



PROYEK AKHIR - VE180626

**SISTEM *AUTOMATIC DOOR LOCK* DENGAN
MENGUNAKAN E-KTP BERBASIS MIKROKONTROLER**

Pratamanu Benwiramadhan
NRP. 1031160000031

Dosen Pembimbing
Ir. Joko Susila, MT.
Fauzi Imaduddin Adhim, S.ST.,MT.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

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



FINAL PROJECT - VE180626

***AUTOMATIC DOOR LOCK SYSTEM USING
MICROCONTROLLER BASED E-KTP***

Pratamanu Benwiramadhan
NRP. 1031160000031

Supervisor
Ir. Joko Susila, MT.
Fauzi Imaduddin Adhim, S.ST.,MT.

DEPARTEMEN OF ELECTRICAL AUTOMATION ENGINEERING
Vocations Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2019

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

PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini penulis menyatakan bahwa isi sebagian maupun keseluruhan buku Proyek Akhir dengan judul "**Sistem Automatic Door Lock Dengan Menggunakan E-KTP Berbasis Mikrokontroler**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang penulis akui sebagai karya sendiri.

Seluruh data hasil pengujian yang ditulis benar-benar asli tanpa penambahan atau pengurangan dan mampu dipertanggungjawabkan. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka dan telah terbukti validitasnya.

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

Surabaya, 17 Desember 2019



Pratamanu B.
NRP. 10311600000031

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

SISTEM *AUTOMATIC DOOR LOCK* DENGAN MENGUNAKAN *E-KTP* BERBASIS MIKROKONTROLER

LEMBAR PENGESAHAN PROYEK AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Menyetujui

Dosen Pembimbing I

Dosen Pembimbing II



Ir. Joko Susila, M.T.

Fauzi Imaduddin Adhim S.ST., MT.

NIP. 196606061991021001

NPP. 1991201711057

**SURABAYA
JANUARI, 2020**

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

SISTEM *AUTOMATIC DOOR LOCK* DENGAN ENGGUNAKAN E-KTP BERBASIS MIKROKONTROLER

Nama : Pratamanu Benwiramadhan
Pembimbing : Ir. Joko Susila, MT.
Fauzi Imaduddin Adhim, S.ST.,MT.

ABSTRAK

Kurang efektifnya pengaksesan ruangan dengan membawa banyak kunci, dimana ketika banyak membawa kunci memungkinkan kunci sering tertinggal maupun hilang, mendorong adanya pembuatan alat pengaman ruangan yang sekaligus mampu digunakan untuk presensi kehadiran yang nantinya dapat dimonitoring secara online. Pintu menjadi hal yang paling disorot dalam sistem keamanan, hal tersebut karena fungsi pintu sebagai akses utama.

Dengan adanya permasalahan ini, maka pada proyek akhir ini dikembangkan. Cara kerja alat ini yaitu dengan cara menempelkan RFID *Tag* pada RFID *Reader* yang telah dipasang, kemudian Arduino akan memproses apakah RFID *Tag* yang digunakan terdaftar atau tidak. Jika Arduino membaca RFID *Tag* yang sudah terdaftar maka pengguna RFID *Tag* tersebut dapat mengakses ruangan. Sebaliknya jika RFID *Tag* yang digunakan belum terdaftar, akses akan ditolak. Alat ini juga dapat digunakan sebagai presensi dengan mengirimkan data RFID *Tag* terdaftar pada *database*. Kunci pintu yang dimaksudkan disini adalah solenoid *door lock*. Apabila RFID *Tag* yang ditempelkan tidak terdaftar maka akses akan ditolak dan tidak ada data yang akan dikirimkan pada *database*. RFID *Tag* yang digunakan disini merupakan E-KTP

Alat pengaman pintu otomatis menggunakan E-KTP ini mampu membaca ID E-KTP dengan jarak maksimal 2cm dengan sensor RFID *reader* yang diletakkan dalam *box* hitam dengan tebal 4mm. Persentase keberhasilan pengiriman data pada *database* yaitu sebesar 70% dan persentase kegagalannya 30%, dimana persentase kegagalan ini dikarenakan sinyal hotspot yang digunakan pada ESP32. Kendala dari *hardware* ini yaitu ketika *push button* yang digunakan untuk membuka pintu dari dalam ditekan secara terus menerus akan membuat Arduino mengalami gangguan atau tidak dapat bekerja.

Kata kunci : E-KTP, RFID, *Database*, *Automatic Door Lock*, Arduino, Presensi.

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

AUTOMATIC DOOR LOCK SYSTEM USING MICROCONTROLLER BASED E-KTP

Name : **Pratamanu Benwiramadhan**
Supervisor : **Ir. Joko Susila, MT.**
Fauzi Imaduddin Adhim, S.ST.,MT.

ABSTRACT

The ineffectiveness of room access by carrying lots of keys, which when carrying lots of keys allows keys to be often left behind or lost, encourages the creation of a room safety device that is at the same time able to be used for presence presence which can later be monitored online. The door is the most highlighted thing in the security system, this is because the door functions as the main access.

With this problem, the final project was developed. The way this tool works is by attaching the RFID Tag to the installed RFID Reader, then Arduino will process whether the RFID Tag used is registered or not. If Arduino reads the RFID Tag that has been registered then the RFID Tag user can access the room. Conversely, if the RFID Tag used is not yet registered, access will be denied. This tool can also be used as a presence by sending RFID Tag data registered to the database. The door lock intended here is the solenoid door lock. If the embedded RFID Tag is not registered then access will be denied and no data will be sent to the database. The RFID Tag used here is an E-KTP

Automatic door safety devices using E-KTP are able to read E-KTP IDs with a maximum distance of 2cm with an RFID reader sensor that is placed in a 4mm thick black box. The percentage of successful sending of data to the database is 70% and the percentage of failures is 30%, where the percentage of failures is due to the hotspot signal used on ESP32. The constraint of this hardware is that when the push button used to open the door from inside is pressed continuously it will make the Arduino experience interference or unable to work.

Keywords: E-KTP, RFID, Database, Automatic Door Lock, Arduino, Presence.

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

KATA PENGANTAR

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

Proyek Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma 3 di Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

SISTEM *AUTOMATIC DOOR LOCK* DENGAN ENGGUNAKAN E-KTP BERBASIS MIKROKONTROLER

Dalam Proyek Akhir ini rancang bangun pengamanan dan presensi pintu guna untuk lebih menambah keamanan pintu loker dengan memberi akses saat terbuka dengan kartu E-KTP sebagai *tag* pasif.

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Joko Susila, MT. dan Bapak Fauzi Imaduddin Adhim S.ST. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Proyek Akhir ini, Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Proyek Akhir ini.

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

Surabaya, 17 Desember 2019

Penulis

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

DAFTAR ISI

SAMPUL LUAR	i
SAMPUL DALAM.....	iii
PERNYATAAN KEASLIAN PROYEK AKHIR	v
LEMBAR PENGESAHAN	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Permasalahan.....	2
1.3 Tujuan	2
1.4 Sistematika Laporan	3
1.5 Relevansi	3
1.6 Batasan Masalah.....	3
BAB II TEORI PENUNJANG	5
2.1 E-KTP	5
2.2 RFID Reader	7
2.2.1 Cara Kerja RFID Reader.....	8
2.2.2 Tag Pasif (<i>Passive Tags</i>).....	9
2.2.3 RFID Reader Membaca Data ID Dari <i>Smart Card</i>	
10	
2.3 Arduino Mega	11
2.4 Solenoid DC	12
2.5 <i>Liquid Crystal Display (LCD)</i>	13
2.6 ESP32S.....	14
2.7 RTC DS3231	15
2.8 Modul Relay.....	16
BAB III PERENCANAAN DAN IMPLEMENTASI	17
3.1 Perancangan Mekanik	19
3.2 Perancangan Perangkat Elektrik.....	21
3.2.1 <i>Wiring</i> Modul RFID Reader RS522.....	21
3.2.2 <i>Wiring</i> Modul LCD 20x4 Dengan I2c	22
3.2.3 Rangkaian <i>Relay</i> Dan Solenoid.....	23
3.2.4 <i>Wiring</i> ESP32.....	24

3.2.5	<i>Wiring Real Time Clock (RTC) DS3231</i>	24
3.3	Perancangan <i>Software</i>	25
3.3.1	<i>Flowchart</i> Keseluruhan	25
3.3.2	<i>Flowchart</i> Membuka Pintu Menggunakan E-KTP 27	
3.3.3	<i>Flowchart</i> LCD <i>Display</i> 20x4	29
3.3.4	<i>Flowchart</i> Pengiriman Data ke <i>Database</i>	31
BAB IV	PENGUJIAN DAN ANALISIS	33
4.1	Pengujian Dan Analisa Perangkat Elektrik	33
4.1.1	Pengujian Arduino Mega	33
4.1.2	Pengujian Jarak Sensor RFID <i>Reader</i> Dengan E-KTP	33
4.1.3	Perangkat Lunak (<i>Software</i>) Pengaman dan Presensi Menggunakan E-KTP Berbasis Arduino Mega	38
4.1.4	Pengujian Sistem Otomasi Pintu Menggunakan E-KTP	39
4.1.4.1	Pengujian LCD 20x4 I2c <i>Display</i>	39
4.1.4.2	Pengujian Pengaturan <i>Delay</i> pada <i>Relay</i>	42
4.1.4.3	Pengujian RFID <i>Reader</i>	43
4.1.4.4	Pengujian (Solenoid) Sistem Untuk Membuka Dan Mengunci	43
4.1.4.5	Pengujian Pengiriman Data dari Arduino ke <i>Database</i>	44
4.2	Pengujian Sistem Keseluruhan	55
BAB V	PENUTUP	59
5.1	Kesimpulan	59
5.2	Saran	59
DAFTAR PUSTAKA	61
LAMPIRAN A	A-1
LAMPIRAN B	B-1
DAFTAR RIWAYAT PENULIS	C-1

DAFTAR GAMBAR

Gambar 2.1 Lapisan E-KTP.....	5
Gambar 2.2 E-KTP Tampak Depan.....	7
Gambar 2.3 Cara Kerja RFID <i>Reader</i>	8
Gambar 2.4 Cara Kerja RFID Tag Pasif.....	10
Gambar 2.5 RFID <i>Reader</i> Membaca Data ID dari <i>Smart Card</i>	11
Gambar 2.6 Board Arduino Mega.....	12
Gambar 2.7 Solenoid DC.....	12
Gambar 2.8 Bagian Solenoid DC.....	13
Gambar 2.9 LCD 20x4.....	13
Gambar 2.10 Skema LCD.....	14
Gambar 2.11 ESP32.....	15
Gambar 2.12 Modul RTC DS3231.....	15
Gambar 3.1 Blok Fungsional Sistem.....	17
Gambar 3.2 Kotak Alat Tampak Depan.....	19
Gambar 3.3 Kotak Alat Tampak Belakang.....	19
Gambar 3.4 Kotak Alat Tampak Kiri.....	20
Gambar 3.5 Kotak Alat Tampak Kanan.....	20
Gambar 3.6 Kotak Alat Tampak Dalam.....	20
Gambar 3.7 Skema Modul RFID <i>Reader</i>	21
Gambar 3.8 Modul LCD 20x4 dengan Arduino.....	22
Gambar 3.9 Rangkain Relay dan Solenoid dengan Arduino.....	23
Gambar 3.10 Rangkaian ESP32 dengan Arduino.....	24
Gambar 3.11 Rangkaian RTC.....	25
Gambar 3.12 Flowchart Keseluruhan Alat.....	26
Gambar 3.13 Flowchart Pembacaan E-KTP dengan RFID.....	28
Gambar 3.14 Flowchart Menampilkan Data Pada LCD.....	30
Gambar 3.15 Flowchart Pengiriman Data ke <i>Database</i>	31
Gambar 4.1 Uji Coba Pembacaan E-KTP.....	34
Gambar 4.2 Pengujian Jarak E-KTP Dengan RFID Dalam <i>Box</i>	35
Gambar 4.3 Program Arduino Pada Alat Pengaman Ruang.....	38
Gambar 4.4 Program Tampilan LCD Awal.....	39
Gambar 4.5 LCD Menampilkan Karakter ID Terdaftar.....	40
Gambar 4.6 Program Menampilkan Karakter ID Pengakses.....	40
Gambar 4.7 LCD Menampilkan ID Kartu Terdaftar.....	41
Gambar 4.8 Program LCD ID Tidak Terdaftar.....	41
Gambar 4.9 LCD Menampilkan Akses Ditolak.....	42
Gambar 4.10 Program Delay Pada Relay.....	42

Gambar 4.11 Program Pengujian Pembacaan E-KTP	43
Gambar 4.12 Program Solenoid Keadaan Mengunci	44
Gambar 4.13 Program Solenoid Keadaan Terbuka.....	44
Gambar 4.14 Solenoid Pada Keadaan Terbuka.....	44
Gambar 4.15 Pengujian Pertama Pada <i>Database</i>	45
Gambar 4.16 Program Data E-KTP Tama	46
Gambar 4.17 Program Data E-KTP ALFAN	47
Gambar 4.18 Pengujian Kedua Pada <i>Database</i>	47
Gambar 4.19 Pengujian Ketiga Pada <i>Database</i>	49
Gambar 4.20 Pengujian Keempat Pada <i>Database</i>	51
Gambar 4.21 Pengujian Kelima Pada <i>Database</i>	53
Gambar 4.22 Keadaan Layar Awal Saat Pintu Dihidupkan	55
Gambar 4.23 Saat E-KTP Terdaftar Untuk Membuka Pintu	55
Gambar 4.24 Solenoid Ketika Membaca E-KTP Terdaftar	56
Gambar 4.25 Data yang Dikirim dari Arduino ke ESP32	56
Gambar 4.26 Data Parsing dari Arduino ke ESP32	57
Gambar 4.27 Pengiriman Data Arduino ke <i>Database</i>	57
Gambar 4.28 E-KTP Tidak Terdaftar Untuk Membuka Pintu	58

DAFTAR TABEL

Tabel 2.1 Banwith Frekuensi RFID	9
Tabel 2.2 Karakteristik RFID Tag Pasif	10
Tabel 2.3 Pin Pada LCD Display	14
Tabel 3.1 Sambungan Pin RFID <i>Reader</i> ke Arduino.....	21
Tabel 3.2 Sambungan Pin LCD ke Arduino	22
Tabel 3.3 Sambungan Pin Relay ke Arduino.....	23
Tabel 3.4 Sambungan Pin Solenoid ke Relay	23
Tabel 4.1 Pengambilan Data Jarak Pembacaan RFID.....	35
Tabel 4.2 Pengambilan Data Jarak 0,1 cm Sensor RFID Dalam <i>Box</i> .	35
Tabel 4.3 Pengambilan Data Jarak 0,5 cm Sensor RFID Dalam <i>Box</i> .	36
Tabel 4.4 Pengambilan Data Jarak 1 cm Sensor RFID Dalam <i>Box</i>	36
Tabel 4.5 Pengambilan Data Jarak 1,5 cm Sensor RFID Dalam <i>Box</i> .	37
Tabel 4.6 Pengambilan Data Jarak 2 cm Sensor RFID Dalam <i>Box</i>	37
Tabel 4.7 Pengujian <i>Database</i> E-KTP Tama.....	46
Tabel 4.8 Pengujian <i>Database</i> E-KTP ALFAN.....	47
Tabel 4.9 Pengujian Kedua <i>Database</i>	49
Tabel 4.10 Pengujian Ketiga <i>Database</i>	51
Tabel 4.11 Pengujian Keempat <i>Database</i>	53
Tabel 4.12 Pengujian Kelima <i>Database</i>	55

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

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini teknologi berkembang sangat cepat. Segala upaya dilakukan demi mempermudah pekerjaan manusia dari waktu ke waktu yang membutuhkan mobilitas tinggi dalam melakukan pekerjaan serta otomatisasi sehingga manusia mendapat kemudahan dari teknologi tersebut.

Dengan semakin pesatnya perkembangan teknologi di era industri moderen sekarang ini, berbagai tujuh macam teknologi banyak bermunculan mulai dari teknologi yang baru ditemukan, sampai teknologi yang merupakan perkembangan dari teknologi sebelumnya. Perkembangan teknologi untuk sebuah sistem keamanan juga diperlukan, khususnya sistem keamanan pada ruangan – ruangan seperti rumah, laboratorium, kantor, dan lain-lain. Mengingat banyaknya kasus pencurian terhadap barang berharga yang semakin meningkat. Pada umumnya pengaman ruangan berupa kunci kurang praktis jika banyak digunakan dan penggunaan nomor kombinasi yang terkesan tidak praktis karena memerlukan waktu lama untuk menemukan nomer yang tepat. Berdasarkan permasalahan tersebut, maka dibutuhkan teknologi pengaman yang terintegrasi yang nantinya juga akan dapat digunakan sebagai presensi dengan database. Teknologi tersebut diantaranya adalah *Radio Frequency Identification* (RFID).

Pada sistem pengaman menggunakan RFID masih terdapat kekurangan dan RFID *reader* masih menggunakan *reader* yang hanya bisa membaca RFID *tag*. Teknologi *Radio Frequency Identification* (RFID) sendiri telah banyak digunakan diberbagai bidang khususnya bidang proteksi keamanan yang dapat mengidentifikasi suatu objek. Teknologi ini jauh lebih menjamin keamanan dibandingkan dengan kunci manual, karena RFID lebih sulit untuk dibajak atau digandakan. Selain itu karena masih jarang penggunaannya. Sistem RFID ini terdiri dari tiga komponen utama, yaitu *tag* atau *transponder*, *reader*, dan *database*. *Tag* berfungsi sebagai alat pelabelan suatu objek yang didalamnya terdapat data tentang objek tersebut. Selanjutnya *reader* berfungsi sebagai alat

scanning atau pembaca informasi yang ada pada *tag*. Sedangkan fungsi *database* disini sebagai pelacak dan penyimpanan informasi objek-objek yang dimiliki oleh *tag*. Dalam penelitian ini, digunakan arduino Mega sebagai pengendalinya dan RFID sebagai sensor pengamannya.

Sensor RFID menggunakan *reader* yang dapat membaca E-KTP agar hanya pemilik E-KTP yang digunakan sebagai *tag* pasif yang sudah terdaftar saja yang dapat mengakses ruangan. Kelebihan dari penggunaan E-KTP sendiri adalah setiap E-KTP memiliki ID yang berbeda sehingga tidak mungkin sama dengan orang lain dan E-KTP merupakan identitas penting yang biasanya semua orang membawanya kemana saja, berbeda dengan kunci yang terkadang orang – orang lupa membawanya. E-KTP yang tidak terdaftar maka secara otomatis sistem akan menolaknya dan pintu tidak akan bisa dibuka.

Mengacu pada latar belakang tersebut penulis mempunyai gagasan untuk membuat sebuah alat akses kontrol loker menggunakan sistem RFID dengan memanfaatkan E-KTP sebagai *tag* pasifberbasis arduino Mega.

1.2 Permasalahan

Perumusan masalah yang dibahas dalam pengerjaan Proyek Akhir ini adalah sebagai berikut:

1. Mudahnya pencuri membuka pengunci pada pintu yang terpasang menggunakan kunci tiruan jika pintu menggunakan sistem kunci yang dapat ditiru.
2. Tidak nyaman jika terlalu banyak kunci yang harus dibawa.
3. Maraknya titip absen karena presensi secara manual dan tanpa dibatasi oleh jam

1.3 Tujuan

Tujuan yang dicapai dalam penyelesaian pada Proyek Akhir ini adalah sebagai berikut

1. Merealisasikan alat pengaman ruangan sekaligus presensi menggunakan E-KTP sebagai RFID *tag* berbasis Arduino MEGA
2. Cara menyajikan dan pengiriman data berupa nama pengakses, tanggal dan waktu akses melalui *database*.

