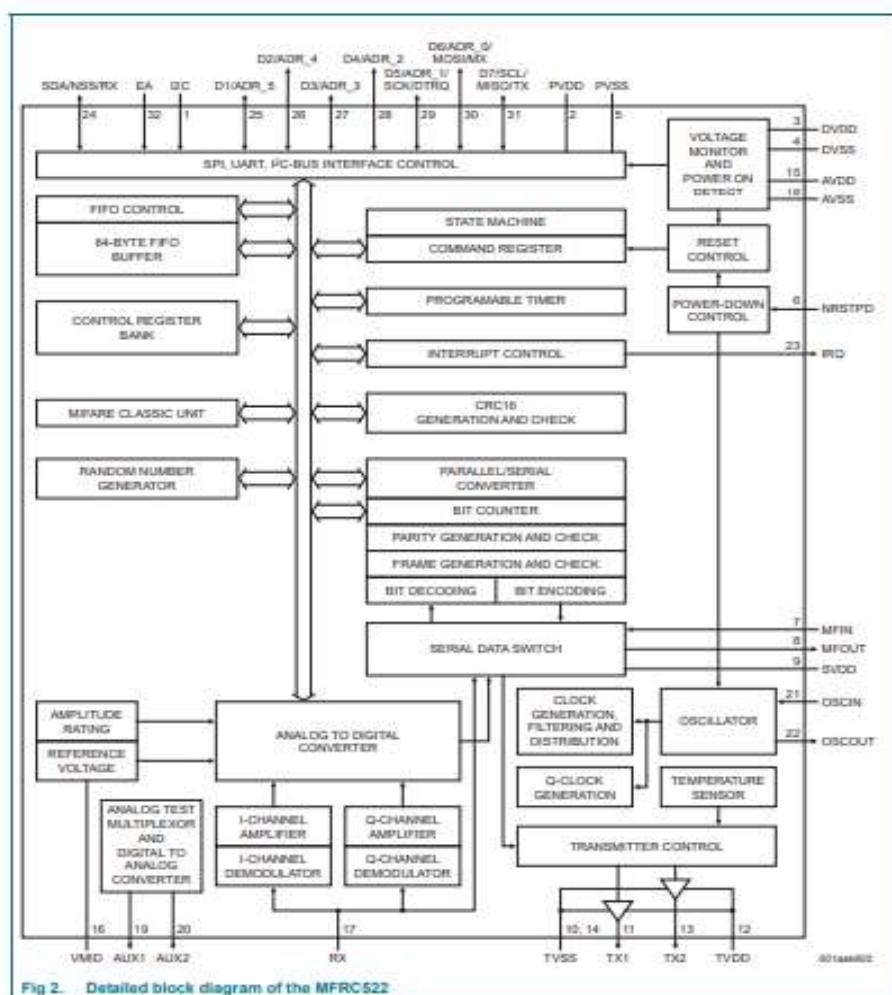


Fig 2. Detailed block diagram of the MFRC522.

7. Pinning information

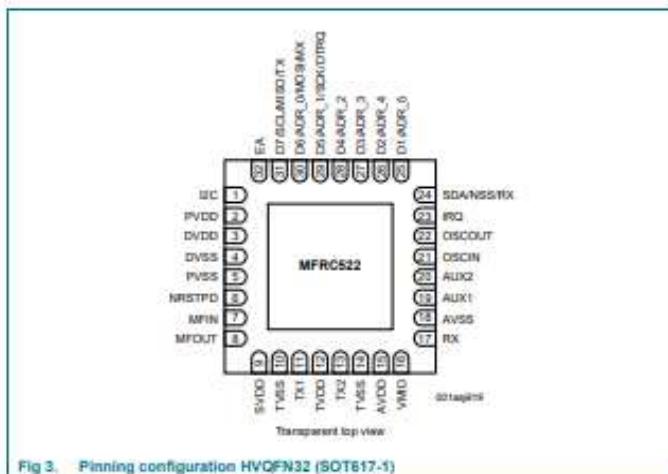


Fig 3. Pinning configuration HVQFN32 (SOT617-1)

7.1 Pin description

Table 3. Pin description

Pin	Symbol	Type	Description
1	I2C	I	PC-bus enable input
2	PVDD	P	pin power supply
3	DVDD	P	digital power supply
4	DVSS	G	digital ground
5	PVSS	G	pin power supply ground
6	NRSTPD	I	reset and power-down input: power-down: enabled when LOW; internal current sinks are switched off, the oscillator is inhibited and the input pins are disconnected from the outside world reset: enabled by a positive edge
7	MFIN	I	MIFARE signal input
8	MFOUT	O	MIFARE signal output
9	SVDD	P	MFIN and MFOUT pin power supply
10	TVSS	G	transmitter output stage 1 ground
11	TX1	O	transmitter 1 modulated 13.56 MHz energy carrier output
12	TVDD	P	transmitter power supply: supplies the output stage of transmitters 1 and 2
13	TX2	O	transmitter 2 modulated 13.56 MHz energy carrier output
14	TVSS	G	transmitter output stage 2 ground
15	AVDD	P	analog power supply

MFRC522

Product data sheet
COMPANY PUBLIC

All information contained in this document is subject to legal protection.

Rev. 3.0 — 17 September 2014
112158

©2007 NXP Semiconductors N.V. All rights reserved.

