

**PEMANFAATAN RFID DALAM PROTOTIPE TIKET
STADION SEPAK BOLA BERBASIS ARDUINO UNO**



**MOHAMAD IDRIS
5215111741**

**Skripsi ini Ditulis untuk Memenuhi Sebagian Persyaratan
dalam Memperoleh Gelar Sarjana**

**PROGRAM STUDI PENDIDIKAN TEKNIK ELEKTRONIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI JAKARTA
2016**


LEMBAR PENGESAHAN

NAMA

TANDA TANGAN

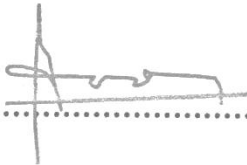
TANGGAL

Drs. Jusuf Bintoro, MT.
(Dosen Pembimbing I)



02/02/2016

Aodah Diamah, ST, M.Eng.
(Dosen Pembimbing II)



26/01/2016

PENGESAHAN PANITIA UJIAN SIDANG

NAMA

TANDA TANGAN

TANGGAL

Drs. Wisnu Djatmiko, MT.
(Ketua Sidang)



01/02/2016

Drs. Pitoyo Yuliatmojo, MT.
(Sekretaris)



29/01/2016

Drs. Mufti Ma'sum, M.Pd.
(Dosen Ahli)



29/01/2016

Tanggal Lulus :

LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Karya tulis skripsi saya yang berjudul “Pemanfaatan RFID Dalam Prototipe Tiket Stadion Sepak Bola Berbasis Arduino Uno” ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan, dan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai norma yang berlaku di Universitas Negeri Jakarta

Jakarta, Januari 2016

Yang Membuat Pernyataan

Mohamad Idris

5215111741

ABSTRAK

Muhamad Idris, *Pemanfaatan RFID Dalam Prototipe Tiket Stadion Sepak Bola Berbasis Arduino Uno*. Skripsi. Jakarta, Program Studi Pendidikan Teknik Elektronika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta, 2015. Dosen Pembimbing, Drs. Jusuf Bintoro, MT dan Aodah Diamah, ST., M., Eng.

Tujuan pembuatan penelitian ini adalah membuat prototipe yang dapat diaplikasikan sebagai gate stadion sepak bola dengan memanfaatkan RFID tag dalam bentuk kartu sebagai E-Tiket. Diharapkan alat ini akan meningkatkan epektivitas registrasi penonton stadion sepak agar masuk sesuai dengan tiket yang dimiliki. dan dapat merekap data pengunjung stadion.

Penelitian ini dilakukan menggunakan metode penelitian dan pengembangan (*Research and Development*) yang meliputi perencanaan, analisis kebutuhan perancangan, pengujian, dan implementasi perangkat keras (*hardware*) dan perangkat lunak (*software*). Implementasi perangkat keras yaitu pembuatan rangkaian E-Tiket, sistem pengendali menggunakan mikrokontroler Arduino UNO, inputan berupa data UID RFID dari modul pembaca RFID, output berupa tampilan LCD, LED dan Servo pada gate serta tampilan pada perangkat lunak server tiketing yang dibuat untuk mengolah data dengan mengkomunikasikan Arduino dengan PC.

Hasil penelitian ini menunjukan prototipe alat tiketing elektronik menggunakan RFID Berbasis Arduino Uno dengan bahasa pemrograman Arduino melalui komunikasi serial menggunakan kabel data dengan server tiketing yang telah dirancang peneliti menggunakan perangkat lunak visual basic 2010 dan microsoft Acces 2010, sehingga memiliki keunggulan antara lain data pengunjung tersimpan di dalam database microsoft Acces 2010, mencegah pengunjung yang tidak memiliki tiket masuk ke dalam stadion.

Kata kunci : Tiketing, Arduino, RFID, Database

KATA PENGANTAR

Puji syukur kehadiran Allah swt. yang telah melimpahkan rahmat, inayah, taufik, dan hidayahnya sehingga saya dapat menyelesaikan penyusunan skripsi yang berjudul “Pemanfaatan RFID Dalam Prototipe Tiket Stadion Sepak Bola Berbasis Arduino Uno”. Semoga penelitian skripsi ini dapat dipergunakan sebagai salah satu acuan, petunjuk maupun pedoman bagi pembaca.