1.4 Sistematika Laporan

Untuk pembahasan lebih lanjut, laporan Proyek Akhir ini disusun dengan sistematika sebagai berikut:

- Bab I **PENDAHULUAN**
Membahas tentang latar belakang, perumusan masalah, batasan masalah, maksud dan tujuan, sistematika laporan, metodologi, serta relevansi Proyek Akhir yang dibuat.
- Bab II **TEORI PENUNJANG**
Menjelaskan teori yang berisi teori-teori dasar yang dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat.
- Bab III **PERENCANAAN DAN IMPLEMENTASI**
Membahas perencanaan dan pembuatan tentang perencanaan dan pembuatan *hardware* yang meliputi desain mekanik dan perancangan *software* yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.
- Bab IV **PENGUJIAN DAN ANALISA**
Membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan penganalisaan terhadap alat.
- Bab V **PENUTUP**
Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari Proyek Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

1.5 Relevansi

Membantu pengguna dalam meningkatkan keamanan tempat maupun ruang penyimpanan pada pengguna yang berada di industri. Mampu mengetahui siapa saja pengakses ruangan serta tanggal dan waktu kapan pintu loker dibuka melalui *database*

1.6 Batasan Masalah

Dalam pembuatan alat untuk proyek akhir ini, alat mempunyai batasan – batasan masalah sebagai berikut ini.

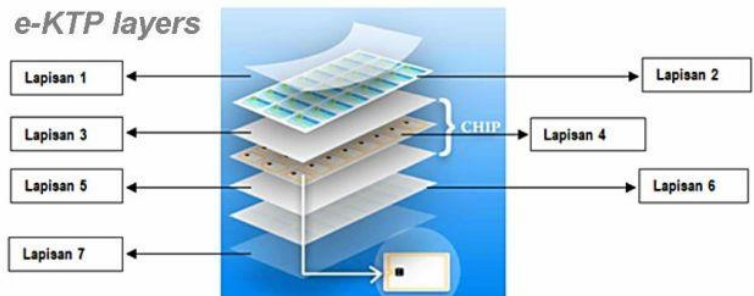
1. Tidak bisa registrasi secara langsung (mendaftarkan E-KTP) harus melepas alat dan memprogram ulang melalui Arduino IDE.
2. Alat tidak bisa bekerja setelah *push button* ditekan berkali-kali.
3. Pengiriman data dari ESP32 ke *database* hanya bisa menggunakan *mobile hotspot* dari pembuat alat.

BAB II TEORI PENUNJANG

2.1 E-KTP

E-KTP adalah KTP yang dilengkapi dengan *contactless chip* berisi biodata, tanda tangan, pas foto dan sidik jari telunjuk kanan & kiri penduduk yang bersangkutan. “*chip*” pada e-KTP pada dasarnya hampir mirip dengan *chip* yang ada pada memori komputer, memori *flashdisk*, memori kartu telepon dan lain-lain. Bahan fisik *chip* yang tipis seperti kertas didominasi oleh silikon dan jenis plastik, tidak tahan panas, korosi, basah atau lembab serta dapat rusak akibat patah, sobek dan jenis pengrusakan fisik lainnya.

Namun kelemahan-kelemahan itu tiada artinya, setelah *micro chip* jenis ini selalu berada di dalam sebuah kartu yang *solid* dan *micro chip* ini secara otomatis sangat terlindungi, menjadi lebih tahan panas, tahan air atau basah, tahan banting, tahan korosi, tahan sobek atau patah, karena *chip* berada didalam lapisan kartu yang bersangkutan. *Chip* E-KTP menggunakan antar muka nirsentuh (*contactless*) yang memenuhi standar ISO 14443 A/B. Transmisi data melalui gelombang radio (RF). Blangko E-KTP terbuat dari bahan PETG, semacam polimer termoplastik, yang tersusun dalam tujuh lapisan seperti pada Gambar 2.1.



Gambar 2. 1 Lapisan E-KTP

Sumber <https://www.bppt.go.id/profil/organisasi/100-press-release/press-release-2013/1664-press-release-pusat-teknologi-informasi-dan-komunikasi-bppt>

Penggunaan *chip* dewasa ini sudah sangat banyak karena tingkat keamanannya lebih tinggi. Misalnya pihak perbankan sudah mengalihkan teknologi ATM mereka dari sistem kartu berbasis *Magnetic Stripe* ke Kartu *Chip (Smart Card)*. Sedangkan E-KTP sendiri secara mekanisme teknis memiliki keuntungan:

1. *Chip* E-KTP dilindungi, salah satunya, dengan mekanisme autentikasi dua arah, yaitu suatu mekanisme untuk saling mengenali antara *chip* E-KTP dengan *card reader*, di mana *chip* harus dapat mengenali *card reader* (arah 1) dan *card reader* harus dapat mengenali *chip* (arah 2), setelah melalui mekanisme autentikasi ini maka data yang tersimpan di dalam *chip* baru dapat dibaca oleh *card reader*.
2. *Card reader* harus menghasilkan medan RF frekuensi tinggi untuk memberikan pasokan daya yang sesuai dengan kebutuhan *chip* E-KTP, di mana medan RF tersebut akan dimodulasikan untuk keperluan komunikasi. Frekuensi dari medan RF pada E-KTP *card reader* adalah $13,56 \text{ MHz} \pm 7 \text{ kHz}$.
3. Kisaran dari besar medan magnet (RF) yang dihasilkan oleh card reader adalah mengikuti ketentuan dalam ISO/IEC 14443, yaitu antara 1,5 A/m sampai dengan 7,5 A/m. Sedangkan besar frekuensi dari modulasi *amplitudo* medan magnet tersebut (yaitu medan RF), yang digunakan untuk mengirimkan data ke *chip* E-KTP, adalah 13,56 MHz.
4. Identifikasi ketunggalan data penduduk pada penerapan E-KTP menggunakan Sistem Identifikasi Biometrik.
5. Setiap manusia memiliki ciri-ciri fisik khusus yg unik dan dapat menunjukkan ketunggalan identitas seseorang dgn tingkat akurasi yg tinggi.

Chip yang tertanam dalam kartu ini memungkinkannya melakukan berbagai proses komputasi yang tidak dapat dilakukan oleh kartu berbasis *magnetic stripe*. Dengan kemampuan ini, kartu *chip* dapat menjalankan berbagai algoritma dan protokol keamanan yang cukup kompleks. Untuk penempatan *chip* dapat dilihat pada Gambar 2.2.



Gambar 2. 2 E-KTP Tampak Depan

Sumber <https://www.bppt.go.id/profil/organisasi/100-press-release/press-release-2013/1664-press-release-pusat-teknologi-informasi-dan-komunikasi-bppt>

Beberapa keamanan digital yang dapat dilakukan oleh *chip* di dalam E-KTP tersebut adalah:

1. Kerahasiaan data dengan menerapkan algoritma enkripsi dan mekanisme kendali akses.
2. Integritas data dengan menerapkan algoritma fungsi *hash* dan tanda tangan digital.
3. Keotentikan data dengan menerapkan protokol otentikasi yang berbasis algoritma enkripsi simetris ataupun asimetris.
4. *Non*-repudiasi dengan menerapkan tanda tangan digital.
5. Ketersediaan transaksi secara *offline* (*availability*) dengan dimungkinkannya pelaksanaan transaksi secara *offline* tersebut.
6. Tingkat keamanan yang disediakan oleh kartu *chip* juga tergantung dari jenis kartu dan standar yang digunakan.
7. Satu cara yang dapat dilakukan agar E-KTP anda masih baik adalah mengeceknya menggunakan *card reader* pembaca data. Tapi dengan syarat *chip* yang tertanam dalam kartu anda sudah diaktivasi dulu di Kecamatan.

Sebab sekarang banyak warga telah mengambil E-KTP langsung ke RT tanpa diaktifkan sebelumnya. Padahal aktivasi ini penting agar bisa langsung terkoneksi dengan sistem yang telah dibuat melalui *card reader*. [1]

2.2 RFID Reader

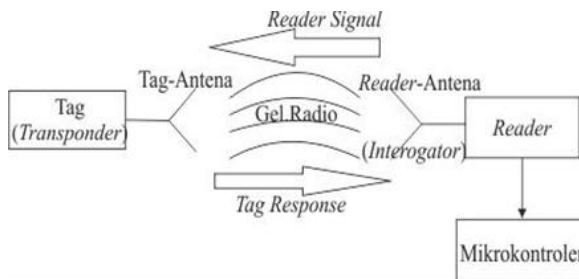
RFID *reader* adalah perangkat yang kompatibel dengan RFID *tag*. RFID *reader* akan memancarkan gelombang radio dan

menginduksi RFID tag, kemudian RFID tag akan mengirim data ID dari antenna yang terdapat pada rangkaian RFID tag melalui gelombang radio yang dipancarkan RFID reader.

Interface Software yang berfungsi untuk membaca data ID dari RFID reader dan mengolah data tersebut sehingga dapat digunakan menjadi *password*.

Sistem RFID akan berfungsi dengan baik diperlukan RFID reader yang dapat membaca RFID tag dan mengirim data yang dibaca ke *database*. Sebuah reader menggunakan antenna untuk berkomunikasi dengan RFID tag. Ketika reader memancarkan gelombang radio seluruh RFID tag yang memiliki frekuensi sama dengan reader akan memberikan respon. RFID reader memancarkan gelombang radio dan menginduksi RFID tag. Gelombang induksi tersebut berisi data ID dan jika dikenali oleh RFID tag, memori RFID tag (ID chip) akan terbuka. Gelombang radio yang dipancarkan oleh reader juga berfungsi sebagai catu daya RFID tag (tag pasif). Kemudian RFID tag akan mengirimkan kode yang terdapat di memori ID chip melalui antenna yang terpasang di RFID tag. RFID reader akan mengirim data tersebut ke mikrokontroler untuk diproses menjadi *password* sebagai pengamanan loker. [2]

2.2.1 Cara Kerja RFID Reader



Gambar 2. 3 Cara Kerja RFID Reader

Sumber <https://docplayer.info/32290372-Rancang-bangun-pengaman-pintu-otomatis-menggunakan-e-ktp-berbasis-mikrokontroler-atmega328.html>

Pada Gambar 2.3 komponen utama RFID tag adalah *chip* yang dapat menyimpan data atau informasi yang berisi nomor ID unik, chip ini terhubung dengan *tag-antena*. Informasi atau data yang tersimpan

dalam *chip* akan terkirim atau terbaca melalui gelombang radio setelah *tag*-antena menerima pancaran gelombang radio dari *reader*-antena (*interogator*) kemudian *reader* akan meneruskan data ke mikrokontroler.

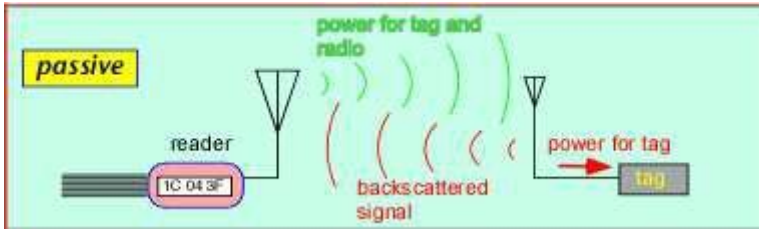
Mikrokontroler akan mengolah data tersebut untuk dijadikan *password* sebagai pengaman pintu. RFID memiliki empat frekuensi berdasarkan gelombang radionya yaitu *low frequency* (LF), *high frequency* (HF) untuk aplikasi jarak dekat (*proximity*), *ultra high frequency*(UHF) untuk aplikasi jarak jauh (*vicinity*) dan *microwave*. *Bandwith* frekuensi RFID dapat dilihat pada Tabel 2.1.

Tabel 2. 1 *Bandwith* Frekuensi RFID

No	Frekuensi RFID	Jenis Frekuensi	Manfaat
1	125KHz-134KHz	<i>Low Frequency</i>	Menandai hewan (<i>Animal tagging</i>)
2	13.56MHz	<i>High Frequency</i>	<i>Smart card</i>
3	860MHz-930MHz	<i>Ultra High Frequency</i>	Membuka otomatis bagasi, identifikasi suatu barang.
4	2.4GHz	<i>Micro-Wave</i>	Akses kontrol bagasi pesawat terbang.

2.2.2 *Tag Pasif (Passive Tags)*

RFID *tag* jenis ini tidak memiliki catu daya sendiri, catu dayanya diterima dari medan elektromagnetik yang dipancarkan oleh RFID *reader*. RFID *tag* akan aktif dan dapat mengirim data hanya ketika didekatkan dengan RFID *reader*. *Tag* pasif dapat beroperasi atau terbaca oleh RFID *reader* dengan jarak sekitar beberapa sentimeter sampai 10cm. Cara kerja RFID *tag* pasif dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Cara Kerja RFID Tag Pasif

Sumber <http://id.kmrfidtag.com/info/the-difference-between-active-and-passive-rfid-24833088.html>

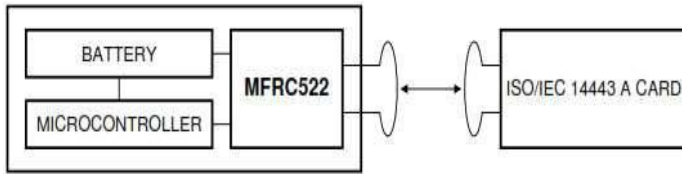
Berikut ini karakteristik dari RFID tag pasif yang dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Karakteristik RFID Tag Pasif

No	Karakteristik Tag Pasif
1	Tidak memiliki sumber tegangan sendiri.
2	Modulasi akan aktif ketika tag menerima gelombang elektromagnetik dari reader.
3	Jarak baca 0cm-10m.
4	Praktis dan mudah dibawa.

2.2.3 RFID Reader Membaca Data ID Dari Smart Card

RFID reader yang digunakan pada rangkaian adalah RFID reader dengan frekuensi 13.56 MHz. RFID reader 13.56MHz dapat digunakan untuk membaca RFID tag jenis *high frequency* (HF) yang digunakan sebagai *smart card*. RFID reader 13.56 MHz dapat membaca *smart card* jenis MIFARE ISO/IEC 14443. Data ID yang berupa nomor unik dari *smart card* akan dibaca oleh RFID reader kemudian dikirim ke mikrokontroler, baterai digunakan untuk suplai tegangan RFID dan mikrokontroler. Diagram blok sistem kerja dari mikrokontroler, RFID reader dan RFID tag (*smart card*) dapat dilihat pada Gambar 2.5.

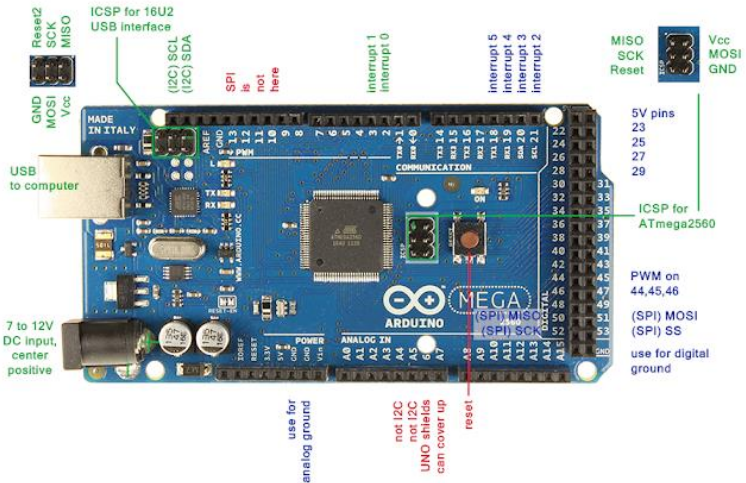


Gambar 2. 5 RFID Reader Membaca Data ID Dari Smart Card

2.3 Arduino Mega

Dalam buku “*Getting started with arduino*” Arduino adalah sebuah *platform* komputasi fisik yang *open source* pada *board input output* sederhana. Yang dimaksud dengan *platform* komputasi fisik di sini adalah sebuah sistem fisik yang interaktif dengan penggunaan *software* dan *hardware* yang dapat mendeteksi dan merespon situasi dan kondisi yang ada di dunia nyata.

Board Arduino 2560 adalah sebuah Board Arduino yang menggunakan ic Mikrokontroler AT 2560. Board ini memiliki Pin I/O yang relatif banyak, 54 digital Input / Output, 15 buah di antaranya dapat di gunakan sebagai output PWM, 16 buah analog Input, 4 UART. Arduino 2560 di lengkapi kristal 16. Mhz Untuk penggunaan relatif sederhana tinggal menghubungkan power dari USB ke PC / Laptop atau melalui Jack DC pakai adaptor 7-12 V DC. Secara fisik, ukuran Arduino Mega 2560 hampir kurang lebih 2 kali lebih besar dari Arduino Uno, ini untuk mengakomodasi lebih banyaknya pin Digital dan Analog pada board Arduino Mega 2560 tersebut. Cara penggunaan Arduino Mega 2560 ini sama persis dengan penggunaan Arduino Uno. Software IDE yang digunakan juga sama, hanya tinggal memilih *board* Arduino Mega 2560 pada pilihan *board*. Keluaran arus DC setiap pin *input* dan *output* adalah 20 mA. Keluaran arus DC dari pin 3,3 volt adalah 50 mA. Pada Arduino mega terdapat *memory flash* sebesar 256kb. Pada Arduino mega mempunyai *clock speed* sebesar 16 MHz. kelebihan pada Arduino mega dibandingkan dengan Arduino uno juga terletak pada banyaknya pin analoag, pada Arduino mega terdapat 16 pin analog dan pada Arduino uno hanya terdapat 4 pin analog. Arduino mega sangat disarankan jika digunakan untuk mengendalikan banyak alat seperti sensor maupun aktuator. Berikut ini penjelasan fungsi pin pada Arduino dapat dilihat pada Gambar 2.6. [3]



Gambar 2. 6 Board Arduino Mega 2560

Sumber [https://4.bp.blogspot.com/-](https://4.bp.blogspot.com/-rvkJgy9KFIk/WLN35RdFGcI/AAAAAAAAAUY/NLxrm1fPzQU6T-Vahlr3APOsP0Af4vTzQCLcB/s640/Arduino-Mega-2560-Pin.png)

[rvkJgy9KFIk/WLN35RdFGcI/AAAAAAAAAUY/NLxrm1fPzQU6T-Vahlr3APOsP0Af4vTzQCLcB/s640/Arduino-Mega-2560-Pin.png](https://4.bp.blogspot.com/-rvkJgy9KFIk/WLN35RdFGcI/AAAAAAAAAUY/NLxrm1fPzQU6T-Vahlr3APOsP0Af4vTzQCLcB/s640/Arduino-Mega-2560-Pin.png)

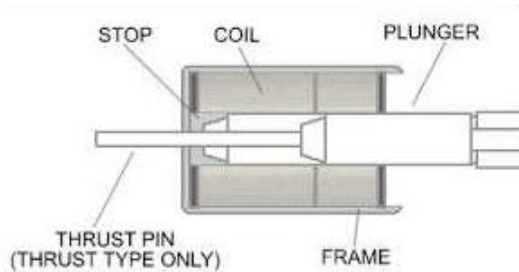
2.4 Solenoid DC

Solenoid adalah aktuator yang mampu melakukan gerakan linier yaitu gerakan lurus menarik atau mendorong. Solenoid DC dapat bekerja. Secara elektro mekanis dengan memberikan sumber tegangan, maka solenoid dapat menghasilkan gaya yang linier dapat dilihat pada Gambar 2.7 dan Gambar 2.8. [4]



Gambar 2. 7 Solenoid DC

Sumber <https://www.cytron.io/p-12vdc-solenoid-door-lock>

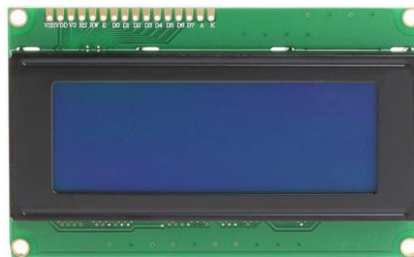


Gambar 2. 8 Bagian Solenoid DC

Sumber <https://www.cytron.io/p-12vdc-solenoid-door-lock>

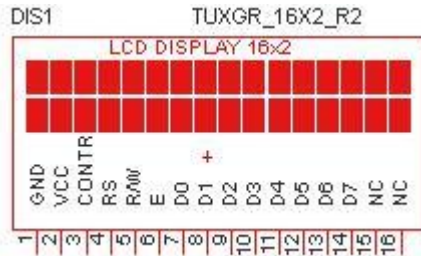
2.5 *Liquid Crystal Display (LCD)*

Liquid crystal display (LCD) adalah komponen yang dapat menampilkan tulisan dengan memanfaatkan kristal cair, salah satu jenisnya adalah LCD 20x4 yang memiliki dua baris setiap baris terdiri dari enam belas karakter. Gambar LCD 20x4 dapat dilihat pada Gambar 2.9 dan Gambar 2.10..