6 of 95

3. Data Sheet Sensor Ultrasonik (HC-SR04)



Tech Support: services@elecfreaks.com

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) If the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level time * velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

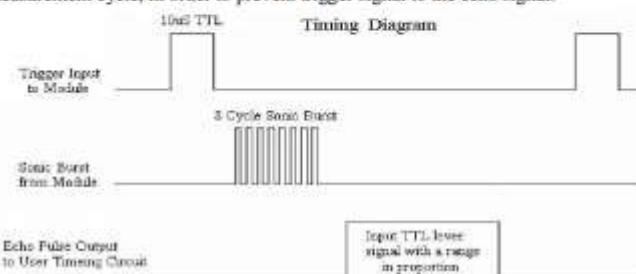
Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm



Timing diagram

The Timing diagram is shown below. You only need to supply a short 10 μ s pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $uS / 58 = \text{centimeters}$ or $uS / 148 = \text{inch}$; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



4. Data Sheet Modul I2C



International Components Distributor
A MOBICON COMPANY

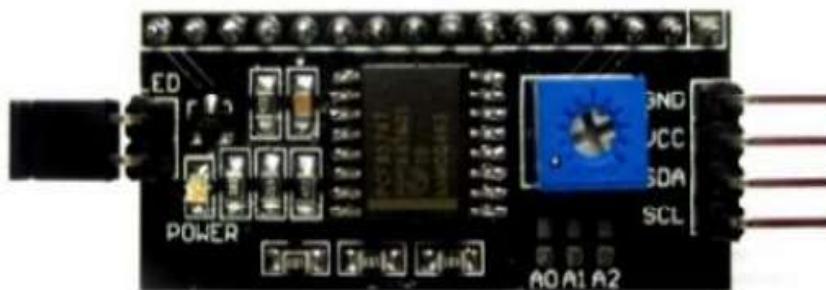
TEL JHB : (011) 493-9307
CAPE : (021) 535-3150
KZN : (031) 309-7686

FAX : (011) 403-0310

sales@mantech.co.za

www.mantech.co.za

I2C Interface for LCD



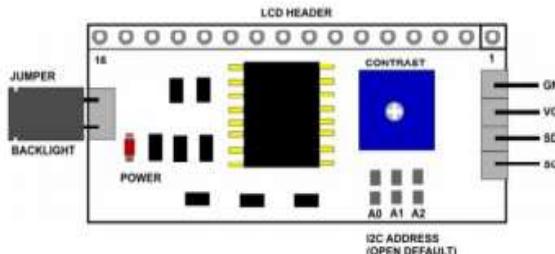
Description:

This LCD2004 is a great I2C interface for 2x16 and 4x20 LCD displays. With the limited pin resources, your project may be out of resources using normal LCD shield. With this I2C interface LCD module, you only need 2 lines (I2C) to display the information. If you already has I2C devices in your project, this LCD module actually cost no more resources at all. Fantastic for Arduino based projects.

Specification:

Compatible with 16x2 and 20x4 LCD's
Default I2C Address = 0X27
Address selectable - Range 0x20 to 0x27

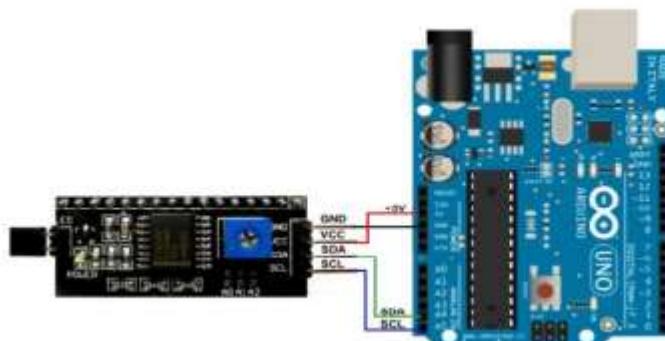
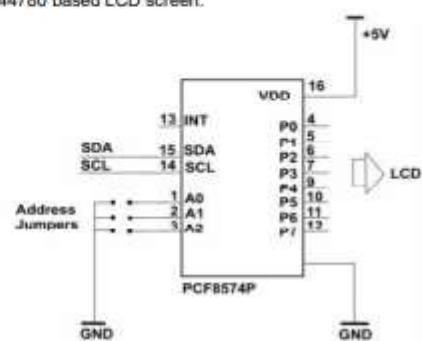
Board Layout:



The LCD2004 board utilized the PCF8574 I/O expander. This nifty little chip provides eight bits of parallel I/O addressable by a I²C bus address – 0x00 to 0x27. SainSmart tied all address leads to V_{cc}, so the LCD2004 board's I²C address is permanently fixed at hex 27. This is rather limiting since no additional LCD2004s can be added to the bus. Anyway, you simply address the board and write an eight bit value which is then presented on the output pins of the PCF8574, which, in this case, are connected to the HD44780 based LCD screen.

INPUTS			I ² C SLAVE ADDRESS
A2	A1	A0	
L	L	L	0x20
L	L	H	0x21
L	H	L	0x22
L	H	H	0x23
H	L	L	0x24
H	L	H	0x25
H	H	L	0x26
H	H	H	0x27

H = Open Jumper L = Close Jumper



5. Data Sheet Motor Stepper 28BYJ-48

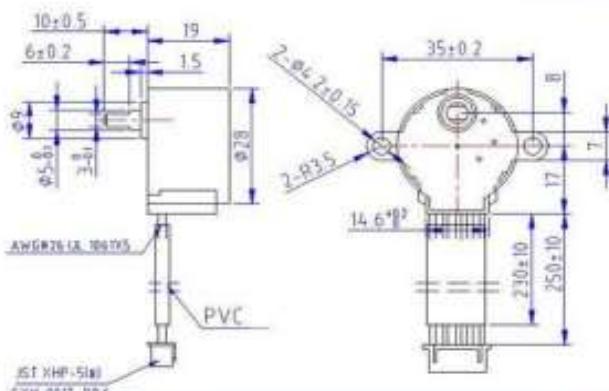
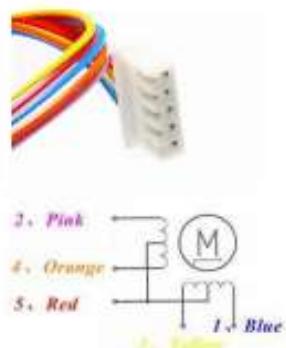


28BYJ-48 - 5V Stepper Motor

The 28BYJ-48 is a small stepper motor suitable for a large range of applications.