Gambar 2. 9 LCD 20x4

Sumber <https://www.indiamart.com/proddetail/lcd-display-20x4-19702905491.html>



Gambar 2. 10 Skema LCD 20x4

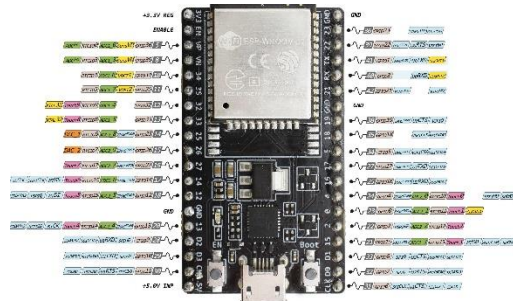
LCD ini memiliki 16 *pin* dengan fungsi *pin* masing – masing diperlihatkan pada Tabel 2.3.

Tabel 2. 3 Pin Pada LCD Display

No. Pin	Nama Pin	I/O	Keterangan
1	<i>GROUND</i>	<i>Power</i>	Catu daya, ground (0V)
2	VCC	<i>Power</i>	Catu daya positif
3	CONTR	<i>Power</i>	Pengatur kontras. Menurut datasheet, pin ini perlu dihubungkan dengan pin VSS melalui resistor 5kΩ. Namun, dalam praktik, resistor yang digunakan sekitar 2,2kΩ.
4	RS	<i>Input</i>	<i>Register Select</i> RS= <i>HIGH</i> : untuk mengirim data RS= <i>LOW</i> : Untuk mengirim instruksi
5	R/W	<i>Input</i>	<i>Read/Write control bus</i> R/W= <i>HIGH</i> : mode untuk membaca data

2.6 ESP32S

Modul ESP32S merupakan modul WiFi yang berfungsi mengkoneksikan antara program Arduino dengan database maupun web. Modul ESP32S dapat berdiri sendiri maupun dihubungkan secara serial dengan Arduino. Untuk pemrograman ESP32S sendiri menggunakan Arduino IDE.



Gambar 2.11 ESP32

Sumber

<https://i.pinimg.com/originals/c6/57/83/c657835e84aaf91832a770ea0d7d0767.jpg>

Pada modul ESP32S hanya memiliki tegangan terminal 3,3 Volt. ESP32 merupakan penerus dari module ESP8266. Pada ESP32 terdapat inti CPU serta Wi-Fi yang lebih cepat, GPIO yang lebih, dan mendukung Bluetooth 4.2 konsumsi daya yang rendah.

2.7 RTC DS3231

Modul RTC DS3231 merupakan salah satu jenis module yang dimana berfungsi sebagai RTC (*Real Time Clock*).



Gambar 2.12 Modul RTC DS3231

Sumber https://4.bp.blogspot.com/-oIxSvTkVrtU/V_snRXWeNI/AAAAAAAAAAN0/LJTB2MhMYfkSS-juSycVKj1xz0QULi0lwCLcB/s1600/RTC-DS3231-TOP.png

Interface atau antarmuka untuk mengakses modul ini yaitu menggunakan i2c atau two wire (SDA dan SCL). Sehingga apabila diakses menggunakan mikrokontroler misal Arduino Mega pin yang dibutuhkan 2 pin saja dan 2 pin power. Module DS3231 RTC ini pada umumnya sudah tersedia dengan battery CR2032 3V yang berfungsi sebagai back up RTC apabila catudaya utama mati.

2.8 Modul Relay

Relay adalah Saklar yang dioperasikan secara listrik dan merupakan komponen Elektromekanikal yang terdiri dari 2 bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat Kontak Saklar/Switch). Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Modul relay dapat dilihat pada gambar 2.14.



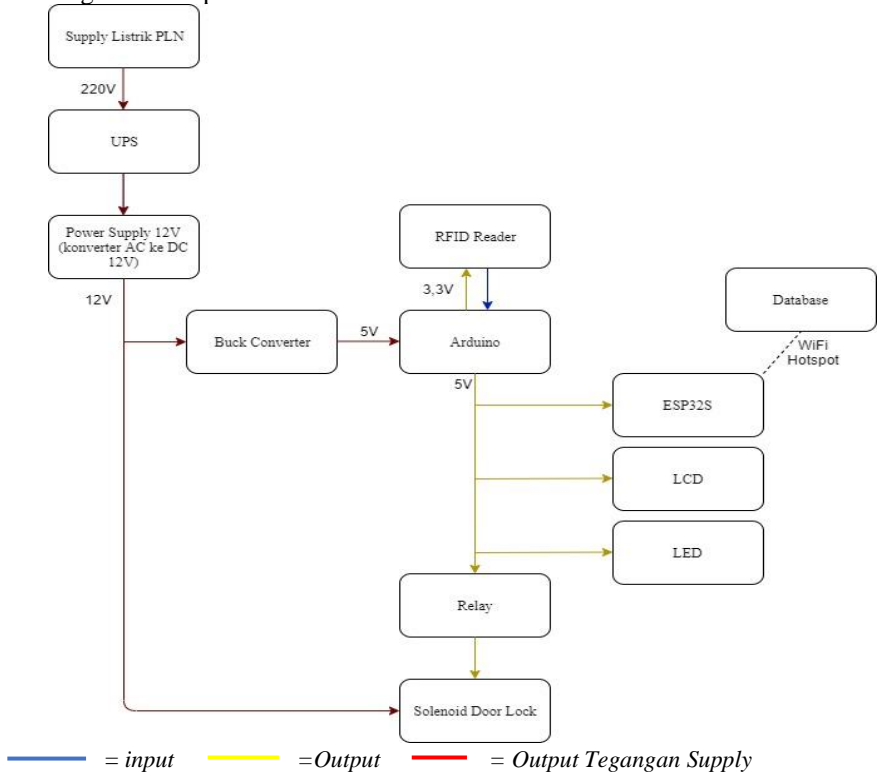
Gambar 2.14 Modul Relay

Sumber https://ecs7.tokopedia.net/img/cache/700/product-1/2018/9/15/9862771/9862771_fc3fb0e4-d427-474e-8bba-53295c3d56b6_600_600.jpg

Modul relay ini dapat digunakan sebagai switch untuk menjalankan berbagai peralatan elektronik. Kendali ON / OFF switch (relay), sepenuhnya ditentukan oleh nilai output sensor, yang setelah diproses Mikrokontroler akan menghasilkan perintah kepada relay untuk melakukan fungsi ON / OFF.

BAB III PERENCANAAN DAN IMPLEMENTASI

Alat pengaman sekaligus alat presensi yang dirancang pada Proyek Akhir ini memiliki fasilitas-fasilitas penunjang yang dapat mempermudah pengguna untuk mengakses ruangan dengan lebih mudah. Selain dari pada itu pengaman ini juga dilengkapi perangkat komunikasi berbasis ESP32 yang berfungsi untuk memberikan informasi kepada pengguna melalui *database* yang nantinya diakses untuk memantau kehadiran (presensi) serta bersifat *realtime*. Alat dan bahan yang digunakan dalam penelitian, secara umum desain seperti diagram blok pada Gambar 3.1.



Gambar 3. 1 Blok Fungsional Sistem

Desain perangkat keras pada Gambar 3.1 memiliki sensor RFID *reader* yang berfungsi untuk membaca data ID dari E-KTP. Arduino MEGA berfungsi untuk mengakses data dari sensor RFID *reader*. LCD 20x4 berfungsi untuk menampilkan karakter sesuai program yang diberikan oleh Arduino Mega. ESP32 berfungsi untuk mengkoneksikan data dari Arduino ke database.

Arduino berfungsi sebagai pusat kendali rangkaian yang akan mengaktifkan *relay* ketika rfid yang dipasangkan pada Arduino membaca E-KTP yang sudah terdaftar sehingga solenoid aktif dan pintu dapat dibuka. Solenoid *Door Lock* digunakan untuk mengunci pintu. *Relay* digunakan untuk *on off* solenoid *Door Lock*.

Led untuk indikator akses diterima atau tidak. Nyala led hijau menandakan E-KTP akses diterima atau sudah terdaftar. Nyala led merah menandakan E-KTP akses ditolak atau belum terdaftar.

Relay yang berguna untuk *switch* otomatis sumber yang digunakan untuk suplai perangkat dimana saat listrik dari PLN mati maka akan tetap hidup karena mendapat suplai dari UPS.

Bab ini membahas perancangan dan realisasi dari perangkat keras dan perangkat lunak Perancangan sistem pada Proyek Akhir ini fokus pada :

1. Perancangan Mekanik
2. Perancangan Elektronik
3. Perancangan *Software*

3.1 Perancangan Mekanik

Mekanik pengaman dan presensi otomatis dirancang dan dibuat menggunakan box seperti pada Gambar 3.2 sampai Gambar 3.5. Pada sisi dalam alat bagian bawah dipasang seperti pada Gambar 3.6.



Gambar 3. 2 Kotak Alat Tampak Depan

Alat memiliki satu pintu yaitu pintu utama dibagian depan. Pada kotak bagian depan dipasang *Liquid Crystal Display* (LCD) 20x4, indikator LED seperti Gambar 3.2.



Gambar 3. 3 Kotak Alat Tampak Belakang

Alat pembuka pintu otomatis pada tampak belakang dirancang dan dibuat menggunakan box hitam dengan tebal 4mm. Pada bagian belakang ini dibuat berlubang agar dapat memasang relay dengan solenoid yang terpasang pada pintu. Gambar tersebut dapat dilihat pada Gambar 3.3.



Gambar 3.4 Kotak Alat Tampak Kiri

Kotak alat pada tampak kiri terdapat dua saklar untuk *supply* alat *supply* dari AC dan DC dan *input* kabel *power* seperti pada Gambar 3.4 diatas.



Gambar 3.5 Kotak Alat Tampak Kanan

Kotak alat tampak kanan dirancang dan dibuat menggunakan bahan plastik dengan tebal 4mm seperti pada Gambar 3.5.



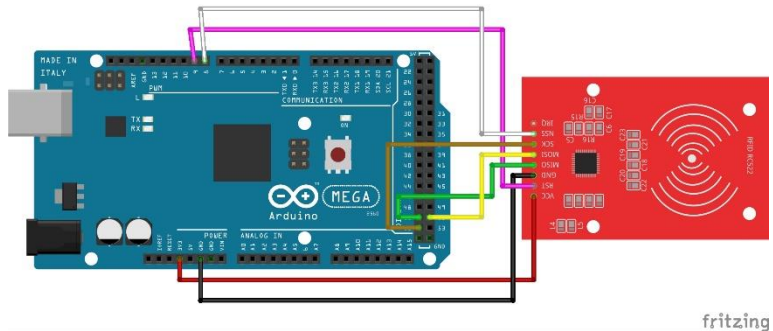
Gambar 3.6 Kotak Alat Tampak Dalam

3.2 Perancangan Perangkat Elektrik

Pada perancangan perangkat elektrik akan dibahas tentang perancangan rangkaian dan komponen yang digunakan pada alat.

3.2.1 Wiring Modul RFID Reader RS522

Pada Gambar 3.7 merupakan *wiring* atau pemasangan tiap *pin* antara RFID Reader dan Arduino Mega agar RFID Reader dapat dikontrol oleh Arduino Mega.



Gambar 3.7 Skema Modul RFID Reader MFRC522 dengan Arduino

Modul RFID reader ini berfungsi untuk membaca data atau nomor ID pada E-KTP yang kemudian mengirim data tersebut ke Arduino MEGA. Pemasangan komponen modul RFID dengan *port* Arduino MEGA dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Sambungan *Pin* RFID Reader Ke Arduino

Modul RFID-RC522	Port Arduino Mega
RST	Digital Pin 9
SS	Digital Pin 8
MOSI	Digital Pin 51
MISO	Digital Pin 50
SCK	Digital Pin 52

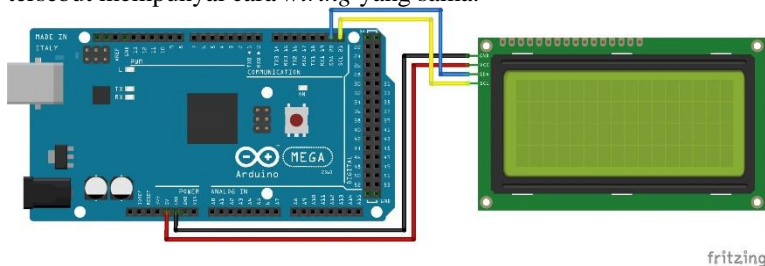
RFID reader akan mengeluarkan gelombang radio dan menginduksi RFID *tag*. Gelombang induksi tersebut berisi nomor ID dan jika dikenali oleh RFID *tag*, maka memori RFID *tag* akan

mengirimkan kode yang terdapat di memori ID *chip* melalui antena yang terpasang di RFID *tag* ke RFID *reader*. Selanjutnya RFID *reader* akan meneruskan kode yang diterima ke Arduino MEGA.

Arduino MEGA akan melaksanakan instruksi yang telah diberikan, jika kode tersebut sesuai maka otomatis akan mengaktifkan *relay* sehingga solenoid aktif dan membuka pintu, namun jika kode atau nomer ID tidak sesuai maka *relay* tidak aktif solenoid akan *off* dan pintu tidak terbuka.

3.2.2 Wiring Modul LCD 20x4 Dengan I2c

Pada Gambar 3.8 merupakan *wiring* atau pemasangan tiap *pin* antara modul LCD 20x4 dengan I2c dan Arduino MEGA agar modul LCD 20x4 dengan I2c dapat dikontrol oleh Arduino MEGA. Dalam simulasi *Eagle* dan *Fritzing* disini LCD yang digunakan 20x4 karena tidak adanya library dari LCD 20x4. Dalam hal *wiring* kedua LCD tersebut mempunyai cara *wiring* yang sama.



Gambar 3.8 Modul LCD 20x4 Dengan Arduino

Modul LCD 20x4 digunakan sebagai perintah atau memberikan instruksi dengan cara menampilkan tulisan untuk mendekati kartu identitas E-KTP ke RFID *reader*. Pemasangan *port* LCD ke *pin* Arduino MEGA ditabelkan pada Tabel 3.2.

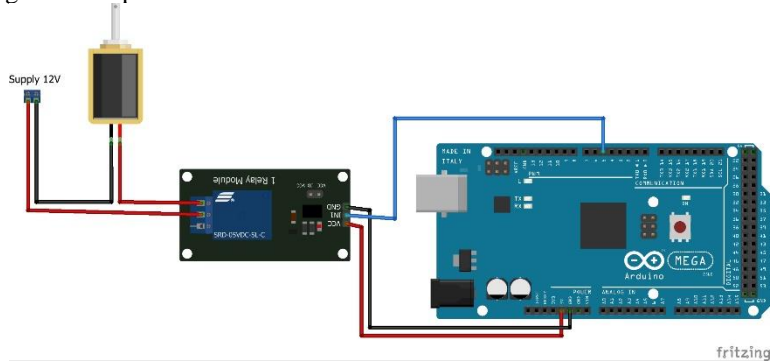
Pada modul LCD 20x4 tersebut menggunakan I2c sebagai modul atau alat yang berfungsi untuk mengurangi *pin* LCD ke arduino, sehingga memudahkan dalam membuat rangkaian.

Tabel 3.2 Sambungan *Pin* LCD Ke Arduino

Modul LCD Display 20x4	Port Arduino MEGA
SDA	Pin SDA
SCL	Pin SCL

3.2.3 Rangkaian *Relay* Dan Solenoid

Relay digunakan sebagai kendali atau kontrol pada solenoid agar sesuai dengan *input* yang diberikan yaitu untuk membuka dan menutup pintu. Pada sistem ini akan digunakan beberapa *pin* Arduino dengan rancangan sesuai pada Gambar 3.9.



Gambar 3.9 Rangkaian *Relay* dan Solenoid Dengan Arduino

Solenoid yang digunakan yaitu solenoid 12V yang berfungsi sebagai aktuator untuk membuka dan menutup pintu, *relay* berfungsi sebagai saklar untuk mengaktifkan (*On*) dan mematikan solenoid (*off*). Dengan VCC 12V dari *supply* diberikan ke *pin* NO *relay* dan *ground* pada solenoid ke *ground supply*. Pemasangan komponen *relay*, solenoid dengan *port* Arduino MEGA dapat dilihat pada Tabel 3.3 dan 3.4.

Tabel 3.3 Sambungan *Pin Relay* Ke Arduino

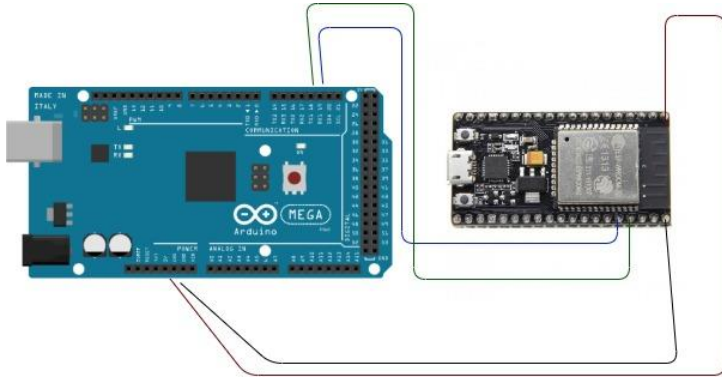
<i>Relay</i>	<i>Port</i> Arduino MEGA
<i>Input</i>	Digital Pin 5

Tabel 3.4 Sambungan *Pin Solenoid* Ke *Relay*

<i>Relay</i>	Solenoid	Suplai 12V
COM	-	VCC
-	GROUND	GROUND
NO	VCC	-

3.2.4 Wiring ESP32

ESP32 digunakan sebagai modul untuk mengirimkan data dari Arduino ke *Database*. Untuk menghubungkan Arduino dengan ESP32 dibutuhkan komunikasi secara serial melalui TX dan RX dari kedua modul tersebut. Berikut rancangan modul dapat dilihat pada Gambar 3.10.