Rated voltage	5VDC
Number of Phases	4
Speed Variation Ratio	1/64
Stride Angle	5.625°/64
Frequency	100Hz
DC resistance	500Ω±2%(25°C)
Idle In-traction Frequency	> 600Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3Nm.m(120Hz)
Self-positioning Torque	>34.3Nm.m
Friction torque	600-1200 g.cm
Pull-in torque	300 g.cm
Insulated resistance	>10MD(500V)
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in Temperature	<40K(120Hz)
Noise	<35dB(120Hz, No load, 10cm)
Modal	28BYJ-48 - 5V



6. Data Sheet Motor Driver ULN2003



**ULN2001A-ULN2002A
ULN2003A-ULN2004A**

SEVEN DARLINGTON ARRAYS

- SEVEN DARLINGTONS PER PACKAGE
- OUTPUT CURRENT 500mA PER DRIVER (600mA PEAK)
- OUTPUT VOLTAGE 50V
- INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS
- OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
- TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT



DIP16

ORDERING NUMBERS: ULN2001A/2A/3A/4A



SO16

ORDERING NUMBERS: ULN2001D/2D/3D/4D

DESCRIPTION

The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

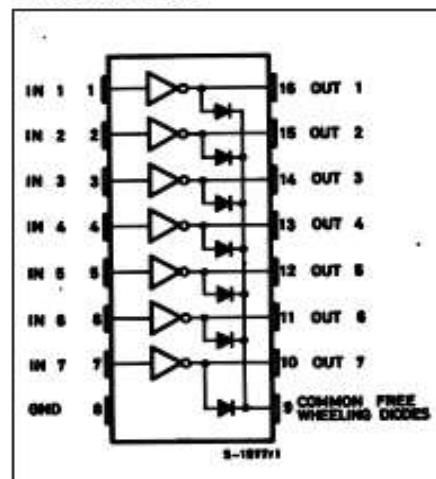
The four versions interface to all common logic families:

ULN2001A	General Purpose, DTL, TTL, PMOS, CMOS
ULN2002A	14-25V PMOS
ULN2003A	5V TTL, CMOS
ULN2004A	6-15V CMOS, PMOS

These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal print-heads and high power buffers.

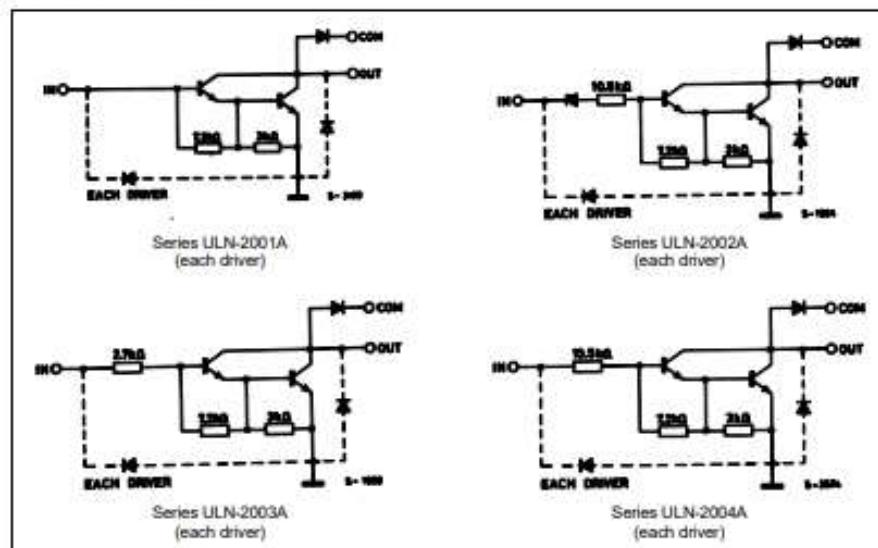
The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper leadframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.

PIN CONNECTION



ULN2001A - ULN2002A - ULN2003A - ULN2004A

SCHEMATIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_{in}	Input Voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
I_c	Continuous Collector Current	500	mA
I_b	Continuous Base Current	25	mA
T_{amb}	Operating Ambient Temperature Range	-20 to 85	°C
T_{stg}	Storage Temperature Range	-55 to 150	°C
T_j	Junction Temperature	150	°C

THERMAL DATA

Symbol	Parameter	DIP16	SO16	Unit
$R_{th(jamb)}$	Thermal Resistance Junction-ambient	Max. 70	120	°C/W