Gambar 3.10 Rangkaian ESP32 dengan Arduino

Untuk melakukan komunikasi serial antara Arduino dengan ESP32, TX dan RX pada Arduino dengan ESP32 dipasangkan secara berkebalikan. TX pada Arduino disambungkan pada RX ESP32 dan RX Arduino disambungkan pada TX ESP32. Sebelum mengkoneksikan antara keduanya dibutuhkan shift register untuk menurunkan tegangan 5 volt dari Arduino ke 3,3 volt ESP32. Pemasangan ESP32 dengan *port* Arduino MEGA dapat dilihat pada Tabel 3.5.

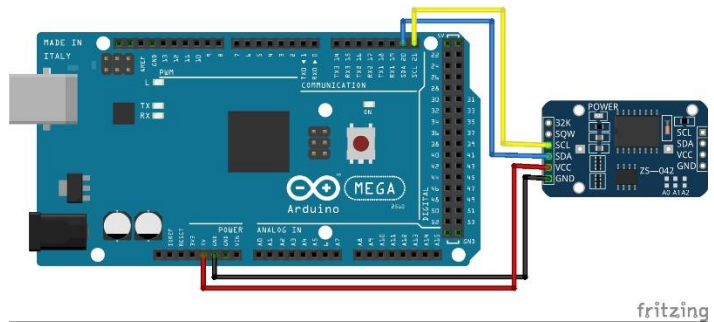
Tabel 3.5 Sambungan *Pin Relay* Ke Arduino

Arduino MEGA	ESP32
5V	3,3V
GND	GND
TX	RX
RX	TX

3.2.5 Wiring Real Time Clock (RTC) DS3231

Real Time Clock (RTC) DS3231 merupakan modul yang berfungsi untuk mendapatkan data waktu secara *real time*. Pada modul

RTC ini terdapat terminal untuk memasang baterai yang berfungsi sebagai *supply* cadangan ketika listrik padam agar tetap memperoleh waktu secara *real time*. Pin SDA dan SCL pada RTC disambungkan pada pin SDA (20) dan SCL (21) Arduino Mega. Berikut skema dapat dilihat pada Gambar 3.11.



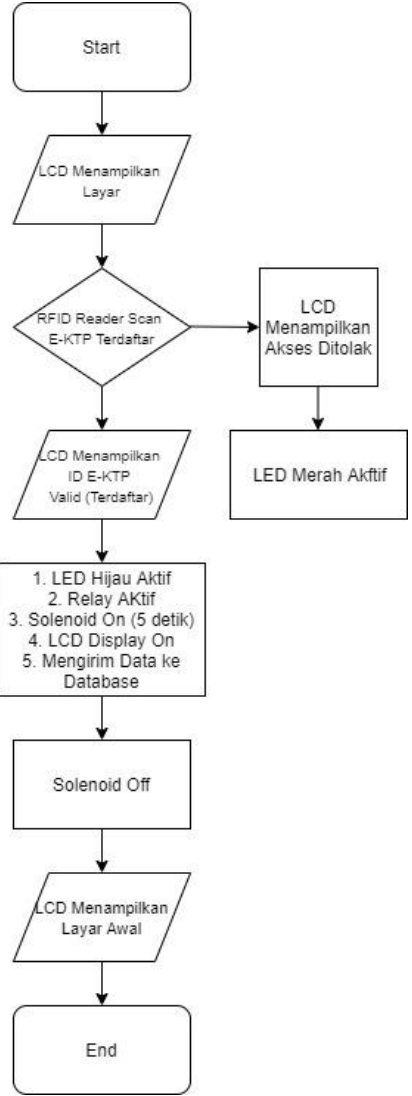
Gambar 3.11 Rangkaian RTC

3.3 Perancangan *Software*.

Pada sub bab ini menjelaskan keseluruhan *flowchart* keseluruhan sistem dan masing masing perintah setiap komponen yang digunakan pada alat ini yang terdiri dari RFID Reader, modul LCD *display*, *relay*, solenoid, led, buzzer.

3.3.1 *Flowchart* Keseluruhan

Pada Gambar 3.12 merupakan *flowchart* keseluruhan sistem. Pada alat ini dimana sistem berjalan dari awal hingga akhir dari alat dihidupkan. Mulai dari LCD menampilkan layar awal, kemudian dilanjutkan bagaimana program jika RFID *reader* membaca E-KTP yang sudah terdaftar maupun belum terdaftar. Selanjutnya proses bagaimana jika RFID *reader* membaca E-KTP yang sudah terdaftar ataupun yang belum terdaftar hingga LCD menampilkan keadaan layar awal lagi. Berikut pada gambar 3.12 akan dijelaskan runtutan *flowchart* dari keseluruhan alat.



Gambar 3.12 Flowchart Keseluruhan Alat

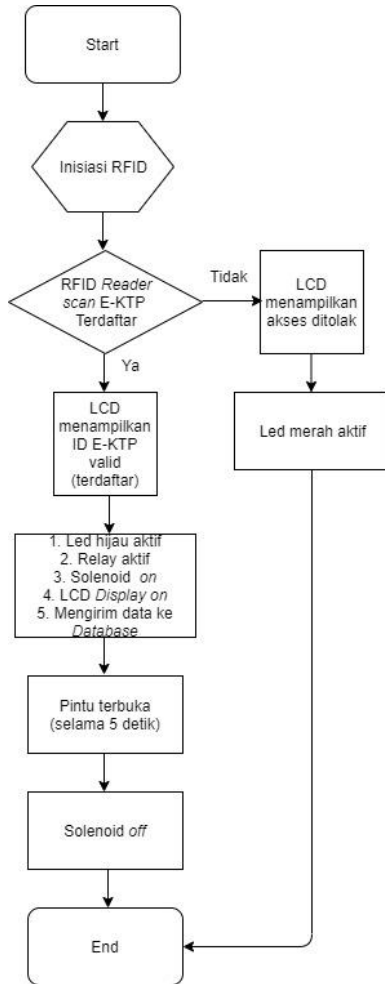
Bagan alir pada perancangan ini merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur

di dalam setiap sistemnya. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambaranya. Dalam perancangan *software* ini terdiri dari 7 bagian :

1. *Flowchart* membuka pintu menggunakan E-KTP
2. *Flowchart LCD Display 20x4*
3. *Flowchart* Arduino mengirim data presensi pada Excel
4. *Flowchart* mengaktifkan led merah
5. *Flowchart* mengaktifkan led hijau
6. *Flowchart* mengaktifkan led buzzer

3.3.2 *Flowchart* Membuka Pintu Menggunakan E-KTP

Flowchart cara kerja RFID *Reader* membaca E-KTP berbasis Arduino MEGA dimana RFID *Reader* dikontrol oleh Arduino Mega untuk membaca ID E-KTP dan id yang sudah terdaftar akan mendapatkan izin untuk membuka pintu dan jika ID yang tidak terdaftar maka tidak akan mendapatkan izin untuk membuka pintu. Mulai dari LCD menampilkan layar awal, kemudian dilanjutkan bagaimana program jika RFID reader membaca E-KTP yang sudah terdaftar maupun belum terdaftar. Selanjutnya proses bagaimana jika RFID reader membaca E-KTP yang sudah terdaftar ataupun yang belum terdaftar hingga LCD menampilkan keadaan layar awal lagi. Jika RFID *reader* membaca E-KTP yang sudah terdaftar maka relay akan aktif dan solenoid akan terbuka selebihnya led hijau akan menyala dan LCD akan menampilkan nama dari E-KTP yang sudah terdaftar. Jika RFID *reader* membaca E-KTP yang belum terdaftar maka LCD akan menampilkan tulisan akses ditolak dan relay tidak akan aktif begitu juga solenoid tidak akan terbuka. *Flowchart* dapat dilihat pada Gambar 3.13.



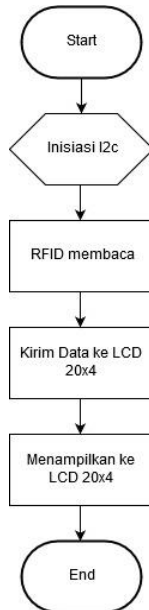
Gambar 3. 13 Flowchart Pembacaan E-KTP Dengan RFID Reader

1. Start
Langkah pertama untuk mengoperasikan alat yaitu dengan memberikan tegangan pada sistem atau rangkaian.
2. Inialisasi RFID
Inialisai kepada mikro Arduino agar dapat menjalankan sensor RFID Reader .

3. *RFID Reader scan E-KTP*
Setelah sistem aktif Arduino MEGA akan melakukan fungsinya sebagai kontrol dari semua *input* dan *output*. Arduino MEGA mengaktifkan *RFID Reader* dan akan membaca kartu yang terdaftar dan tidak terdaftar.
4. *Relay on led hijau on*
Jika kartu terdaftar maka Arduino akan menyalakan led hijau, *relay*.
5. *Mengirim data ke database*
Jika kartu terdaftar maka data dari Arduino akan dikirimkan ke ESP32, kemudian oleh ESP32 akan diteruskan ke *database* dengan jaringan WiFi
6. *Led merah on*
Jika kartu tidak terdaftar maka Arduino akan menyalakan led merah.
7. *Solenoid on*
Solenoid akan *on* jika kartu terdaftar.
8. *Delay 5 detik*
Solenoid akan terbuka selama 5 detik.
9. *Solenoid off*
Solenoid akan tertutup
10. *Return*
Program akan kembali setelah *start*

3.3.3 Flowchart LCD Display 20x4

Pada *flowchart* ini dimana *LCD Display 20x4* sebagai tampilan *hardware* untuk menampilkan akses diterima dan akses ditolak oleh *RFID Reader*. *Flowchart* dapat dilihat pada Gambar 3.14.

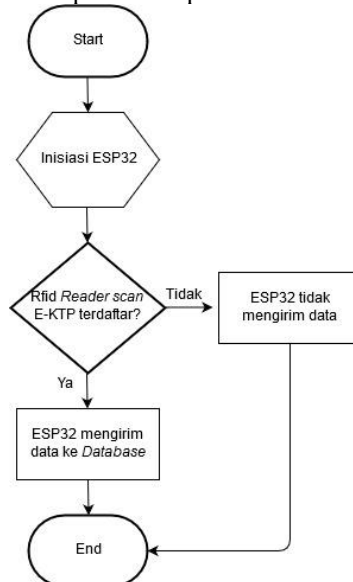


Gambar 3.14 Flowchart Menampilkan Data Pada LCD Display

1. Start.
Langkah pertama untuk mengoperasikan alat yaitu dengan memberikan tegangan pada sistem atau rangkaian.
2. Inialisasi I2c.
Inialisasi kepada mikro Arduino agar dapat menjalankan sensor LCD Display..
3. RFID membaca
RFID akan membaca kartu yang terdaftar dan tidak terdaftar.
4. Kirim data ke 20x4
Setelah RFID membaca kartu maka data akan dikirim ke LCD Display 20x4
5. Menampilkan ke LCD 20x4
Setelah data terkirim data langsung akan ditampilkan ke LCD Display 20x4.
6. Return.
Menampilkan layar LCD tampilan awal

3.3.4 *Flowchart Pengiriman Data ke Database*

Flowchart kali ini menjelaskan bagaimana data dari Arduino bisa dikirim sampai ke *Database* melalui komunikasi oleh ESP32. *Flowchart* dapat dilihat pada Gambar



Gambar 3.15 *Flowchart Pengiriman Data ke Database*

1. Start.
Langkah pertama untuk mengoperasikan alat yaitu dengan memberikan tegangan pada sistem atau rangkaian.
2. Inialisasi ESP32.
Inialisai kepada mikro Arduino agar dapat menjalankan ESP32..
3. RFID membaca.
RFID akan membaca kartu yang terdaftar dan tidak terdaftar.
4. Kirim data ke database.

Setelah RFID membaca kartu yang terdaftar maka data akan dikirim pada *database*, ketika membaca kartu yang tidak terdaftar maka tidak akan mengirim data ke *database*.

5. Menampilkan ke LCD 20x4
Setelah data terkirim data langsung akan ditampilkan ke LCD *Display* 20x4.

BAB IV

PENGUJIAN DAN ANALISIS

Pengujian dan analisis pada bab ini dilakukan untuk mengetahui performa sistem kekurangan dan kelebihan serta bagaimana alat itu bekerja secara *realtime* atau nyata yang terdiri dari

1. Pengukuran dan analisa rangkaian elektronik (cara pengukuran, rangkaian pengukuran, hasil pengukuran)
2. Pengukuran dan analisa perangkat lunak (cara menjalankan, tampilan, hasil tampilan).

4.1 Pengujian Dan Analisa Perangkat Elektrik

Pada sub bab ini membahas semua komponen yang digunakan pada perangkat ini dimana masing masing komponen dapat melakukan perintah sesuai dengan sistem keseluruhan yang di jelaskan pada *flowchart* sistem keseluruhan pada Gambar 3.11.

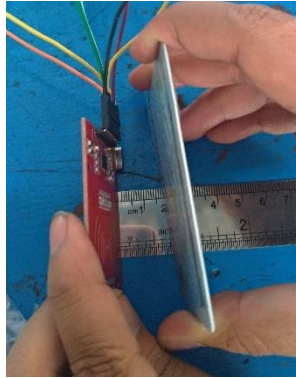
4.1.1 Pengujian Arduino Mega

Arduino Mega digunakan untuk mengendalikan *input* dan *output* pada alat, sehingga Arduino memerlukan *supply* tegangan yang sesuai.

Pengukuran tegangan *input* pada mikrokontroler Mega menggunakan *Multimeter* analog adalah 12V. Dari pengukuran tegangan *input* tersebut menunjukkan bahwa hasil pengukuran sesuai dengan *datasheet* 5V dan bisa sampai 12V, Arduino Mega membutuhkan tegangan operasional sebesar 5V – 12V . Arduino Mega berfungsi sebagai pusat kendali *input/output* pada alat.

4.1.2 Pengujian Jarak Sensor RFID Reader Dengan E-KTP

Pada Gambar 4.1 merupakan pengujian jarak pembacaan sensor RFID *reader* dengan E-KTP dilakukan menggunakan RFID *reader* tanpa pembatas. Pengukuran jarak E-KTP dengan RFID *reader* bertujuan untuk mengetahui jarak RFID *reader* dapat membaca ID pada E-KTP. Pada hasil pengukuran jarak E-KTP dan RFID *Reader* dapat dilihat pada Tabel 4.1.



Gambar 4.1 Uji Coba Pembacaan E-KTP Dengan Menghadap Sensor RFID Reader

Tabel 4.1 Pengambilan Data Jarak E-KTP dengan RFID Reader

No	Jarak	Keterangan
1	0 cm	Terbaca
2	0,2 cm	Terbaca
3	0,4 cm	Terbaca
4	0,6 cm	Terbaca
5	0,8 cm	Terbaca
6	1 cm	Terbaca
7	1,2 cm	Terbaca
8	1,4 cm	Terbaca
9	1,6 cm	Terbaca
10	1,8 cm	Terbaca
11	2 cm	Terbaca
12	2,2 cm	Tidak Terbaca
13	2,4 cm	Tidak Terbaca

Pada Gambar 4.1 merupakan pengujian jarak pembacaan sensor RFID reader dengan E-KTP dilakukan menggunakan RFID reader tanpa pembatas. Pengukuran jarak E-KTP dengan RFID reader bertujuan untuk mengetahui jarak RFID reader dapat membaca ID pada E-KTP Pengujian Dan Analisa Perangkat *Software*. Selanjutnya pada Gambar 4.2 merupakan pengujian jarak pembacaan sensor RFID setelah

dimasukkan kedalam *box* hitam dengan pengujian pertama menggunakan jarak 0,1 cm.



Gambar 4.2 Pengujian Jarak E-KTP Dengan Sensor RFID Dalam *Box*

Tabel 4.2 Pengambilan Data Jarak 0,1 cm E-KTP Dengan Sensor RFID Dalam *Box*

No	Jarak	Keterangan
1	0,1 cm	Terbaca
2	0,1 cm	Terbaca
3	0,1 cm	Terbaca
4	0,1 cm	Terbaca
5	0,1 cm	Terbaca
6	0,1 cm	Terbaca
7	0,1 cm	Terbaca
8	0,1 cm	Terbaca
9	0,1 cm	Terbaca
10	0,1 cm	Terbaca

Pada pengujian pembacaan antara sensor RFID dengan E-KTP dengan jarak 0,1 cm , sensor dapat membaca E-KTP dengan percobaan sepuluh kali berhasil. Selanjutnya pada tabel 4.3 pengujian jarak E-KTP

dengan sensor RFID dalam *box* menggunakan jarak 0,5 cm dengan sepuluh kali percobaan.

Tabel 4.3 Pengambilan Data Jarak 0,5 cm E-KTP Dengan Sensor RFID Dalam *Box*

No	Jarak	Keterangan
1	0,5 cm	Terbaca
2	0,5 cm	Terbaca
3	0,5 cm	Terbaca
4	0,5 cm	Terbaca
5	0,5 cm	Terbaca
6	0,5 cm	Terbaca
7	0,5 cm	Terbaca
8	0,5 cm	Terbaca
9	0,5 cm	Terbaca
10	0,5 cm	Terbaca

Pada pengujian pembacaan antara sensor RFID dengan E-KTP dengan jarak 0,5 cm , sensor dapat membaca E-KTP dengan percobaan sepuluh kali berhasil. Selanjutnya pada tabel 4.4 pengujian jarak E-KTP dengan sensor RFID dalam *box* menggunakan jarak 1 cm dengan sepuluh kali percobaan.

Tabel 4.4 Pengambilan Data Jarak 1 cm E-KTP Dengan Sensor RFID Dalam *Box*

No	Jarak	Keterangan
1	1 cm	Terbaca
2	1 cm	Terbaca
3	1 cm	Terbaca
4	1 cm	Terbaca
5	1 cm	Terbaca
6	1 cm	Terbaca
7	1 cm	Terbaca
8	1 cm	Terbaca
9	1 cm	Terbaca
10	1 cm	Terbaca

Pada pengujian pembacaan antara sensor RFID dengan E-KTP dengan jarak 1 cm , sensor dapat membaca E-KTP dengan percobaan sepuluh kali berhasil. Selanjutnya pada tabel 4.5 pengujian jarak E-KTP dengan sensor RFID dalam *box* menggunakan jarak 1,5 cm dengan sepuluh kali percobaan.

Tabel 4.5 Pengambilan Data Jarak 1,5 cm E-KTP Dengan Sensor RFID Dalam *Box*

No	Jarak	Keterangan
1	1,5 cm	Terbaca
2	1,5 cm	Terbaca
3	1,5 cm	Terbaca
4	1,5 cm	Terbaca
5	1,5 cm	Terbaca
6	1,5 cm	Terbaca
7	1,5 cm	Tidak Terbaca
8	1,5 cm	Tidak Terbaca
9	1,5 cm	Tidak Terbaca
10	1,5 cm	Tidak Terbaca

Pada pengujian pembacaan antara sensor RFID dengan E-KTP dengan jarak 1,5 cm , sensor dapat membaca E-KTP dengan percobaan 6 kali terbaca dan 4 kali tidak terbaca. Untuk pengujian kali ini sensor 4 kali tidak bisa membaca E-KTP dikarenakan oleh gerakan tangan yang tidak sengaja menjauh dari sensor E-KTP. Selanjutnya pada tabel 4.6 pengujian jarak E-KTP dengan sensor RFID dalam *box* menggunakan jarak 2 cm dengan sepuluh kali percobaan.

Tabel 4.6 Pengambilan Data Jarak 2 cm E-KTP Dengan Sensor RFID Dalam *Box*

No	Jarak	Keterangan
1	2 cm	Tidak Terbaca
2	2 cm	Tidak Terbaca
3	2 cm	Tidak Terbaca
4	2 cm	Tidak Terbaca

No	Jarak	Keterangan
5	2 cm	Tidak Terbaca
6	2 cm	Tidak Terbaca
7	2 cm	Tidak Terbaca
8	2 cm	Tidak Terbaca
9	2 cm	Tidak Terbaca
10	2 cm	Tidak Terbaca

Pada pengujian pembacaan antara sensor RFID dengan E-KTP dengan jarak 2 cm , sensor sudah tidak dapat membaca E-KTP dengan percobaan sepuluh kali tidak terbaca.

Pada sub bab ini menjelaskan mengenai *software* untuk mengontrol setiap komponen yang berada pada alat ini.

4.1.3 Perangkat Lunak (*Software*) Pengaman dan Presensi

Menggunakan E-KTP Berbasis Arduino Mega

Pada Gambar 4.3 merupakan *software* yang digunakan pada alat pengaman pintu sekaligus menggunakan E-KTP adalah *software* Arduino IDE berfungsi untuk memasukkan program pada mikrokontroler Arduino Mega.

```
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>
#include "RTClib.h"

#define SS_PIN 8
#define RST_PIN 9

RTC_DS3231 rtc;

char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
LiquidCrystal_I2C lcd(0x3F, 20, 4);

MFRC522 mfrc522(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte card_ID[7]; //card UID size 4byte
String Name1 = "04 78 2B 7A 14 2A 80"; //kartu 28 TAMA
String Name2 = "04 45 8E 3A 81 5B 80"; //kartu 29 ALFAN
String Name3 = "04 87 50 8A FE 4D 80"; //kartu 30 MOH. AJI SURYA MARARDIGA
String Name4 = "04 40 5A 32 4B 57 80"; //kartu 31 FERIAN BAGUS KUSUMA
String Name5 = "04 38 70 7A 18 5B 80"; //kartu 32 JUAN ALFREDO KURNIAWAN
String Name6 = "04 73 85 2A 9C 56 80"; //kartu 33 PANDU DINATA AL AMIRI
String Name7 = "04 52 1A DA 70 5B 80"; //kartu 34 ARGON LUTHEFAN GHAZ
String Name8 = "04 4F 63 A2 F4 57 80"; //kartu 35 FARIZ AKBAR FIRMANSYAH
```

Gambar 4.3 *Software* Arduino IDE Pada Alat Pengaman Ruangan

4.1.4 Pengujian Sistem Otomasi Pintu Menggunakan E-KTP

Disini akan menjelaskan pengujian *software* atau program serta tampilan dari pengaturan setiap komponen telah diprogram oleh *software* Arduino IDE untuk menjalankan setiap tugasnya.

4.1.4.1 Pengujian LCD 20x4 I2c Display

Berikut ini merupakan program Arduino untuk menampilkan karakter pada LCD sesuai dengan perintah dari RFID Reader. Dapat dilihat pada Gambar 4.4.

```
pinMode(LED_R, OUTPUT);
pinMode(LED_G, OUTPUT);
pinMode(Relay, OUTPUT);
lcd.setCursor(4, 0);
lcd.print("Laboratorium");
lcd.setCursor(0, 1);
lcd.print("Elektronika Terapan");
lcd.setCursor(1, 2);
lcd.print("Silahkan Scan Kartu");
lcd.setCursor(1, 3);
lcd.print("Anda :)");
delay(200);
digitalWrite (Relay, HIGH);
delay(200);
}
```

Gambar 4.4 Program Menampilkan “Laboratorium Elektronika Terapan Silahkan Scan Kartu Anda” Pada LCD

Hasil program pada Gambar 4.4 menampilkan tulisan Laboratorium Elektronika Terapan Silahkan Scan Kartu Anda pada LCD Display dapat dilihat pada Gambar 4.5.



Gambar 4.5 LCD Menampilkan Karakter ID Yang Terdaftar

```
if (NumbCard[j] == 1 && statu[s] == 0) {  
    statu[s] = 1;  
    lcd.clear();  
    lcd.setCursor(4, 0);  
    lcd.print("Laboratorium");  
    lcd.setCursor(0, 1);  
    lcd.print("Elektronika Terapan");  
    lcd.setCursor(4, 2);  
    lcd.print("Welcome ");  
    lcd.setCursor(0, 3);  
    lcd.print(Name);  
    delay(2000);  
}
```

Gambar 4.6 Program Untuk Menampilkan Karakter Pada LCD Nama E-KTP Pengakses

Hasil program pada Gambar 4.6 untuk menampilkan karakter nama E-KTP pengakses pada LCD dapat dilihat pada Gambar 4.7.



Gambar 4.7 LCD Menampilkan Karakter Saat Kartu Terdaftar

```
else
{
    lcd.clear();
    lcd.setCursor(4, 2);
    lcd.print("Akses Ditolak");
    digitalWrite(Relay, HIGH);
    digitalWrite(LED_R, HIGH);
    delay(500);
    lcd.clear();
    lcd.setCursor(4, 0);
    lcd.print("Laboratorium");
    lcd.setCursor(0, 1);
    lcd.print("Elektronika Terapan");
    lcd.setCursor(1, 2);
    lcd.print("Silahkan Scan Kartu");
    lcd.setCursor(1, 3);
    lcd.print("Anda :)");
    goto cont;//go directly to line 71
}
```

Gambar 4.8 Program Untuk Menampilkan Karakter Pada LCD Saat Kartu Tidak Terdaftar

Hasil program pada Gambar 4.8 untuk menampilkan karakter nama E-KTP pengakses pada LCD dapat dilihat pada Gambar 4.9.



Gambar 4. 9 LCD Menampilkan Karakter Saat Kartu Tidak Terdaftar

4.1.4.2 Pengujian Pengaturan *Delay* pada *Relay*

Pada Gambar 4.10 merupakan pengaturan kontrol dapat ditambahkan dengan pengaturan waktu agar peralatan listrik yang digunakan lebih efektif, misalnya ketika membuka pintu loker yang dimasukan dari E-KTP adalah benar maka mikro akan memberikan *input high* pada *relay* untuk mengaktifkan solenoid, dan waktu untuk memberikan *input high* tersebut dapat diatur sesuai dengan waktu yang diinginkan. Pengaturan waktu pada mikrokontroler Arduino dapat dilakukan dengan pengaturan pada programannya yaitu pengaturan *delay*. Pemasukan $delay\ 1000 = 1$ detik sehingga *relay* akan aktif 1 detik setelah kode E-KTP yang dimasukan benar, dan berarti jika ingin diperoleh relay aktif 5 detik setelah kode E-KTP yang dimasukan benar, maka dapat dimasukan angka 3.000 yang setara 3 detik pada pemrograman Arduino Mega. Untuk lebih jelasnya dapat dilihat pada contoh program dibawah ini:

```
digitalWrite(Relay, HIGH);  
delay(3000);
```

Gambar 4.10 Program *Delay* Pada *Relay*

4.1.4.3 Pengujian RFID Reader

Pada pengujian ini RFID *Reader* membaca ID E-KTP menggunakan program seperti pada Gambar 4.11 dan tampilan pada laptop atau serial monitor *software* Arduino IDE dapat dilihat pada Gambar 4.11.

```
Tugas_Akhir$
// Look for new cards
if ( ! mfrc522.PICC_IsNewCardPresent() )
{
    return;
}
// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial() )
{
    return;
}
Serial.println("**Card Detected:**");

//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();
```

Gambar 4.11 Program Pengujian Pembacaan E-KTP

4.1.4.4 Pengujian (Solenoid) Sistem Untuk Membuka Dan Mengunci

Pengujian ini dilakukan untuk menguji *output* Arduino dalam memberi *output* pada *Relay* untuk membuka dan mengunci solenoid. Kabel positif dari suplai adaptor 12V dihubungkan ke port COM pada *relay*, dan yang satunya lagi dihubungkan ke *port* NO pada *relay*. Keadaan ini membuat *supply* dari adaptor tidak dapat mengalir karena posisi dalam keadaan Normaly Open. Ketika *relay* aktif maka saklar dari *relay* yang pada awalnya terhubung *port* COM dan *port* NC, akan menjadi terhubung *port* COM dan *port* NO. Keadaan ini membuat

supply dari adaptor mengalir ke solenoid dan membuat solenoid aktif untuk membuka solenoid.

```
digitalWrite(RELAY, LOW);
```

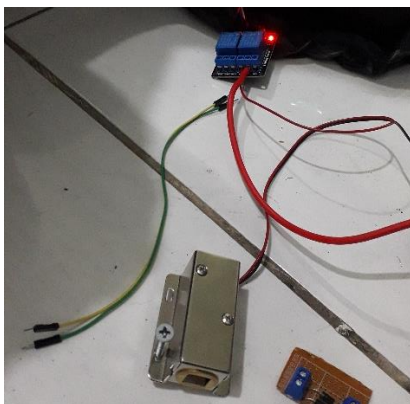
Gambar 4.12 Program Solenoid Keadaan Mengunci

Hasil dari program pada gambar 4.11 yaitu keadaan solenoid dalam keadaan mengunci atau dalam keadaan seperti awal karena tidak mendapatkan aliran listrik karena relay dalam keadaan mati.

```
digitalWrite(RELAY, HIGH);
```

Gambar 4.13 Program Solenoid Keadaan Membuka

Hasil dari program pada gambar 4.13 yaitu keadaan solenoid dalam keadaan terbuka karena mendapatkan aliran listrik karena relay dalam keadaan aktif.



Gambar 4.14 Solenoid Pada Keadaan Terbuka

4.1.4.5 Pengujian Pengiriman Data dari Arduino ke Database

Pengujian ini dilakukan untuk mengetahui bentuk data yang dikirim dari Arduino ke *database*. Data yang sudah dikirim pada

database nantinya dapat dilihat secara online dengan cara membuka hosting yang digunakan dan nantinya data dapat disimpan. Berikut ini gambar 4.15 menunjukkan data yang ditampilkan pada sebuah *database*.

	no	date	name	number	timein	timeout
<input type="checkbox"/>	61	2019/12/17-Tuesday	Tama	1011	0:22:23	kosong
<input type="checkbox"/>	62	2019/12/17-Tuesday	ALFAN	1012	0:26:15	kosong
<input type="checkbox"/>	63	2019/12/17-Tuesday	ALFAN	1012	kosong	0:38:31
<input type="checkbox"/>	64	2019/12/17-Tuesday	Tama	1011	kosong	0:46:6
<input type="checkbox"/>	65	2019/12/17-Tuesday	ALFAN	1012	kosong	0:46:13
<input type="checkbox"/>	66	2019/12/17-Tuesday	Tama	1011	0:23:9	kosong

Gambar 4.15 Pengujian 2 ID pada *Database*

Pada gambar diatas digunakan dua ID *card* berupa E-KTP dengan nama pengguna yang terdaftar Tama dan Alfian. Tabel yang ditampilkan yaitu no, date, name, number, timein, dan timeout. Untuk data E-KTP pertama kali *tap* data waktu yang diisikan pada tabel timein dan tabel timeout menunjukkan kosong. Ketika sudah *tap* E-KTP kedua kali dan seterusnya sampai jam 12 malam data waktu yang diisikan tabel timeout. Data tanggal dan hari diperoleh dari komponen RTC (*Real Time Clock*). Berikut ini data yang didapatkan dari dua ID berupa E-KTP. Pada Gambar 4.15 tabel “*date*” menunjukkan tanggal dan hari akses ruangan. Kemudian tabel “*name*” merupakan nama identitas dari pengakses ruangan. Tabel “*number*” menunjukkan nomor identitas dari pengakses ruangan. Selanjutnya tabel “*timein*” dan “*timeout*” menunjukkan waktu masuk dan waktu keluar dari pengakses ruangan. untuk waktu keluar pengakses ruangan.

Tabel 4.7 Pengujian *Database* dengan E-KTP Tama

No	Tanggal	Nama	Identitas	Waktu Masuk	Waktu Keluar
1	2019/12/17-Tuesday	Tama	1011	0:22:23	Kosong
2	2019/12/17-Tuesday	Tama	1011	kosong	0:46:6

3	2019/12/17- Tuesday	Tama	1011	kosong	0:46:13
---	------------------------	------	------	--------	---------

Data name dan number pada tabel 4.7 sudah sesuai dengan ID E-KTP yang telah didaftar atau diprogram pada Arduino IDE. Dimana pemrograman dapat dilihat pada gambar 4.16 berikut ini.

```

if (content.substring(1) == Nama1)
{
  Name = "Tama"; //user name
  Number = 1011; //user number
  j = 0;
  s = 0;
  digitalWrite(Relay, LOW);
  digitalWrite(LED_G, HIGH);
  delay(3000);
  digitalWrite(Relay, HIGH);
}

```

Gambar 4.16 Program Data E-KTP Tama

Pada program seperti pada gambar 4.16 data ID pertama dituliskan Name = “Tama” dan Number = 1011, program telah sesuai dengan data yang dikirimkan pada *database*. Berikutnya pengujian data ID kedua dapat dilihat dibawah ini.

Tabel 4.8 Pengujian *Database* dengan E-KTP ALFAN

No	Tanggal	Name	Identitas	Waktu Masuk	Waktu Keluar
1	2019/12/17- Tuesday	ALFAN	1012	0:26:15	Kosong
2	2019/12/17- Tuesday	ALFAN	1012	Kosong	0:38:31
3	2019/12/17- Tuesday	ALFAN	1012	Kosong	0:46:13

Data name dan number pada tabel 4.8 sudah sesuai dengan ID E-KTP yang telah didaftar atau diprogram pada Arduino IDE. Dimana pemrograman dapat dilihat pada gambar 4.17 berikut ini.

```

else if (content.substring(1) == Name2)
{
    Name = "ALFAN"; //user name
    Number = 1012; //user number
    j = 1;
    s = 1;
    digitalWrite(Relay, LOW);
    digitalWrite(LED_G, HIGH);
    delay(3000);
    digitalWrite(Relay, HIGH);
}

```

Gambar 4.17 Program Data E-KTP ALFAN

Pada program seperti pada gambar 4.17 data ID pertama dituliskan Name = “ALFAN” dan Number = 1012, program telah sesuai dengan data yang dikirimkan pada *database*. Selanjutnya dilakukan pengujian terhadap dua puluh lima ID *card*, dimana setiap ID *card* dilakukan pengujian sebanyak 3 kali.

	no	date	name	number	timein	timeout
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	165	2019/12/25-Wednesday	ARGON LUTHFAN	1017	18:18:5	kosong
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	166	2019/12/25-Wednesday	ARGON LUTHFAN	1017	kosong	18:18:14
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	167	2019/12/25-Wednesday	ARGON LUTHFAN	1017	kosong	18:18:20
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	168	2019/12/25-Wednesday	PANDU DINATA	1016	18:19:48	kosong
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	169	2019/12/25-Wednesday	PANDU DINATA	1016	kosong	18:19:55
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	170	2019/12/25-Wednesday	PANDU DINATA	1016	kosong	18:20:1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	171	2019/12/25-Wednesday	MUCH ARIF RAHMAN	1020	18:20:10	kosong
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	172	2019/12/25-Wednesday	MUCH ARIF RAHMAN	1020	kosong	18:20:16
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	173	2019/12/25-Wednesday	MUCH ARIF RAHMAN	1020	kosong	18:20:22
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	174	2019/12/25-Wednesday	DZULFIKAR	1022	18:25:3	kosong
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	175	2019/12/25-Wednesday	DZULFIKAR	1022	kosong	18:25:9
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	176	2019/12/25-Wednesday	DZULFIKAR	1022	kosong	18:25:15
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	177	2019/12/25-Wednesday	FERIAN BAGUS	1014	18:18:13	kosong
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	178	2019/12/25-Wednesday	FERIAN BAGUS	1014	kosong	18:18:19
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	179	2019/12/25-Wednesday	FERIAN BAGUS	1014	kosong	18:18:24
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	180	2020/1/9-Thursday	DWIKI	1028	6:22:42	kosong
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	181	2020/1/9-Thursday	DWIKI	1028	kosong	6:22:47
	no	date	name	number	timein	timeout
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	182	2020/1/9-Thursday	DWIKI	1028	kosong	6:22:54
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	183	2020/1/9-Thursday	RAMADHAN	1028	6:23:0	kosong

Gambar 4.18 Pengujian Kedua Pada *Database* Dengan 6 ID *Card*

Pada pengujian seperti pada Gambar 4.18 telah dilakukan pengujian pengiriman data pada *database* dengan 6 ID *card*, dimana

setiap ID card dilakukan pengujian sebanyak 3 kali. Berikut tabel dibawah ini merupakan pengujian dari 6 ID card.

Tabel 4.9 Pengujian *Database* Kedua dengan 6 ID Card

No	Tanggal	Nama	Identitas	Waktu Masuk	Waktu Keluar
1	2019/12/25-Wednesday	ARGON LUTHFAN	1017	18:18:50	Kosong
2	2019/12/25-Wednesday	ARGON LUTHFAN	1017	Kosong	18:18:14
3	2019/12/25-Wednesday	ARGON LUTHFAN	1017	Kosong	18:18:20
4	2019/12/25-Wednesday	PANDU DINATA	1016	18:19:48	Kosong
5	2019/12/25-Wednesday	PANDU DINATA	1016	Kosong	18:19:55
6	2019/12/25-Wednesday	PANDU DINATA	1016	Kosong	18:20:1
7	2019/12/25-Wednesday	MUCH ARIF RAHMAN	1020	18:20:10	Kosong
8	2019/12/25-Wednesday	MUCH ARIF RAHMAN	1020	Kosong	18:20:16
9	2019/12/25-Wednesday	MUCH ARIF RAHMAN	1020	Kosong	18:20:22
10	2019/12/25-Wednesday	DZULFIKAR	1022	18:25:3	Kosong
11	2019/12/25-Wednesday	DZULFIKAR	1022	Kosong	18:25:9
12	2019/12/25-Wednesday	DZULFIKAR	1022	Kosong	18:25:15
13	2019/12/25-Wednesday	FERIAN BAGUS	1014	18:18:13	Kosong
14	2019/12/25-Wednesday	FERIAN BAGUS	1014	Kosong	18:18:19
15	2019/12/25-Wednesday	FERIAN BAGUS	1014	Kosong	18:18:24
16	2020/1/9-Thursday	DWIKI	1028	6:22:42	Kosong

17	2020/1/9-Thursday	DWIKI	1028	Kosong	6:22:47
18	2020/1/9-Thursday	DWIKI	1028	Kosong	6:22:54

Setelah itu dilakukan pengujian ketiga dengan ID *card* sebanyak 5 dan setiap ID *card* dilakukan pengujian sebanyak 3 kali. Berikut dibawah pengujian dapat dilihat pada Gambar 4.19

	no	date	name	number	timein	timeout
<input type="checkbox"/>	183	2020/1/9-Thursday	RAMADHAN	1028	6:23:0	kosong
<input type="checkbox"/>	184	2020/1/9-Thursday	RAMADHAN	1028	kosong	6:23:6
<input type="checkbox"/>	185	2020/1/9-Thursday	RAMADHAN	1028	kosong	6:23:11
<input type="checkbox"/>	186	2020/1/9-Thursday	DWIKI	1028	kosong	6:23:55
<input type="checkbox"/>	187	2020/1/9-Thursday	DWIKI	1028	kosong	6:24:1
<input type="checkbox"/>	188	2020/1/9-Thursday	DWIKI	1028	kosong	6:24:6
<input type="checkbox"/>	189	2020/1/9-Thursday	ILHAM	1028	kosong	6:24:12
<input type="checkbox"/>	190	2020/1/9-Thursday	ILHAM	1028	kosong	6:24:17
<input type="checkbox"/>	191	2020/1/9-Thursday	ILHAM	1028	kosong	6:24:23
<input type="checkbox"/>	192	2020/1/9-Thursday	DWI	1028	6:24:29	kosong
<input type="checkbox"/>	193	2020/1/9-Thursday	DWI	1028	kosong	6:24:35
<input type="checkbox"/>	194	2020/1/9-Thursday	DWI	1028	kosong	6:24:41
<input type="checkbox"/>	195	2020/1/9-Thursday	ADIT	1028	6:24:47	kosong
<input type="checkbox"/>	196	2020/1/9-Thursday	ADIT	1028	kosong	6:24:53
<input type="checkbox"/>	197	2020/1/9-Thursday	ADIT	1028	kosong	6:24:58
<input type="checkbox"/>	198	2020/1/9-Thursday	ALWAN	1028	6:25:4	kosong
<input type="checkbox"/>	199	2020/1/9-Thursday	ALWAN	1028	kosong	6:25:11
<input type="checkbox"/>	200	2020/1/9-Thursday	ALWAN	1028	kosong	6:25:17
<input type="checkbox"/>	201	2020/1/9-Thursday	ALWAN	1028	kosong	6:25:23

Gambar 4.19 Pengujian Ketiga Pada *Database* Dengan 5 ID Card

Pada pengujian seperti pada Gambar 4.19 telah dilakukan pengujian pengiriman data pada *database* dengan 5 ID *card*, dimana setiap ID *card* dilakukan pengujian sebanyak 3 kali. Berikut tabel dibawah ini merupakan pengujian dari 5 ID *card*.