ULN2001A - ULN2002A - ULN2003A - ULN2004A

ELECTRICAL CHARACTERISTICS ($T_{amb} = 25^\circ\text{C}$ unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CE}	Output Leakage Current	$V_{CE} = 50\text{V}$, $T_{amb} = 70^\circ\text{C}$, $V_{CE} = 50\text{V}$			50 100	μA	1a 1a
		$T_{amb} = 70^\circ\text{C}$ for ULN2002A $V_{CE} = 50\text{V}$, $V_i = 6\text{V}$ for ULN2004A $V_{CE} = 50\text{V}$, $V_i = 1\text{V}$			500	μA	1b
					500	μA	1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	$I_C = 100\text{mA}$, $I_B = 250\mu\text{A}$ $I_C = 200\text{mA}$, $I_B = 350\mu\text{A}$ $I_C = 350\text{mA}$, $I_B = 500\mu\text{A}$		0.9 1.1 1.3	1.1 1.3 1.6	V	2 2 2
$I_{(sat)}$	Input Current	for ULN2002A, $V_i = 17\text{V}$ for ULN2003A, $V_i = 3.85\text{V}$ for ULN2004A, $V_i = 5\text{V}$ $V_i = 12\text{V}$		0.82 0.93 0.35 1	1.25 1.35 0.5 1.45	mA	3 3 3 3
$I_{(sat)}$	Input Current	$T_{amb} = 70^\circ\text{C}$, $I_C = 500\mu\text{A}$	50	65		μA	4
$V_{(sat)}$	Input Voltage	$V_{CE} = 2\text{V}$ for ULN2002A $I_C = 300\text{mA}$ for ULN2003A $I_C = 200\text{mA}$ $I_C = 250\text{mA}$ $I_C = 300\text{mA}$ for ULN2004A $I_C = 125\text{mA}$ $I_C = 200\text{mA}$ $I_C = 275\text{mA}$ $I_C = 350\text{mA}$			13 2.4 2.7 3 5 6 7 8	V	5
h_{FE}	DC Forward Current Gain	for ULN2001A $V_{CE} = 2\text{V}$, $I_C = 350\text{mA}$	1000				2
C_i	Input Capacitance			15	25	pF	
t_{PLH}	Turn-on Delay Time	0.5 V_i to 0.5 V_d		0.25	1	μs	
t_{PHL}	Turn-off Delay Time	0.5 V_i to 0.5 V_d		0.25	1	μs	
I_R	Clamp Diode Leakage Current	$V_R = 50\text{V}$ $T_{amb} = 70^\circ\text{C}$, $V_R = 50\text{V}$			50 100	μA	6 6
V_F	Clamp Diode Forward Voltage	$I_R = 350\text{mA}$		1.7	2	V	7

ULN2001A - ULN2002A - ULN2003A - ULN2004A

TEST CIRCUITS

Figure 1a.

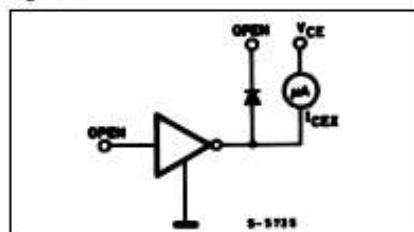


Figure 1b.

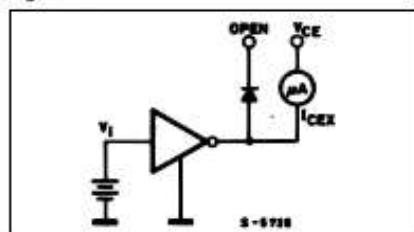


Figure 2.

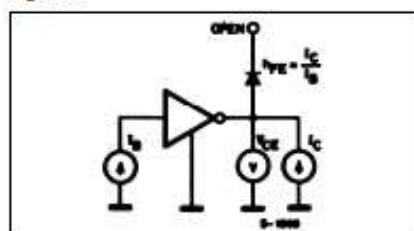


Figure 3.

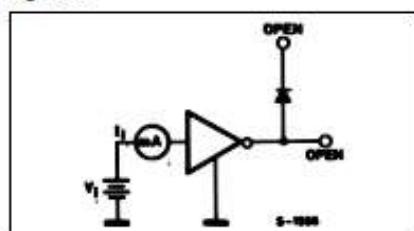


Figure 4.

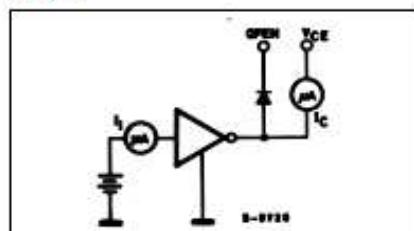


Figure 5.

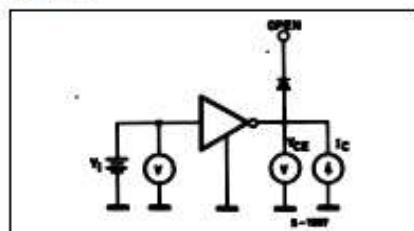


Figure 6.

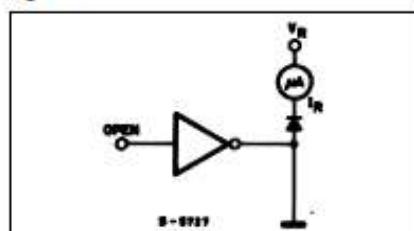
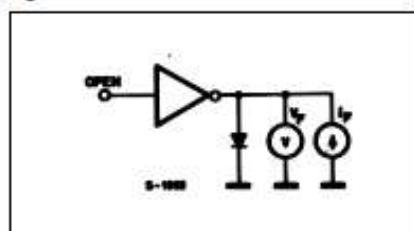


Figure 7.



ULN2001A - ULN2002A - ULN2003A - ULN2004A

Figure 8: Collector Current versus Input Current

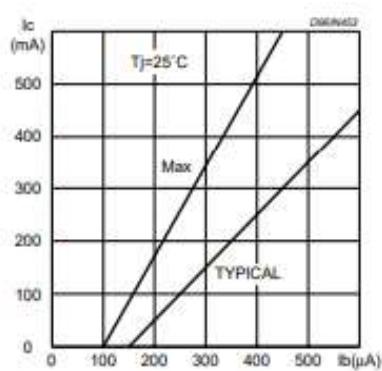


Figure 9: Collector Current versus Saturation Voltage

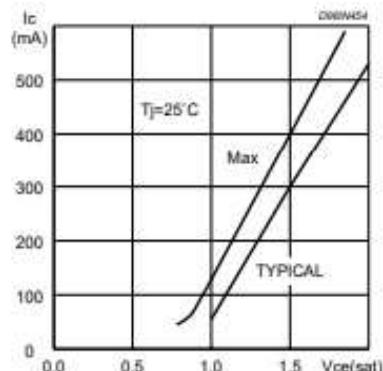


Figure 10: Peak Collector Current versus Duty Cycle

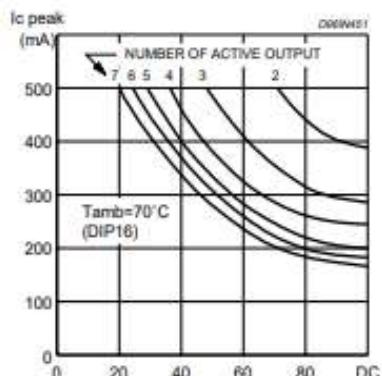
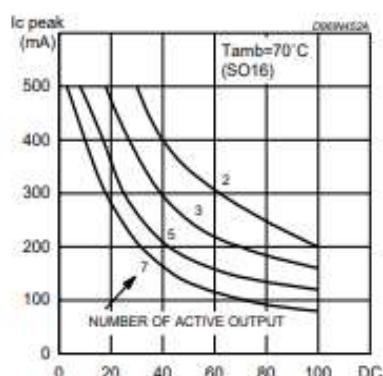


Figure 11: Peak Collector Current versus Duty Cycle



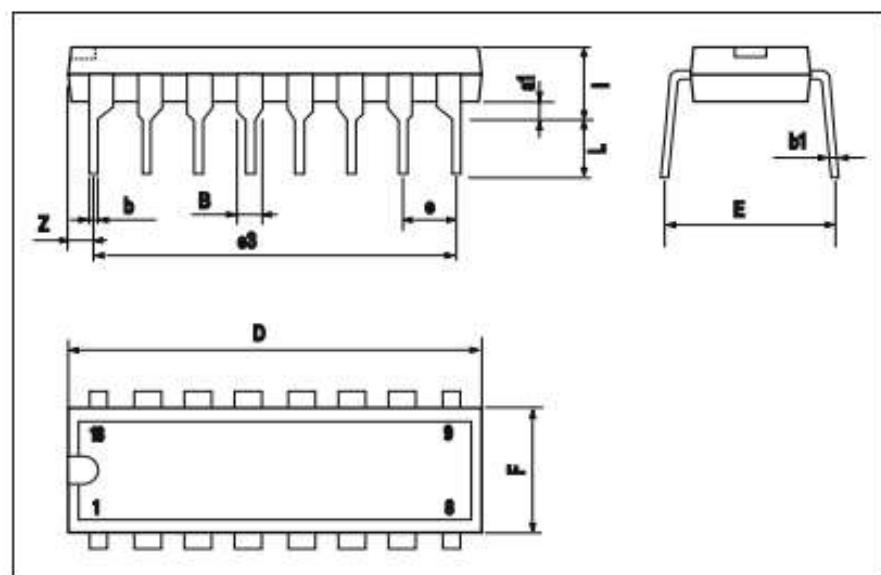
ULN2001A - ULN2002A - ULN2003A - ULN2004A

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.51			0.020		
B	0.77		1.65	0.030		0.065
b		0.5			0.020	
b1		0.25			0.010	
D			20			0.787
E		8.5			0.335	
e		2.54			0.100	
e3		17.78			0.700	
F			7.1			0.280
I			5.1			0.201
L		3.3			0.130	
Z			1.27			0.050

**OUTLINE AND
MECHANICAL DATA**



DIP16



B1. Listing Program Keseluruhan

```
#include <SPI.h> // serial peripheral interface  
#include <MFRC522.h> // library rfid  
#include <Wire.h> // wire virtual  
#include <LiquidCrystal_I2C.h> //library lcd i2c  
#include <Stepper.h> // library motor stepper  
  
#define SS_PIN 10 // menetapkan pin rfid dengan pin 10 arduino  
#define RST_PIN 9 // menetapkan pin rfid dengan pin 9 arduino  
#define LED_ACCESS_PIN 7 // pin 7 led pada arduino  
  
#define motorSteps 64 // step dari motor stepper  
#define motorPin1 6 // pin IN1 motor driver ULN2003  
#define motorPin2 5 // pin IN2 motor driver ULN2003  
#define motorPin3 4 // pin IN3 motor driver ULN2003  
#define motorPin4 3 // pin IN4 motor driver ULN2003  
#define button 8  
const int step_ = 1600; //putaran motor stepper  
int trigPin = A0; //pin sensor hc-sr04 di A0 arduino  
int echoPin = A1; ///pin sensor hc-sr04 di A0 arduino  
int button1 = 0; //push button
```

```

// initialize of the Stepper library:

Stepper myStepper(motorSteps, motorPin4, motorPin2, motorPin3, motorPin1);

LiquidCrystal_I2C lcd(0x27, 16, 2); // alamat modul i2c

MFRC522 mfrc522(SS_PIN, RST_PIN); // jarak dari modul mfrc522

void setup() // fungsi yang berjalan sekali

{

    myStepper.setSpeed(100); // kecepatan motor stepper

    // initialize the Serial port:

    Serial.begin(9600); // Initiate a serial communication

    lcd.begin(); //lcd 16x2

    lcd.setCursor(1,0); // posisi text pada lcd

    lcd.clear(); //hapus text pada lcd

    SPI.begin(); // Initiate SPI bus

    mfrc522.PCD_Init(); // Initiate MFRC522

    Serial.println("PLEASE TAG YOUR E-KTP CARD to the READER TAG...");

    Serial.println();

    pinMode( LED_ACCESS_PIN , OUTPUT); // pin led sebagai output

    pinMode( echoPin, INPUT ); //pin echo sebagai input

    pinMode( trigPin, OUTPUT); // pin trig sebagai output

    pinMode( button, INPUT_PULLUP); // pin push button sebagai input

}

void distance1()

{

    long duration, distance; //fungsi jarak sensor hc-sr04

    digitalWrite(trigPin,HIGH); //pin trig aktif