Tabel 4.10 Pengujian *Database* Ketiga dengan 5 ID Card

No	Tanggal	Nama	Identitas	Waktu Masuk	Waktu Keluar
1	2020/1/9-Thursday	RAMADHAN	1028	6:23:0	Kosong

2	2020/1/9-Thursday	RAMADHAN	1028	Kosong	6:23:6
3	2020/1/9-Thursday	RAMADHAN	1028	Kosong	6:23:11
4	2020/1/9-Thursday	ILHAM	1028	6:24:12	Kosong
5	2020/1/9-Thursday	ILHAM	1028	Kosong	6:24:17
6	2020/1/9-Thursday	ILHAM	1028	Kosong	6:24:23
7	2020/1/9-Thursday	DWI	1028	6:24:29	Kosong
8	2020/1/9-Thursday	DWI	1028	Kosong	6:24:35
9	2020/1/9-Thursday	DWI	1028	Kosong	6:24:58
10	2020/1/9-Thursday	ADIT	1028	6:24:27	Kosong
11	2020/1/9-Thursday	ADIT	1028	Kosong	6:24:53
12	2020/1/9-Thursday	ADIT	1028	Kosong	6:24:58
13	2020/1/9-Thursday	ALWAN	1028	6:25:4	Kosong
14	2020/1/9-Thursday	ALWAN	1028	Kosong	6:25:11
15	2020/1/9-Thursday	ALWAN	1028	Kosong	6:25:17

Setelah itu dilakukan pengujian keempat dengan ID *card* sebanyak 7 dan setiap ID *card* dilakukan pengujian sebanyak 3 kali. Berikut dibawah pengujian dapat dilihat pada Gambar 4.20.

← T →	no	date	name	number	timein	timeout	
<input type="checkbox"/> Edit Copy Delete	202	2020/1/9-Thursday	DANU	1028	6:25:29	kosong	
<input type="checkbox"/> Edit Copy Delete	203	2020/1/9-Thursday	DANU	1028	kosong	6:25:35	
<input type="checkbox"/> Edit Copy Delete	204	2020/1/9-Thursday	DANU	1028	kosong	6:25:42	
<input type="checkbox"/> Edit Copy Delete	210	2020/1/10-Friday	Aldo	1028	2:11:55	kosong	
<input type="checkbox"/> Edit Copy Delete	211	2020/1/10-Friday	Aldo	1028	kosong	2:12:2	
<input type="checkbox"/> Edit Copy Delete	212	2020/1/10-Friday	Aldo	1028	kosong	2:12:8	
<input type="checkbox"/> Edit Copy Delete	213	2020/1/10-Friday	ARIF RAHMAN	1039	2:12:33	kosong	
<input type="checkbox"/> Edit Copy Delete	214	2020/1/10-Friday	ARIF RAHMAN	1039	kosong	2:12:40	
<input type="checkbox"/> Edit Copy Delete	215	2020/1/10-Friday	ARIF RAHMAN	1039	kosong	2:12:47	
<input type="checkbox"/> Edit Copy Delete	216	2020/1/10-Friday	WIRAYUDHA	1043	2:13:7	kosong	
<input type="checkbox"/> Edit Copy Delete	217	2020/1/10-Friday	WIRAYUDHA	1043	kosong	2:13:19	
<input type="checkbox"/> Edit Copy Delete	218	2020/1/10-Friday	WIRAYUDHA	1043	kosong	2:13:26	
<input type="checkbox"/> Edit Copy Delete	219	2020/1/10-Friday	ACHMAD	1042	2:13:48	kosong	
<input type="checkbox"/> Edit Copy Delete	220	2020/1/10-Friday	ACHMAD	1042	kosong	2:13:54	
<input type="checkbox"/> Edit Copy Delete	221	2020/1/10-Friday	ACHMAD	1042	kosong	2:14:1	
<input type="checkbox"/> Edit Copy Delete	222	2020/1/10-Friday	ANDY RAHMA	1038	2:14:12	kosong	
<input type="checkbox"/> Edit Copy Delete	223	2020/1/10-Friday	ANDY RAHMA	1038	kosong	2:14:18	
<input type="checkbox"/> Edit Copy Delete	224	2020/1/10-Friday	ANDY RAHMA	1038	kosong	2:14:26	
<input type="checkbox"/> Edit Copy Delete	225	2020/1/10-Friday	KHABIB NUR	1037	2:14:34	kosong	
<input type="checkbox"/> Edit Copy Delete	226	2020/1/10-Friday	KHABIB NUR	1037	kosong	2:14:40	
6751508251/3rdparty/phpMyAdmin/tbl_import.php?db=k2015996_data_kehadiran&table=KehadiranData 7						kosong	2:14:46

Gambar 4.20 Pengujian Keempat Pada *Database* Dengan 7 ID Card

Pada pengujian seperti pada Gambar 4.20 telah dilakukan pengujian pengiriman data pada *database* dengan 7 ID card, dimana setiap ID card dilakukan pengujian sebanyak 3 kali. Berikut tabel dibawah ini merupakan pengujian dari 7 ID card.

Tabel 4.11 Pengujian *Database* Keempat dengan 5 ID Card

No	Tanggal	Nama	Identitas	Waktu Masuk	Waktu Keluar
1	2020/1/9-Thursday	DANU	1028	6:25:29	Kosong
2	2020/1/9-Thursday	DANU	1028	Kosong	6:25:35
3	2020/1/9-Thursday	DANU	1028	Kosong	6:25:42
4	2020/1/10-Friday	Aldo	1028	2:11:55	Kosong
5	2020/1/10-Friday	Aldo	1028	Kosong	2:12:2

6	2020/1/10-Friday	Aldo	1028	Kosong	2:12:8
7	2020/1/10-Friday	ARIF RAHMAN	1039	2:12:33	Kosong
8	2020/1/10-Friday	ARIF RAHMAN	1039	Kosong	2:12:40
9	2020/1/10-Friday	ARIF RAHMAN	1039	Kosong	2:12:47
10	2020/1/10-Friday	WIRAYUDHA	1043	2:13:7	Kosong
11	2020/1/10-Friday	WIRAYUDHA	1043	Kosong	2:13:9
12	2020/1/10-Friday	WIRAYUDHA	1043	Kosong	2:13:16
13	2020/1/10-Friday	ACHMAD	1042	2:13:48	Kosong
14	2020/1/10-Friday	ACHMAD	1042	Kosong	2:13:54
15	2020/1/10-Friday	ACHMAD	1042	Kosong	2:14:1
16	2020/1/10-Friday	ANDY RAHMA	1038	2:14:12	Kosong
17	2020/1/10-Friday	ANDY RAHMA	1038	Kosong	2:14:18
18	2020/1/10-Friday	ANDY RAHMA	1038	Kosong	2:14:26
19	2020/1/10-Friday	KHABIB NUR	1037	2:14:34	Kosong
20	2020/1/10-Friday	KHABIB NUR	1037	Kosong	2:14:40
21	2020/1/10-Friday	KHABIB NUR	1037	Kosong	2:14:46

Setelah itu dilakukan pengujian kelima dengan ID *card* sebanyak 7 dan setiap ID *card* dilakukan pengujian sebanyak 3 kali. Berikut dibawah pengujian dapat dilihat pada Gambar 4.21.

	no	date	name	number	timein	timeout
<input type="checkbox"/> Edit Copy Delete	228	2020/1/10-Friday	TANIARA CHEVIN	1040	2:15:9	kosong
<input type="checkbox"/> Edit Copy Delete	229	2020/1/10-Friday	TANARA CHEVIN	1040	kosong	2:15:15
<input type="checkbox"/> Edit Copy Delete	230	2020/1/10-Friday	TANARA CHEVIN	1040	kosong	2:15:21
<input type="checkbox"/> Edit Copy Delete	231	2020/1/10-Friday	PRAYOGA	1049	2:15:27	kosong
<input type="checkbox"/> Edit Copy Delete	232	2020/1/10-Friday	PRAYOGA	1049	kosong	2:15:33
<input type="checkbox"/> Edit Copy Delete	233	2020/1/10-Friday	PRAYOGA	1049	kosong	2:15:38
<input type="checkbox"/> Edit Copy Delete	234	2020/1/10-Friday	MARKO	1048	2:15:44	kosong
<input type="checkbox"/> Edit Copy Delete	235	2020/1/10-Friday	MARKO	1048	kosong	2:15:50
<input type="checkbox"/> Edit Copy Delete	236	2020/1/10-Friday	MARKO	1048	kosong	2:15:56
<input type="checkbox"/> Edit Copy Delete	237	2020/1/10-Friday	CHEMAL	1047	2:16:1	kosong
<input type="checkbox"/> Edit Copy Delete	238	2020/1/10-Friday	CHEMAL	1047	kosong	2:16:7
<input type="checkbox"/> Edit Copy Delete	239	2020/1/10-Friday	CHEMAL	1047	kosong	2:16:13
<input type="checkbox"/> Edit Copy Delete	240	2020/1/10-Friday	PRATAMA	1046	2:16:18	kosong
<input type="checkbox"/> Edit Copy Delete	241	2020/1/10-Friday	PRATAMA	1046	kosong	2:16:24
<input type="checkbox"/> Edit Copy Delete	242	2020/1/10-Friday	PRATAMA	1046	kosong	2:16:31
<input type="checkbox"/> Edit Copy Delete	243	2020/1/10-Friday	ALLICA	1045	2:16:37	kosong
<input type="checkbox"/> Edit Copy Delete	244	2020/1/10-Friday	ALLICA	1045	kosong	2:16:43
<input type="checkbox"/> Edit Copy Delete	245	2020/1/10-Friday	ALLICA	1045	kosong	2:16:49
<input type="checkbox"/> Edit Copy Delete	246	2020/1/10-Friday	RAHMAD	1044	2:16:55	kosong
<input type="checkbox"/> Edit Copy Delete	247	2020/1/10-Friday	RAHMAD	1044	kosong	2:17:1
					kosong	2:17:6

Gambar 4.21 Pengujian Kelima Pada *Database* Dengan 7 ID Card

Pada pengujian seperti pada Gambar 4.21 telah dilakukan pengujian pengiriman data pada *database* dengan 7 ID card, dimana setiap ID card dilakukan pengujian sebanyak 3 kali. Berikut tabel dibawah ini merupakan pengujian dari 7 ID card.

Tabel 4.12 Pengujian *Database* Kelima dengan 5 ID Card

No	Tanggal	Nama	Identitas	Waktu Masuk	Waktu Keluar
1	2020/1/9-Thursday	TANARA CHEVIN	1040	2:15:9	Kosong
2	2020/1/9-Thursday	TANARA CHEVIN	1040	Kosong	2:15:15
3	2020/1/9-Thursday	TANARA CHEVIN	1040	Kosong	2:15:21
4	2020/1/10-Friday	PRAYOGA	1049	2:15:17	Kosong

5	2020/1/10-Friday	PRAYOGA	1049	Kosong	2:15:22
6	2020/1/10-Friday	PRAYOGA	1049	Kosong	2:15:38
7	2020/1/10-Friday	MARKO	1039	2:15:44	Kosong
8	2020/1/10-Friday	MARKO	1039	Kosong	2:15:50
9	2020/1/10-Friday	MARKO	1039	Kosong	2:15:56
10	2020/1/10-Friday	CHEMAL	1043	2:13:7	Kosong
11	2020/1/10-Friday	CHEMAL	1043	Kosong	2:13:9
12	2020/1/10-Friday	CHEMAL	1043	Kosong	2:13:16
13	2020/1/10-Friday	PRATAMA	1042	2:13:48	Kosong
14	2020/1/10-Friday	PRATAMA	1042	Kosong	2:13:54
15	2020/1/10-Friday	PRATAMA	1042	Kosong	2:14:1
16	2020/1/10-Friday	ALLICA	1038	2:14:12	Kosong
17	2020/1/10-Friday	ALLICA	1038	Kosong	2:14:18
18	2020/1/10-Friday	ALLICA	1038	Kosong	2:14:26
19	2020/1/10-Friday	RAHMAD	1037	2:14:34	Kosong
20	2020/1/10-Friday	RAHMAD	1037	Kosong	2:14:40
21	2020/1/10-Friday	RAHMAD	1037	Kosong	2:14:46

Seluruh pengujian *database* dilakukan dengan menggunakan ID *card* sebanyak 27 ID *card* dan untuk setiap ID *card* dilakukan tiga kali pengujian untuk mendapatkan data waktu *time in* dan *time out*.

4.2 Pengujian Sistem Keseluruhan

Dalam pengujian keseluruhan ini terdiri dari awal alat saat dihidupkan untuk dibuka dengan E-KTP yang terdaftar dan akan mengirim data pengakses terdaftar ke *database* serta akan menolak akses membuka pintu jika E-KTP tidak terdaftar.

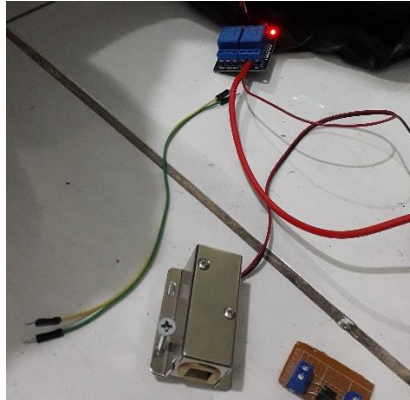


Gambar 4.22 Keadaan Layar Awal Saat Pintu Dihidupkan

Layar pada Gambar 4.22 menunjukkan jika alat telah siap digunakan dan dalam keadaan hidup dengan itu loker siap digunakan dengan menempelkan E-KTP yang telah didaftarkan agar pintu loker dapat dibuka.



Gambar 4.23 Saat E-KTP Terdaftar Untuk Membuka Pintu



Gambar 4.24 Keadaan Solenoid Ketika Membaca E-KTP Terdaftar

Pada Gambar 4.24 menunjukkan keadaan solenoid akan terbuka ketika membaca E-KTP yang sudah terdaftar. Solenoid dapat terbuka karena adanya aliran listrik dari keadaan relay yang aktif dimana common relay disambungkan pada *power supply* 12v dan NO relay disambungkan pada kabel positif solenoid. Pada Gambar 4.23 layar menunjukkan nama E-KTP yang telah terdaftar agar dapat membuka pintu dengan delay 3 detik dan setelah itu alat akan mengirim data E-KTP terdaftar ke *database*. Data yang dikirim seperti pada Gambar 4.25.

```
Date,Name,Number,Time IN,Time OUT
2019/12/17-Tuesday,Tama,1011,0:22:22,kosong
2019/12/17-Tuesday,Tama,1011,kosong,0:22:28
2019/12/17-Tuesday,Tama,1011,kosong,0:22:35
```

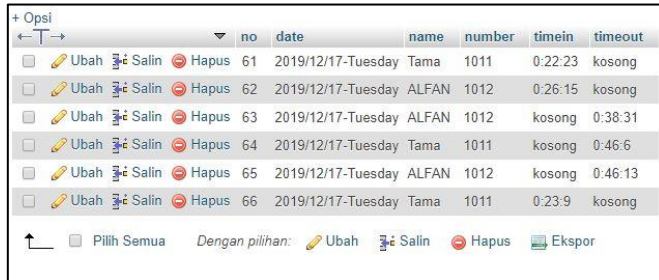
Gambar 4.25 Data yang Dikirim dari Arduino ke ESP32

Pada gambar diatas percobaan dilakukan tiga kali. Kemudian pada Gambar 4.26 Data dari Arduino akan di parsing pada ESP32 yang selanjutnya akan dikirim pada *database*. Berikut ini data yang di parsing oleh ESP32.

```
Connected to WiFi network with IP Address: 192.168.43.10
2019/12/17-Tuesday
Tama
1011
kosong
0:34:24
```

Gambar 4.26 Data Parsing dari Arduino ke ESP32

Selanjutnya data dari ESP32 yang sudah diparsing akan dikirim pada *database*. Berikut ini gambar 4.27 pengujian pengiriman ESP32 ke *database*.



	no	date	name	number	timein	timeout
<input type="checkbox"/>	61	2019/12/17-Tuesday	Tama	1011	0:22:23	kosong
<input type="checkbox"/>	62	2019/12/17-Tuesday	ALFAN	1012	0:26:15	kosong
<input type="checkbox"/>	63	2019/12/17-Tuesday	ALFAN	1012	kosong	0:38:31
<input type="checkbox"/>	64	2019/12/17-Tuesday	Tama	1011	kosong	0:46:6
<input type="checkbox"/>	65	2019/12/17-Tuesday	ALFAN	1012	kosong	0:46:13
<input type="checkbox"/>	66	2019/12/17-Tuesday	Tama	1011	0:23:9	kosong

Gambar 4.27 Pengiriman Data Arduino ke *Database*

Data yang sudah tersimpan pada *database* nantinya akan dapat disimpan dan kemudian setelah data menumpuk banyak, data juga dapat dihapus melalui MySQL.



Gambar 4.28 Saat E-KTP Tidak Terdaftar Untuk Membuka Pintu

Pada Gambar 4.28 layar menunjukkan akses ditolak karena E-KTP yang ditempelkan merupakan E-KTP yang tidak terdaftar dalam program. Layar menunjukkan akses ditolak karena E-KTP yang ditempelkan merupakan E-KTP yang tidak terdaftar dalam program selanjutnya led merah sebagai indikator akan menyala dan alat tidak akan mengirim data apapun ke *database*.

BAB V

PENUTUP

Dari hasil yang telah didapatkan selama proses dan pembuatan serta proses analisis data untuk Proyek Akhir ini, maka dapat diambil beberapa kesimpulan dan saran yang berguna untuk perbaikan dan pengembangan agar nantinya bisa bermanfaat.