```

```

delayMicroseconds(1000); //delay waktu

digitalWrite(trigPin, LOW);

duration=pulseIn(echoPin, HIGH);

distance =(duration/2)/29.1;

Serial.print(distance);

Serial.println("CM");

if(distance<=10) //jika jarak kuang dari 10cm, sensor bekerja

{

myStepper.step(0); // motor stepper berhenti ketika sensor hc-sr04 memdeteksi jarak

yang sesuai

delay(2000); //delay 2detik

digitalWrite(LED_ACCESS_PIN, HIGH); //led menyala

lcd.clear(); //hapus text

lcd.setBacklight(HIGH);

lcd.setCursor(1,0);

lcd.print("ADA OBJEK!"); //text pada lcd

digitalWrite(LED_ACCESS_PIN, HIGH); //led menyala

delay(1000);

digitalWrite(LED_ACCESS_PIN, LOW); // led mati

lcd.clear();

Serial.println("TRUE"); //teks pada serial monitor Arduino IDE

return distance1(); //kembali mendeteksi jarak sensor hc-sr04

}

void buttoon(){ //fungsi dari push button

button1 = digitalRead(button); //membaca push button

```

```

if (button1 == LOW) { //jika button di tekan, maka proses akan berjalan
    Serial.println("Authorized Access"); //teks di serial monitor
    Serial.println();
    lcd.setCursor(1, 0);
    lcd.println("ACCESS GRANTED"); //teks di lcd ketika akses diijinkan oleh rfid
    lcd.println();
    lcd.clear();
    lcd.setBacklight(HIGH);
    lcd.setCursor(4, 0); // intruksi menentukan kursor
    lcd.print("Sky Hunter"); //intruksi menulis string/karakter pada lcd 16x2
    lcd.setCursor(4, 1);
    lcd.print("Project");
    Serial.println("BUKA"); // intruksi menulis string/ karakter pada serial monitor
    delay(3000);
    digitalWrite(7, HIGH); // intruksi pin led 7 di Arduino agar menyala
    myStepper.step(step_);
    lcd.clear();
    lcd.setBacklight(HIGH);
    lcd.setCursor(2, 0);
    lcd.print("PASS THE DOOR");
    lcd.setCursor(0, 1);
    lcd.print("Welcome to Home");
    Serial.println("PASS THE DOOR");
    digitalWrite(7, LOW);
    delay(6000);
}

```

```

distance1() // variabel sensor hc-sr04

delay(6000);

lcd.clear();

lcd.setBacklight(HIGH);

lcd.setCursor(4, 0);

lcd.print("CLOSING");

lcd.setCursor(2, 1);

lcd.print("See You ^_^");

Serial.println("CLOSING");

digitalWrite(7, HIGH);

delay(3000);

myStepper.step(-step_);

Serial.println("TUTUP");

digitalWrite(7, LOW);

delay(3000);

lcd.clear();

myStepper.step(-step_); //intruksi motor stepper untuk menutup pagar

Serial.println("TUTUP");

digitalWrite(7, LOW);

delay(3000);

lcd.clear();

}

}

```

```

void loop() {

    lcd.setCursor(4, 0);

    // Print at cursor Location:

    lcd.print("ASSALAMUALAIKUM");

    //goto column 2 and second line

    lcd.setCursor(0, 1);

    lcd.print("Sky Hunter Team ");

    buttoon();

    // Look for new cards

    if ( ! mfrc522.PICC_IsNewCardPresent()) //inisialisasi rfid

    {

        return;

    }

    // Select one of the cards

    if ( ! mfrc522.PICC_ReadCardSerial()) //baca kartu

    {

        return;

    }

    //Show UID on serial monitor

    Serial.print("UID tag :");

    String content= "";

    byte letter;

    for (byte i = 0; i < mfrc522.uid.size; i++)

    {

```

```

Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
Serial.print(mfrc522.uid.uidByte[i], HEX);
content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
content.concat(String(mfrc522.uid.uidByte[i], HEX));
}

Serial.println();
Serial.print("Message : ");
content.toUpperCase()

if (content.substring(1) == "04 28 5D 4A F9 2C 80") // nomor tag kartu yang
mendapat akses

{
    Serial.println("Authorized access");
    Serial.println();
    lcd.setCursor(1, 0);
    lcd.println("ACCESS GRANTED");
    lcd.println();
    lcd.clear();
    lcd.setBacklight(HIGH);
    lcd.setCursor(4, 0);
    lcd.print("OPENING");
    lcd.setCursor(4, 1);
    lcd.print("GARAGE");
    Serial.println("BUKA");
    delay(3000);
    digitalWrite(7, HIGH);
}

```

```
myStepper.step(step_);

lcd.clear();

lcd.setBacklight(HIGH);

lcd.setCursor(2, 0);

lcd.print("PASSED");

lcd.setCursor(0, 1);

lcd.print("Welcome to Home");

Serial.println("PASSED");

digitalWrite(7, LOW);

delay(6000);

distance1();

delay(6000);

lcd.clear();

lcd.setBacklight(HIGH);

lcd.setCursor(4, 0);

lcd.print("CLOSING");

lcd.setCursor(2, 1);

lcd.print("See You ^_^");

Serial.println("CLOSING");

digitalWrite(7, HIGH);

delay(3000);

myStepper.step(-step_);

Serial.println("TUTUP");

digitalWrite(7, LOW);
```

```

delay(3000);

lcd.clear();

myStepper.step(-step_);

Serial.println("TUTUP");

digitalWrite(7, LOW);

delay(3000);

lcd.clear();

}

Else //jika kartu di tolak maka tidak ada proses berjalan

{

Serial.println("ACCESS DENIED"); //keluaran teks di serial monitor ketika akses

ditolak

digitalWrite(7, HIGH);

delay(100);

digitalWrite(7, LOW);

delay(100);

digitalWrite(7, HIGH);

delay(100);

digitalWrite(7, LOW);

delay(100);

digitalWrite(7, HIGH);

delay(100);

digitalWrite(7, LOW);

delay(100);

```

```
Serial.println(" Access denied");

Serial.println();

lcd.clear();

lcd.setBacklight(HIGH);

lcd.setCursor(1, 0);

lcd.print("Access Denied");

delay(3000);

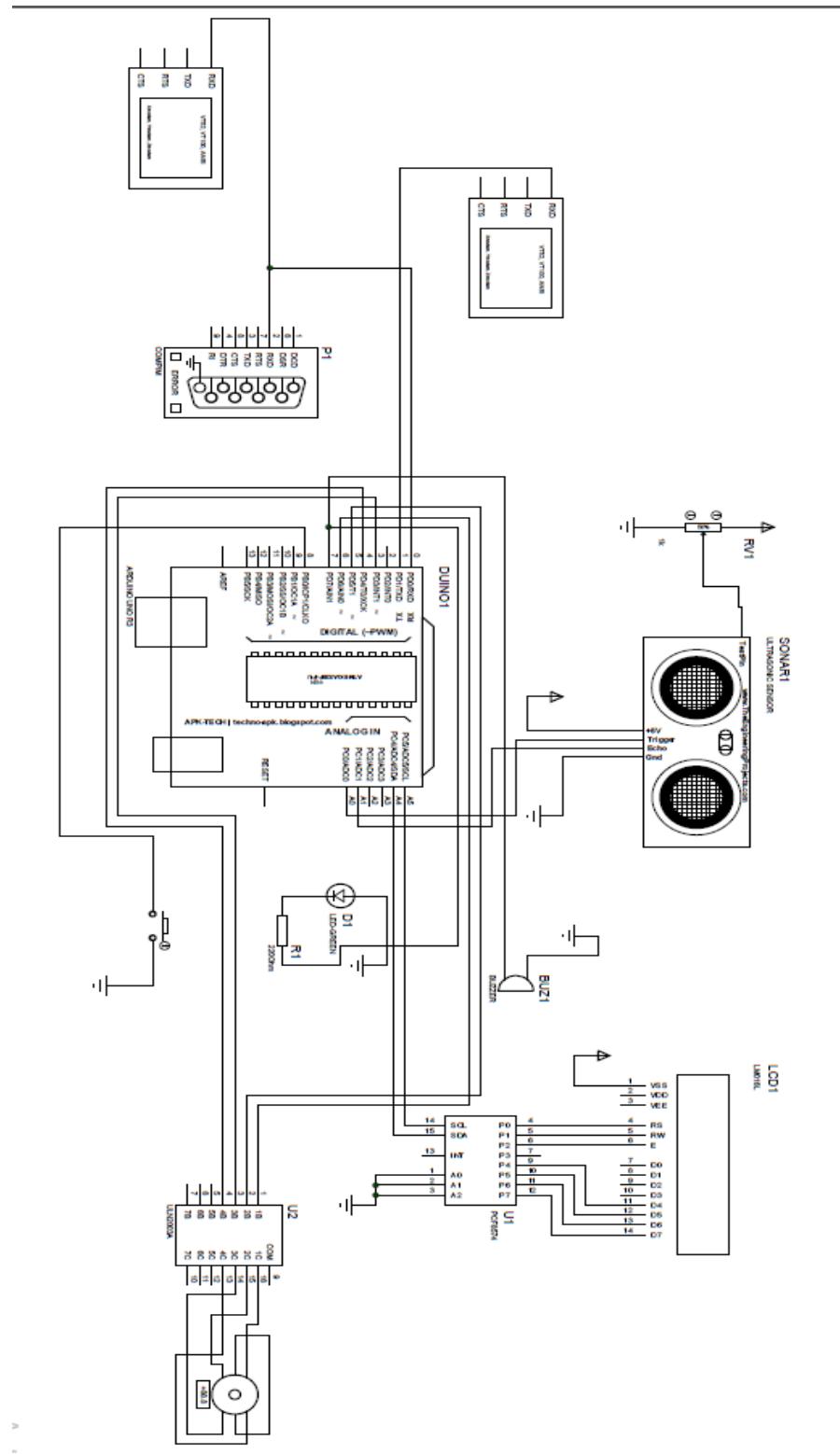
lcd.clear();

}

lcd.clear();

}
```