5.1 Kesimpulan

Dari hasil penelitian dan pembahasan dapat disimpulkan bahwa:

1. Alat pengaman pintu otomatis menggunakan E-KTP ini mampu membaca ID E-KTP terdaftar dan tidak terdaftar dengan pengujian selama 10 kali berturut turut.
2. Alat pengaman pintu juga dapat digunakan sebagai presensi kehadiran, dimana nantinya data presensi dikirim pada *database* dengan melalui jaringan WiFi Hotspot.
3. Selama pengujian menggunakan 27 ID *card*, alat mampu mengirimkan *database* setiap ID tiga kali pengujian atau pengiriman pada *database*.
4. Kendala pengiriman data pada *database* terjadi ketika jaringan WiFi Hotspot lemot. Biasanya data yang dikirim tidak diisikan pada tabel yang tersedia.
5. Kendala pada alat yaitu ketika *push button* ditekan secara terus menerus, Arduino akan berhenti menjalankan perintah
6. Pada pengujian, batas jarak sensor RFID adalah 1,5 cm. Ketika diuji pada jarak 2 cm RFID sudah tidak bisa membaca E-KTP

5.2 Saran

Dari hasil penelitian dan pembahasan dapat diketahui tambahan saran yaitu alat dapat ditambahkan registrasi secara otomatis agar tidak memprogram ulang Arduino ketika akan menambah *member* baru untuk mengakses ruangan.

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

DAFTAR PUSTAKA

- [1] Lim Charles, *Design and Implementation of E-KTP (Indonesian Electronic Identity Card) Key Management System*. Swiss: Swiss German University, 2011.
- [2] Dang, H, T, "*Investigate And Design A 13.56MHz RFID Reader*", *Tesis, School Of Electrical Engineering Ho Chi Minh City International University*, Vietnam: Vietnam National University 2013.
- [3] Kadir, Abdul, *Panduan Praktis Mempelajari Aplikasi Mikrokontroler Dan Pemrogramannya Menggunakan Arduino*, Andi Yogyakarta, 2013.
- [4] Nwaji, O. G., N. C. Onyebuchi, dan Dr. O. F. Kelechi, "*Automatic Door Unit Radio Frequency Identification (RFID) Based Attendance System*", *International Journal Science and Emerging Technologies*, Vol 5 No. 6, Halaman 200-211, 2013.
- [5] Margunadi, A, R, *Pengantar Umum Elektro Teknik Edisi Pertama*, PT Dian Rakyat. Jakarta, 1983.
- [6] Istiyanto, J, E, *Pengantar Elektronika & Instrumentasi (Pendekatan Project Arduino dan Android) Edisi Pertama*, Andi, Yogyakarta, 2014.
- [7] Pratama, H, S, "*RFID Sebagai Pengaman Pintu Laboratorium Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang*", *Skripsi*, Program Studi Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang, Semarang, 2014.
- [8] Yunas, L, "*Sistem Peringatan dan Pelacakan Kendaraan Bermotor Hilang*", *Tugas Akhir*, Program Studi Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya, 2013.

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

LAMPIRAN A

Datasheet MFRC522

NXP Semiconductors

MFRC522

Contactless Reader IC

8. Functional description

MFRC522 transmission module supports the Reader/Writer mode for ISO/IEC 14443A/MIFARE® with different transfer speeds and modulation schemes.

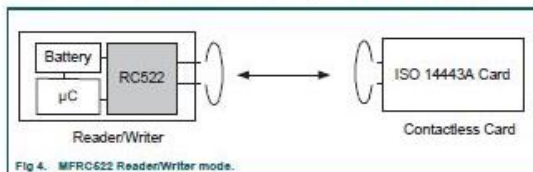


Fig 4. MFRC522 Reader/Writer mode.

The following diagram [Figure 5 "ISO/IEC 14443A/MIFARE® Reader/Writer mode communication diagram."](#) describes the communication on a physical level.

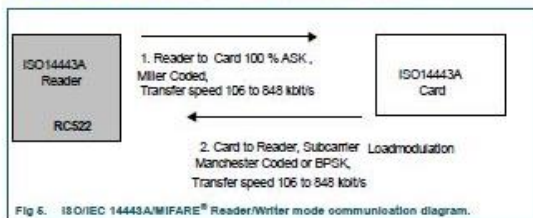


Fig 5. ISO/IEC 14443A/MIFARE® Reader/Writer mode communication diagram.

The communication overview in [Table 4 "Communication overview for ISO/IEC 14443A/MIFARE® reader/writer"](#) describes the physical parameters.

Table 4: Communication overview for ISO/IEC 14443A/MIFARE® reader/writer

Communication direction	ISO/IEC 14443A/ MIFARE®				
	transfer speed	106 kbit/s	212 kbit/s	424 kbit/s	848 kbit/s
Reader → Card (send data from the MFRC522 to a card)	Modulation on reader side	100% ASK	100% ASK	100% ASK	100% ASK
	bit coding	Modified Miller coding	Modified Miller coding	Modified Miller coding	Modified Miller coding
	Bitlength	(128/13.56) µs	(64/13.56) µs	(32/13.56) µs	(16/13.56) µs

Table 3: Pin description ...continued

Symbol	Pin	Type	Description
TX1	11	O	Transmitter 1: delivers the modulated 13.56 MHz energy carrier
TVDD	12	PWR	Transmitter Power Supply: supplies the output stage of TX1 and TX2
TX2	13	O	Transmitter 2: delivers the modulated 13.56 MHz energy carrier
TVSS	10, 14	PWR	Transmitter Ground: supplies the output stage of TX1 and TX2
AVDD	15	PWR	Analog Power Supply
VMID	16	PWR	Internal Reference Voltage: This pin delivers the internal reference voltage.
RX	17	I	Receiver Input: Pin for the received RF signal.
AVSS	18	PWR	Analog Ground
AUX1	19	O	Auxiliary Outputs: These pins are used for testing.
AUX2	20	O	
OSCIN	21	I	Crystal Oscillator Input: input to the inverting amplifier of the oscillator. This pin is also the input for an externally generated clock ($f_{osc} = 27.12$ MHz).
OSCCOUT	22	O	Crystal Oscillator Output: Output of the inverting amplifier of the oscillator.
IRQ	23	O	Interrupt Request: output to signal an interrupt event
SDA	24	I	Serial Data Line ^[1]
D1	25	I/O	Data Pins for different interfaces (test port, I ² C, SPI, UART) ^[2]
D2	26	I/O	
D3	27	I/O	
D4	28	I/O	
D5	29	I/O	
D6	30	I/O	
D7	31	I/O	
EA	32	I	External Address: This Pin is used for coding I2C Address ^[1]

[1] Connection of heat sink pad on package bottom side is not necessary. Optional connection to DVSS is possible.

[2] The pin functionality for the interfaces is explained in [Section 10 "DIGITAL Interfaces"](#).

7. Pinning information

7.1 Pinning

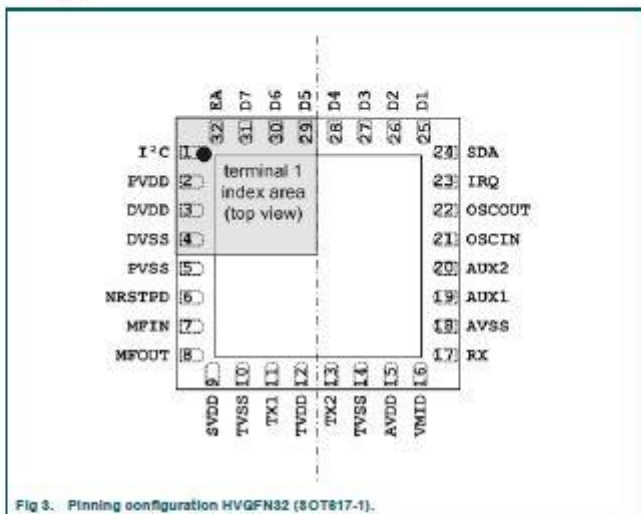


Fig 3. Pinning configuration HVGFN32 (SOT817-1).

7.2 Pin description

Table 3: Pin description

Symbol	Pin	Type	Description
I ² C	1	I	I ² C enable
FVDD	2	PWR	Pad power supply
DVDD	3	PWR	Digital Power Supply
DVSS	4	PWR	Digital Ground
FVSS	5	PWR	Pad power supply ground
NRSTPD	6	I	Not Reset and Power-down: When LOW, internal current sinks are switched off, the oscillator is inhibited, and the input pads are disconnected from the outside world. With a positive edge on this pin the internal reset phase starts.
MFIN	7	I	Mifare Signal Input
MFOUT	8	O	Mifare Signal Output
SVDD	9	PWR	MFIN / MFOUT Pad Power Supply: provides power to for the MFIN / MFOUT pads
TVSS	10, 14	PWR	Transmitter Ground: supplies the output stage of TX1 and TX2

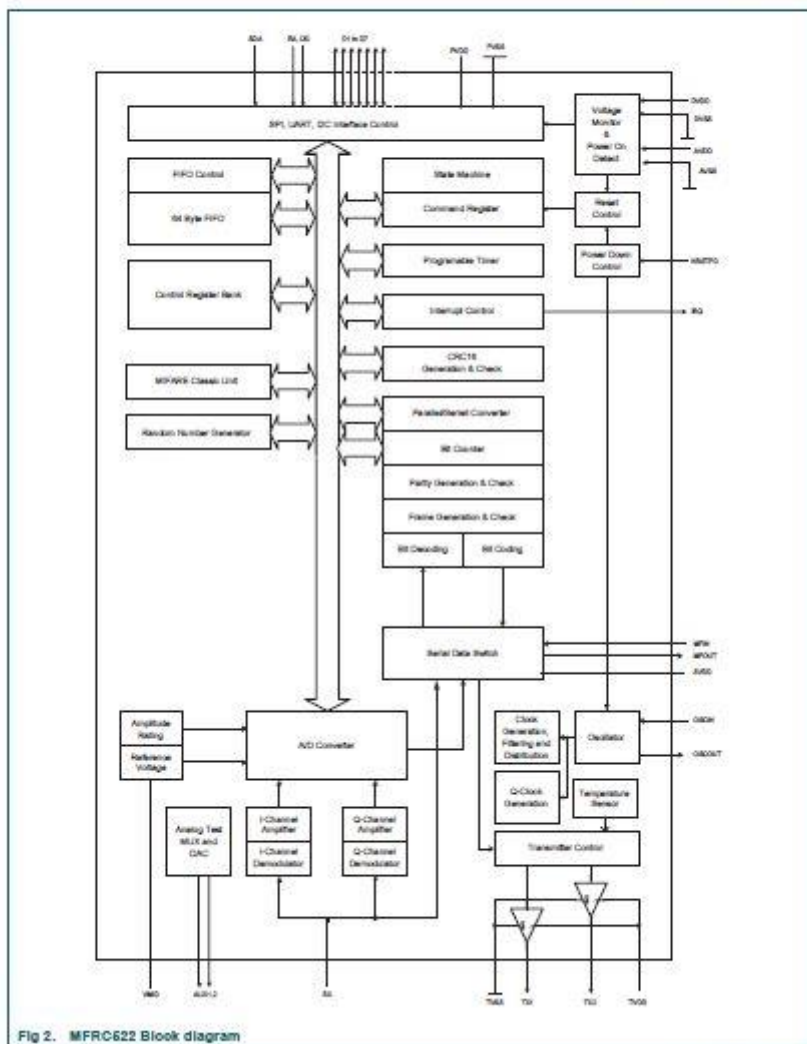


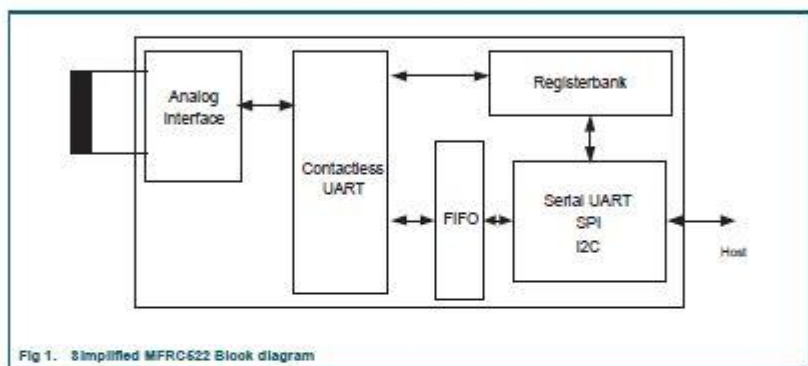
Fig 2. MFRC522 Block diagram

6. Block diagram

The Analog Interface handles the modulation and demodulation of the analog signals.

The contactless UART handles the protocol requirements for the communication schemes in co-operation with the host. The comfortable FIFO buffer allows a fast and convenient data transfer from the host to the contactless UART and vice versa.

Various host interfaces are implemented to fulfill different customer requirements.



5. Ordering information

Table 2: Ordering information

Type number	Package		Version
	Name	Description	
MFRC52201HN1/TRAYB (delivered in 1 Tray)	HVQFN32	see Package Outline in Figure 33 "Package outline package version HVQFN32" , see Packing information in Figure 34 "Packing information 1 Tray"	SOT617-1
MFRC52201HN1/TRAYBM (delivered in 5 Tray)	HVQFN32	see Package Outline in Figure 33 "Package outline package version HVQFN32" , see Packing information in Figure 35 "Packing information 5Tray"	SOT617-1

4. Quick reference data

Table 1. Quick reference data

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Supply Voltage	$V_{SS} = DV_{DD} = PV_{DD} = TV_{DD} = 0\text{ V}$, $PV_{DD} \leq AV_{DD} = DV_{DD} = TV_{DD}$	[1] 2.5	-	3.6	V
DV_{DD}			[1]			
TV_{DD}			[1]			
PV_{DD}	Pad power supply	$V_{SS} = DV_{DD} = PV_{DD} = TV_{DD} = 0\text{ V}$, $PV_{DD} \leq AV_{DD} = DV_{DD} = TV_{DD}$	[2] 1.6	-	3.6	V
SV_{DD}	MFIMFOUT Pad Power Supply	$V_{SS} = DV_{DD} = PV_{DD} = TV_{DD} = 0\text{ V}$,	1.6	-	3.6	V
I_{EPD}	Hard Power-down Current	$V_{DD} = DV_{DD} = TV_{DD} = PV_{DD} = 3\text{ V}$, RESET = LOW	[3] -	-	5	μA
I_{SPD}	Soft Power-down Current	$V_{DD} = DV_{DD} = TV_{DD} = PV_{DD} = 3\text{ V}$, RF level detector on	[3] -	-	10	μA
I_{DD}	Digital Supply Current	$DV_{DD} = 3\text{ V}$	-	6.5	9	mA
I_{AVDD}	Analog Supply Current	$V_{DD} = 3\text{ V}$, bit RCVOFF = 0	-	7	10	mA
$I_{AVDD,RCVOFF}$	Analog Supply Current, receiver switched off	$V_{DD} = 3\text{ V}$, bit RCVOFF = 1	-	3	5	mA
I_{PVDD}	Pad Supply Current		[4] -	-	40	mA
I_{TVDD}	Transmitter Supply Current	Continuous Wave	[5] -	60	100	mA
T_{amb}	operating ambient temperature		-25		+85	$^{\circ}\text{C}$

[1] Supply voltage below 3 V reduces the performance (e.g. the achievable operating distance).

[2] V_{DD} , DV_{DD} and TV_{DD} shall always be on the same voltage level.

[3] PV_{DD} shall always be on the same or lower voltage level than DV_{DD} .

[4] I_{PVDD} depends on TV_{DD} and the external circuitry connected to Tx1 and Tx2.

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

[6] During operation with a typical circuitry the overall current is below 100 mA.

[7] I_{EPD} and I_{SPD} are the total currents over all supplies.

[8] Typical value using a complementary driver configuration and an antenna matched to 40 Ω between Tx1 and Tx2 at 13.56 MHz.

- ◆ SPI interface up to 10 Mbit/s
- ◆ I²C interface up to 400 kbit/s in Fast mode, up to 3400 kbit/s in High-speed mode
- ◆ serial UART in different transfer speeds up to 1228.8 kbit/s, framing according to the RS232 interface with voltage levels according pad voltage supply
- Comfortable 64 byte send and receive FIFO-buffer
- Flexible Interrupt modes
- Hard reset with low power function
- Power-down mode per software
- Programmable timer
- Internal oscillator to connect 27.12 MHz quartz
- 2.5 - 3.3 V power supply
- CRC Co-processor
- Free programmable I/O pins
- Internal self test

MFRC522

Contactless Reader IC

Rev. 3.2 — 22 May 2007

112132

Product data sheet
PUBLIC INFORMATION

1. Introduction

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

2. General description

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

The MFRC522's internal transmitter part is able to drive a reader/writer antenna designed to communicate with ISO/IEC 14443A/MIFARE® cards and transponders without additional active circuitry. The receiver part provides a robust and efficient implementation of a demodulation and decoding circuitry for signals from ISO/IEC 14443A/MIFARE® compatible cards and transponders. The digital part handles the complete ISO/IEC 14443A framing and error detection (Parity & CRC). The MFRC522 supports MIFARE®Classic (e.g. MIFARE® Standard) products. The MFRC522 supports contactless communication using MIFARE® higher transfer speeds up to 848 kbit/s in both directions.

Various host interfaces are implemented:

- SPI interface
- serial UART (similar to RS232 with voltage levels according pad voltage supply)
- I²C interface.

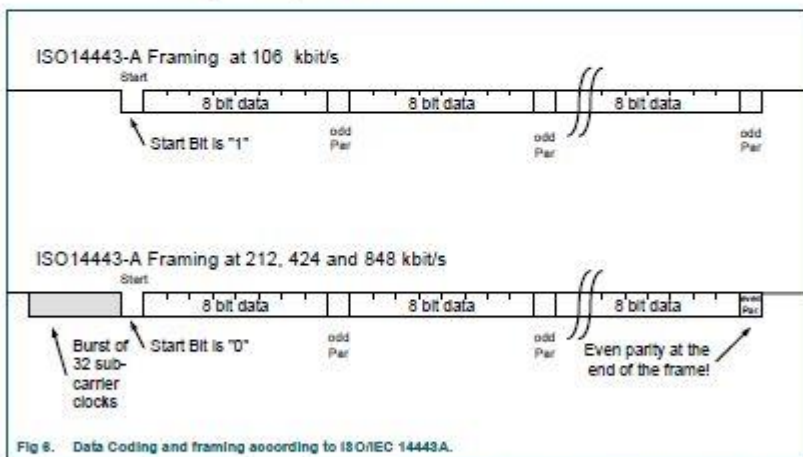
3. Features

- Highly integrated analog circuitry to demodulate and decode responses
- Buffered output drivers to connect an antenna with minimum number of external components
- Supports ISO/IEC 14443A / MIFARE®
- Typical operating distance in Reader/Writer mode for communication to a ISO/IEC 14443A / MIFARE® up to 50 mm depending on the antenna size and tuning
- Supports MIFARE® Classic encryption in Reader/Writer mode
- Supports ISO/IEC 14443A higher transfer speed communication up to 848 kbit/s
- Support of the MFIN / MFOUT
- Additional power supply to directly supply the smart card IC connected via MFIN / MFOUT
- Supported host interfaces

Table 4: Communication overview for ISO/IEC 14443A/MIFARE® reader/writer ...continued

Communication direction	transfer speed	ISO/IEC 14443A/ MIFARE®			
		106 kbit/s	212 kbit/s	424 kbit/s	848 kbit/s
Card → Reader (MFRC522 receives data from a card)	modulation on card side	subcarrier load modulation	subcarrier load modulation	subcarrier load modulation	subcarrier load modulation
	subcarrier frequency	13.56 MHz/16	13.56MHz/16	13.56MHz/16	13.56MHz/16
	bit coding	Manchester coding	BPSK	BPSK	BPSK

The contactless UART of MFRC522 and a dedicated external host are required to handle the complete MIFARE® / ISO/IEC 14443A / MIFARE® protocol. The following [Figure 6](#) "Data Coding and framing according to ISO/IEC 14443A," shows the Data Coding and framing according to ISO/IEC 14443A / MIFARE®.