C1. Skema Rangkaian Keseluruhan



D1. Daftar Komponen dan Daftar Harga

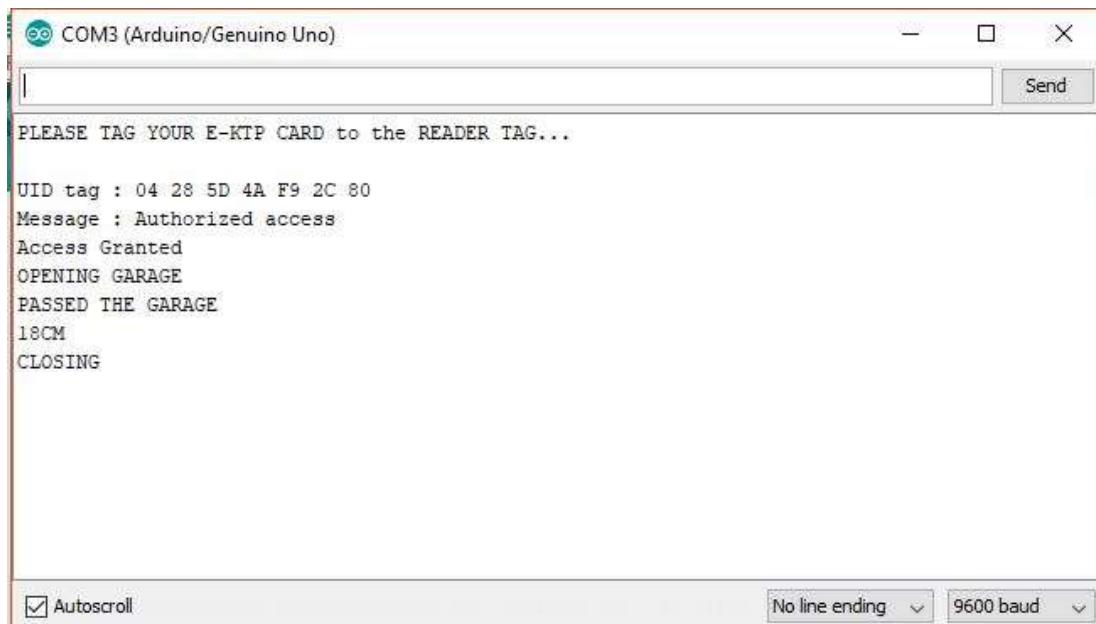
No	Nama Komponen	Harga (Rp)
1	(1pcs) Arduino Uno	Rp. 95.000,-
2	(1pcs) Sensor HC-SR 04	Rp. 30.000,-
3	(1pcs) Modul I2C	Rp. 25.000,-
4	(1pcs) Buzzer	Rp. 7.500,-
5	(40pcs) Kabel Male – Female	Rp. 40.000,-
6	(1pcs) PCB 10x20cm	Rp. 7.000,-
7	(5pcs) LED	Rp. 1500,-
8	(1pcs) Resistor	Rp. 100,-
9	(1pcs) Kabel Jumper	Rp. 2.000,-
10	(1pcs) Modul RFID MFRC522	Rp. 35.000,-
11	(1pcs) BreadBoard	Rp. 10.000,-
12	(1pcs) Motor Driver ULN2003	Rp. 25000,-
13	(4pcs) Push Button	Rp. 2000,-
Total Harga Rp. 280.100,-		

E1. Foto Alat



F1. Tampilan Serial Monitor Arduino IDE

1. Tampilan Serial Monitor



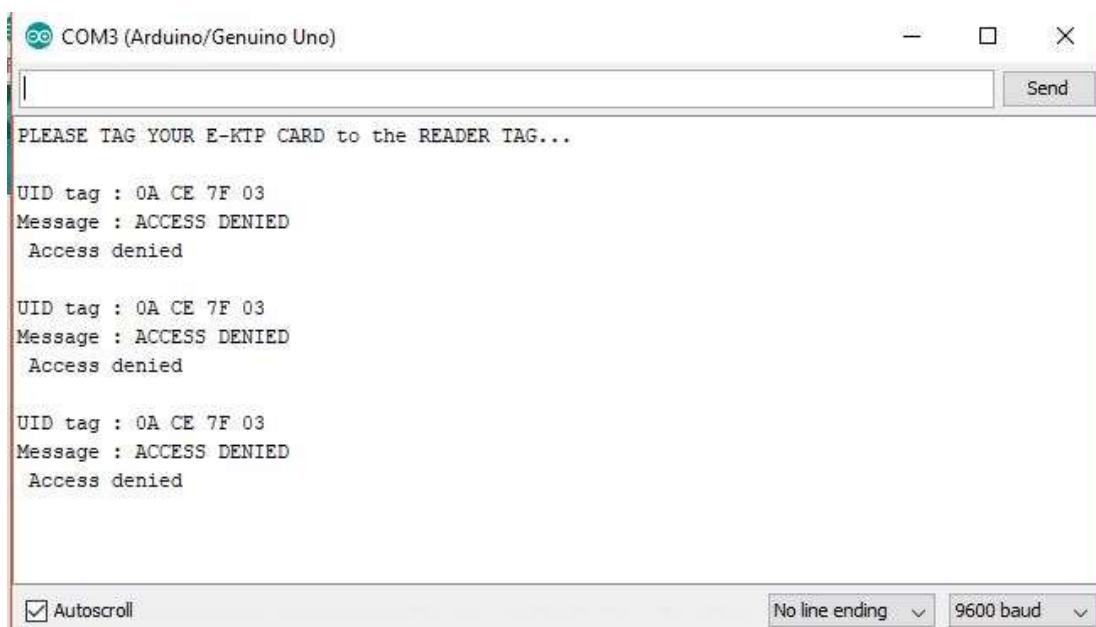
The screenshot shows the Arduino Serial Monitor window titled "COM3 (Arduino/Genuino Uno)". The text output is as follows:

```
PLEASE TAG YOUR E-KTP CARD to the READER TAG...

UID tag : 04 28 5D 4A F9 2C 80
Message : Authorized access
Access Granted
OPENING GARAGE
PASSED THE GARAGE
18CM
CLOSING
```

The bottom status bar shows "Autoscroll" checked, "No line ending" selected, and "9600 baud" as the baud rate.

a). Tampilan Serial monitor saat – akses kartu diijinkan dan memulai proses



The screenshot shows the Arduino Serial Monitor window titled "COM3 (Arduino/Genuino Uno)". The text output is as follows:

```
PLEASE TAG YOUR E-KTP CARD to the READER TAG...

UID tag : 0A CE 7F 03
Message : ACCESS DENIED
Access denied

UID tag : 0A CE 7F 03
Message : ACCESS DENIED
Access denied

UID tag : 0A CE 7F 03
Message : ACCESS DENIED
Access denied
```

The bottom status bar shows "Autoscroll" checked, "No line ending" selected, and "9600 baud" as the baud rate.

b). Tampilan Serial Monitor- saat akses kartu di tolak.

```
PLEASE TAG YOUR E-KTP CARD to the READER TAG...

UID tag : 04 28 5D 4A F9 2C 80
Message : Authorized access
Access Granted
OPENING GARAGE
PASSED THE GARAGE
5CM
JARAK SESUAI
7CM
JARAK SESUAI
41CM
CLOSING
CLOSING THE GARAGE

 Autoscroll
No line ending
9600 baud
```

c). Tampilan Serial Monitor- saat sensor HC-SR04 mendekripsi objek ketika ada objek yang berhenti saat pagar sedang menutup