The internal CRC co-processor calculates the CRC value according to the definitions given in the ISO/IEC 14443A part 3 and handles parity generation internally according to the transfer speed. Automatic parity generation can be switched off by bit ParityDisable in register 0x1D ManualRCVReg.

LAMPIRAN B

```
#include <LiquidCrystal_I2C.h>

#include <SPI.h>

#include <MFRC522.h>

#include <Wire.h>

#include "RTCLib.h"

#define SS_PIN 53

#define RST_PIN 5

RTC_DS3231 rtc;

char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday"};

LiquidCrystal_I2C lcd(0x3F, 20, 4);

MFRC522 mfrc522(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

byte card_ID[7]; //card UID size 4byte

String Name1 = "04 78 2B 7A 14 2A 80"; //kartu 28 TAMA
```

String Name2 = "04 45 8E 3A 81 5B 80"; //kartu 29 ALFAN

String Name3 = "04 87 50 8A FE 4D 80"; //kartu 30 MOH. AJI SURYA MAHARDIKA

String Name4 = "04 40 5A 32 4B 57 80"; //kartu 31 FERIAN BAGUS KUSUMA

String Name5 = "04 38 70 7A 18 5B 80"; //kartu 32 JUAN ALFREDO KURNIAWAN

String Name6 = "04 73 85 2A 9C 56 80"; //kartu 33 PANDU DINATA AL AMIRI

String Name7 = "04 52 1A DA 70 5B 80"; //kartu 34 ARGON LUTHFAN GHAZ

String Name8 = "04 4F 63 A2 F4 57 80"; //kartu 35 FARIZ AKBAR FIRMANSYAH

String Name9 = "04 30 67 1A 3D 5C 80"; //kartu 36 KHAFIT IMRON MUZAKI

String Name10 = "04 71 4E 72 9E 4F 80"; //kartu 37 MUCH ARIF RAHMAN HAKIM

String Name11 = "04 5B 81 62 9A 53 80"; //kartu 38 MOCHAMAD RAIHAN DWI WARDHANA

String Name12 = "04 40 48 52 3A 58 80"; //kartu 39 DZULFIKAR ATS TSAURI

String Name13 = "04 63 6B B2 E5 5B 80"; //kartu 40 AHMAD RIZAL FIRMANSYAH

String Name14 = "04 74 22 0A C7 61 80"; //kartu 41 DHIMAS ALFAUZI

String Name15 = "04 80 6A F2 95 4C 80"; //kartu 42 MUHAMMAD BACHROT TAMA

```
String Name16 = "04 2C 79 62 E8 5B 80"; //kartu 43 PAK FAUZI  
IMADUDDIN ADHIM
```

```
String Name17 = "04 77 7C 6A E6 5B 80"; //kartu 44 PAK CIPTIAN  
WERIED PRIANANDA
```

```
String Name18 = "04 60 5D 4A D0 2E 80"; //kartu 45 Aldo
```

```
int NumbCard[18];//the number of cards. in my case i have just two  
cards.
```

```
int j = 0;
```

```
int statu[18];//the number of cards. in my case i have just two cards.
```

```
int s = 0;
```

```
int const LED_R = 7;
```

```
int const LED_G = 6;
```

```
int const Relay = 4;
```

```
String Log;
```

```
String Name;//user name
```

```
long Number;//user number
```

```
int n ;//The number of card you want to detect (optional)
```

```
void(*ku_reset)
```

```

(void) = 0;

void setup() {

    lcd.init();

    lcd.backlight();

    Serial.begin(115200); // Initialize serial communications with the PC
    Serial1.begin(115200);
    SPI.begin(); // Init SPI bus
    mfrc522.PCD_Init(); // Init MFRC522 card

    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    pinMode(48, INPUT_PULLUP);

    Serial.println("CLEAR SHEET");           // clears starting at row 1
    Serial.println("Date,Name,Number,Time IN,Time OUT");// make
four columns (Date,Time,[Name:"user name"]line 48 &
52,[Number:"user number"]line 49 & 53)

    pinMode(LED_R, OUTPUT);
    pinMode(LED_G, OUTPUT);
    pinMode(Relay, OUTPUT);

    lcd.setCursor(4, 0);
    lcd.print("Laboratorium");

```



```

lcd.setCursor(0, 1);

lcd.print("Elektronika Terapan");

lcd.setCursor(1, 2);

lcd.print("Silahkan Scan Kartu");

lcd.setCursor(1, 3);

lcd.print("Anda :)");

delay(200);

digitalWrite (Relay, HIGH);

delay(200);

}

void loop() {

  DateTime now = rtc.now();

  if (now.hour()==24){

    ku_reset();

  }

  //look for new card

  if ( ! mfrc522.PICC_IsNewCardPresent()) {

    return;//got to start of loop if there is no card present

  }

  // Select one of the cards

```

```

if ( ! mfrc522.PICC_ReadCardSerial() ) {
    return;//if read card serial(0) returns 1, the uid struct contains the ID
of the read card.
}
String content = "";
byte letter;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
}

content.toUpperCase();

if (content.substring(1) == Name1)
{
    Name = "Tama"; //user name
    Number = 1011; //user number
    j = 0;
    s = 0;
    digitalWrite(Relay, LOW);
    digitalWrite(LED_G, HIGH);
}

```

```
    delay(3000);  
    digitalWrite(Relay, HIGH);  
  
    }  
  
    else if (content.substring(1) == Name2)  
    {  
        Name = "ALFAN"; //user name  
        Number = 1012; //user number  
        j = 1;  
        s = 1;  
        digitalWrite(Relay, LOW);  
        digitalWrite(LED_G, HIGH);  
        delay(3000);  
        digitalWrite(Relay, HIGH);  
  
    }  
  
    else if (content.substring(1) == Name3) {  
        Name = "M. AJI SURYA"; //user name  
        Number = 1013; //user number  
        j = 2;
```

```
s = 2;

digitalWrite(Relay, LOW);
digitalWrite(LED_G, HIGH);
delay(3000);
digitalWrite(Relay, HIGH);

}

else if (content.substring(1) == Name4) {
    Name = "FERIAN BAGUS"; //user name
    Number = 1014; //user number
    j = 3;
    s = 3;
    digitalWrite(Relay, LOW);
    digitalWrite(LED_G, HIGH);
    delay(3000);
    digitalWrite(Relay, HIGH);

}

else if (content.substring(1) == Name5) {
    Name = "JUAN ALFREDO"; //user name
```

```
Number = 1015; //user number

j = 4;
s = 4;
digitalWrite(Relay, LOW);
digitalWrite(LED_G, HIGH);
delay(3000);
digitalWrite(Relay, HIGH);

}

else if (content.substring(1) == Name6) {
    Name = "PANDU DINATA"; //user name
    Number = 1016; //user number
    j = 5;
    s = 5;
    digitalWrite(Relay, LOW);
    digitalWrite(LED_G, HIGH);
    delay(3000);
    digitalWrite(Relay, HIGH);

}
```

```
else if (content.substring(1) == Name7) {  
    Name = "ARGON LUTHFAN"; //user name  
    Number = 1017; //user number  
    j = 6;  
    s = 6;  
    digitalWrite(Relay, LOW);  
    digitalWrite(LED_G, HIGH);  
    delay(3000);  
    digitalWrite(Relay, HIGH);  
  
}
```

```
else if (content.substring(1) == Name8) {  
    Name = "FARIZ AKBAR"; //user name  
    Number = 1018; //user number  
    j = 7;  
    s = 7;  
    digitalWrite(Relay, LOW);  
    digitalWrite(LED_G, HIGH);  
    delay(3000);  
    digitalWrite(Relay, HIGH);
```

```
}
```

```
else if (content.substring(1) == Name9) {  
    Name = "KHAFIT IMRON"; //user name  
    Number = 1019; //user number  
    j = 8;  
    s = 8;  
    digitalWrite(Relay, LOW);  
    digitalWrite(LED_G, HIGH);  
    delay(3000);  
    digitalWrite(Relay, HIGH);  
}
```

```
else if (content.substring(1) == Name10) {  
    Name = "MUCH ARIF RAHMAN"; //user name  
    Number = 1020; //user number  
    j = 9;  
    s = 9;  
    digitalWrite(Relay, LOW);  
    digitalWrite(LED_G, HIGH);  
    delay(3000);
```

```
digitalWrite(Relay, HIGH);

}

else if (content.substring(1) == Name11) {
    Name = "MOCHAMAD RAIHAN"; //user name
    Number = 1021; //user number
    j = 10;
    s = 10;
    digitalWrite(Relay, LOW);
    digitalWrite(LED_G, HIGH);
    delay(3000);
    digitalWrite(Relay, HIGH);

}

else if (content.substring(1) == Name12) {
    Name = "DZULFIKAR"; //user name
    Number = 1022; //user number
    j = 11;
    s = 11;
    digitalWrite(Relay, LOW);
```



```
digitalWrite(LED_G, HIGH);  
delay(3000);  
digitalWrite(Relay, HIGH);  
  
}  
  
else if (content.substring(1) == Name13) {  
    Name = "AHMAD RIZAL"; //user name  
    Number = 1023; //user number  
    j = 12;  
    s = 12;  
    digitalWrite(Relay, LOW);  
    digitalWrite(LED_G, HIGH);  
    delay(3000);  
    digitalWrite(Relay, HIGH);  
  
}  
  
else if (content.substring(1) == Name14) {  
    Name = "DHIMAS ALFAUZI"; //user name  
    Number = 1024; //user number  
    j = 13;
```

```
s = 13;

digitalWrite(Relay, LOW);
digitalWrite(LED_G, HIGH);
delay(3000);
digitalWrite(Relay, HIGH);

}

else if (content.substring(1) == Name15) {
    Name = "MUHAMMAD BACHROT"; //user name
    Number = 1025; //user number
    j = 14;
    s = 14;
    digitalWrite(Relay, LOW);
    digitalWrite(LED_G, HIGH);
    delay(3000);
    digitalWrite(Relay, HIGH);

}

else if (content.substring(1) == Name16) {
    Name = "PAK FAUZI"; //user name
```

```
Number = 1026; //user number

j = 15;

s = 15;

digitalWrite(Relay, LOW);

digitalWrite(LED_G, HIGH);

delay(3000);

digitalWrite(Relay, HIGH);

}

else if (content.substring(1) == Name17) {

Name = "PAK CIPTIAN"; //user name

Number = 1027; //user number

j = 16;

s = 16;

digitalWrite(Relay, LOW);

digitalWrite(LED_G, HIGH);

delay(3000);

digitalWrite(Relay, HIGH);

}
```

```
else if (content.substring(1) == Name18) {  
    Name = "Aldo"; //user name  
    Number = 1028; //user number  
    j = 17;  
    s = 17;  
    digitalWrite(Relay, LOW);  
    digitalWrite(LED_G, HIGH);  
    delay(3000);  
    digitalWrite(Relay, HIGH);  
  
}
```

```
else
```

```
{  
  
    lcd.clear();  
    lcd.setCursor(4, 2);  
    lcd.print("Akses Ditolak");  
    digitalWrite(Relay, HIGH);  
    digitalWrite(LED_R, HIGH);  
    delay(500);  
    lcd.clear();
```

```
lcd.setCursor(4, 0);  
lcd.print("Laboratorium");  
lcd.setCursor(0, 1);  
lcd.print("Elektronika Terapan");  
lcd.setCursor(1, 2);  
lcd.print("Silahkan Scan Kartu");  
lcd.setCursor(1, 3);  
lcd.print("Anda :)");  
goto cont;//go directly to line 71  
  
}
```

```
// DateTime now = rtc.now();
```

```
if (NumbCard[j] == 1 && statu[s] == 0) {  
    statu[s] = 1;  
    lcd.clear();  
    lcd.setCursor(4, 0);  
    lcd.print("Laboratorium");  
    lcd.setCursor(0, 1);
```

```
lcd.print("Elektronika Terapan");  
lcd.setCursor(4, 2);  
lcd.print("Welcome ");  
lcd.setCursor(0, 3);  
lcd.print(Name);  
delay(2000);
```

```
Serial.print(now.year(), DEC);  
Serial1.print(now.year(), DEC);
```

```
Serial.print('/');  
Serial1.print('/');
```

```
Serial.print(now.month(), DEC);  
Serial1.print(now.month(), DEC);
```

```
Serial.print('/');  
Serial1.print('/');
```

```
Serial.print(now.day(), DEC);  
Serial1.print(now.day(), DEC);
```

```
Serial.print('-');
```

```
Serial1.print('-');
```

```
Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
```

```
Serial1.println(daysOfTheWeek[now.dayOfTheWeek()]);
```

```
Serial.print(",");
```

```
Serial.print(Name);
```

```
Serial1.println(Name);
```

```
Serial.print(",");
```

```
Serial.print(Number);
```

```
Serial1.println(Number);
```

```
Serial.print(",");
```

```
Serial.print("kosong");
```

```
Serial1.println("kosong");
```

```
Serial.print(",");
```

```
Serial.print(now.hour(), DEC);  
Serial1.print(now.hour(), DEC);
```

```
Serial.print(':');  
Serial1.print(':');
```

```
Serial.print(now.minute(), DEC);  
Serial1.print(now.minute(), DEC);
```

```
Serial.print(':');  
Serial1.print(':');
```

```
Serial.print(now.second(), DEC);  
Serial1.print(now.second(), DEC);
```

```
Serial.println("");  
Serial1.println("");  
}
```



```
else if (NumbCard[j] == 1 && statu[s] == 1) {  
    statu[s] = 1;  
    lcd.clear();  
    lcd.setCursor(4, 0);  
    lcd.print("Laboratorium");  
    lcd.setCursor(0, 1);  
    lcd.print("Elektronika Terapan");  
    lcd.setCursor(4, 2);  
    lcd.print("Welcome ");  
    lcd.setCursor(0, 3);  
    lcd.print(Name);  
    digitalWrite(LED_G, HIGH);  
    delay(2000);  
  
    Serial.print(now.year(), DEC);  
    Serial1.print(now.year(), DEC);  
  
    Serial.print('/');  
    Serial1.print('/');  
  
    Serial.print(now.month(), DEC);
```

```
Serial1.print(now.month(), DEC);
```

```
Serial.print('/');
```

```
Serial1.print('/');
```

```
Serial.print(now.day(), DEC);
```

```
Serial1.print(now.day(), DEC);
```

```
Serial.print('-');
```

```
Serial1.print('-');
```

```
Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
```

```
Serial1.println(daysOfTheWeek[now.dayOfTheWeek()]);
```

```
Serial.print(",");
```

```
Serial.print(Name);
```

```
Serial1.println(Name);
```

```
Serial.print(",");
```

```
Serial.print(Number);
```

```
Serial1.println(Number);
```

```
Serial.print(",");
```

```
Serial.print("kosong");
```

```
Serial1.println("kosong");
```

```
Serial.print(",");
```

```
Serial.print(now.hour(), DEC);
```

```
Serial1.print(now.hour(), DEC);
```

```
Serial.print(':');
```

```
Serial1.print(':');
```

```
Serial.print(now.minute(), DEC);
```

```
Serial1.print(now.minute(), DEC);
```

```
Serial.print(':');
```

```
Serial1.print(':');
```

```
Serial.print(now.second(), DEC);
```

```
Serial1.print(now.second(), DEC);
```

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

```
Serial1.println("");
```

```
}
```

```
else if (NumbCard[j] == 0) {
```

```
    NumbCard[j] = 1;
```

```
    n++; //(optional)
```

```
    lcd.clear();
```

```
    lcd.setCursor(4, 0);
```

```
    lcd.print("Laboratorium");
```

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

```
    lcd.print("Elektronika Terapan");
```

```
    lcd.setCursor(4, 2);
```

```
    lcd.print("Welcome ");
```

```
    lcd.setCursor(0, 3);
```

```
    lcd.print(Name);
```

```
    digitalWrite(LED_G, HIGH);
```

```
    delay(2000);
```

```
Serial.print(now.year(), DEC);
```

```
Serial1.print(now.year(), DEC);
```

```
Serial.print('/');
```

```
Serial1.print('/');
```

```
Serial.print(now.month(), DEC);
```

```
Serial1.print(now.month(), DEC);
```

```
Serial.print('/');
```

```
Serial1.print('/');
```

```
Serial.print(now.day(), DEC);
```

```
Serial1.print(now.day(), DEC);
```

```
Serial.print('-');
```

```
Serial1.print('-');
```

```
Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
```

```
Serial1.println(daysOfTheWeek[now.dayOfTheWeek()]);
```

```
Serial.print(",");
```

```
Serial.print(Name);
```

```
Serial.println(Name);
```

```
Serial.print(",");
```

```
Serial.print(Number);
```

```
Serial.println(Number);
```

```
Serial.print(",");
```

```
Serial.print(now.hour(), DEC);
```

```
Serial.print(now.hour(), DEC);
```

```
Serial.print(':');
```

```
Serial.print(':');
```

```
Serial.print(now.minute(), DEC);
```

```
Serial.print(now.minute(), DEC);
```

```
Serial.print(':');
```

```
Serial.print(':');
```

```
Serial.print(now.second(), DEC);  
Serial.println(now.second(), DEC);  
  
Serial.print(",");  
  
Serial.println("kosong");  
Serial.println("kosong");  
}  
  
else if (statu[s] == 1) {  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print(Name);  
    lcd.setCursor(0, 1);  
    lcd.print("Already Left");  
}  
delay(100);  
lcd.clear();  
lcd.setCursor(4, 0);  
lcd.print("Laboratorium");  
lcd.setCursor(0, 1);  
lcd.print("Elektronika Terapan");
```

```
lcd.setCursor(1, 2);  
lcd.print("Silahkan Scan Kartu");  
lcd.setCursor(1, 3);  
lcd.print("Anda :)");  
cont:  
digitalWrite(LED_G, LOW);  
digitalWrite(LED_R, LOW);  
return;  
  
//if you want to close the Excel when all card had detected and save  
Excel file in Names Folder. in my case i have just 2 card (optional)  
// if (n == 2) {  
// Serial.println("SAVEWORKBOOKAS,Names/WorkNames");  
// Serial.println("FORCEEXCELQUIT");  
  
}
```


DAFTAR RIWAYAT PENULIS



Nama : Pratanu
Benwiramadhan
TTL : Kediri, 21 Januari 1998
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Jl. KH Ahmad Dahlan
no.147 Mojoroto, Kediri
Telp/HP : 0856-0413-0696
E-mail : pratamanub@gmail.com

RIWAYAT PENDIDIKAN

1. 2004 – 2010 : SDN Mojoroto 3 Kediri
2. 2010 – 2013 : SMP Negeri 8 Kediri
3. 2013 – 2016 : SMA Negeri 7 Kediri
4. 2016 – 2019 : Departemen Teknik Elektro Otomasi - Fakultas
Vokasi - Institut Teknologi Sepuluh Nopember
(ITS)

PENGALAMAN KERJA

1. Magang di PT. PAL Indonesia (Persero) Surabaya

PENGALAMAN ORGANISASI

1. Staff Kesejahteraan Mahasiswa HIMAD3TEKTRO

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