Penyusunan skripsi ini dilakukan untuk memenuhi salah satu syarat untuk mencapai gelar Sarjana Pendidikan Teknik Elektronika FT UNJ. Peneliti menyadari bahwa penyusunan skripsi ini masih belum sempurna sehingga peneliti membutuhkan kritik dan saran untuk membangun penyempurnaan skripsi ini. Oleh karena itu, Peneliti mengucapkan terima kasih kepada:

1. Pitoyo Yuliatmojo, MT, selaku Ketua Program Studi FT UNJ.
2. Drs. Jusuf Bintoro. MT, selaku Dosen Pembimbing I.
3. Aodah Diamah, ST., M, Eng, selaku Dosen Pembimbing II
4. Drs. Wisnu Djatmiko, MT, selaku Pembimbing Akademik.
5. Kedua orang tua dan keluarga saya yang telah memberikan semangat serta doa yang tidak pernah terhenti diucapkan untuk kelancaran dan keberhasilan.

Akhir kata, semoga Allah Subhanahu wa Ta’ala membalas segala kebaikannya

Peneliti

Mohamad Idris

5215111741

DAFTAR ISI

	Halaman
ABSTRAK	i
KATA PENGANTAR	ii
Daftar isi	iii
Daftar Gambar.....	vii
Daftar Tabel	viii
BAB I	1
PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Identifikasi Masalah.....	2
1.3. Pembatasan Masalah.....	2
1.4. Perumusan Masalah	3
1.5. Tujuan Penelitian	3
1.6. Kegunaan Penelitian	3
BAB II	4
KERANGKA TEORETIK, KERANGKA BERFIKIR DAN HIPOTESIS	
PENELITIAN	4
2.1. Kerangka Teoritik	4
2.1.1. Prototipe.....	4
2.1.2. E-Tiketing	4
2.1.3. Arduino	5
2.1.4. LCD Philips PCF8833	7
2.1.5. RFID	8
2.1.4.1 RFID Tag.....	8
2.1.6. Visual Basic	14
2.1.7. <i>Microsoft Access</i>	15

2.1.8. Database Tiketing	15
2.1.9. Tiket Stadion Sepak Bola Berbasis Arduino UNO	16
2.2. Kerangka Berfikir	17
2.2.1. Blok Diagram	17
2.2.2. <i>Flowchart</i> Sistem Kerja Pembelian Tiket Stadion Elektronik.....	19
BAB III	20
METODE PENELITIAN.....	20
3.1. Tempat dan Waktu Penelitian	20
3.2. Metodologi Penelitian.....	20
3.2.1 Pengumpulan Data.....	21
3.2.2 Desain Produk.....	21
3.2.3 Uji Coba Produk	22
3.2.4 Revisi Produk	22
3.3. Rancangan Penelitian.....	23
3.3.1 Menentukan Diagram Blok Sistem.....	23
3.3.2 Perencanaan Desain Alat	24
3.3.3 Perancangan Perangkat Keras.....	25
3.3.4 Perancangan Integrasi komponen	25
3.3.5 Perencanaan Perangkat Lunak	28
3.3.6 Perancangan Program Arduino untuk Arduino Board.....	28
3.3.7 Perancangan Program VB.Net untuk membangun server tiketing.....	29
3.4. Instrumen Penelitian	30
3.5. Prosedur Penelitian	31
3.6. Pengujian Alat.....	32

3.6.1.1	Kriteria Pengujian <i>Hardware</i> dan <i>Software</i>	32
3.6.1.1	Pengujian Pembacaan Kartu e-Ticket	32
3.6.1.2	Pengujian penyimpanan data E-tiket Langganan	32
3.6.1.3	Pengujian pengeditan data E-tiket Langganan	33
3.6.1.4	Pengujian penambahan saldo E-tiket Langganan.....	33
3.6.1.5	Pengujian penyimpanan data E-tiket Harian.....	34
3.6.1.6	Pengujian pengeditan harga tiket	34
3.6.1.7	Pengujian gate saldo E-tiket langganan mencukupi.	34
3.6.1.8	Pengujian gate saldo E-tiket langganan tidak mencukupi	35
3.6.1.9	Pengujian gate E-tiket harian sesuai dengan kelas...	35
3.6.1.10	Pengujian gate E-tiket harian sesuai dengan kelas...	36
3.7.	Spesifikasi Sistem.	36
BAB IV	37
HASIL PENELITIAN	37
4.1.	Hasil Penelitin.....	37
4.1.1.	Hasil Pengujian dan Pembuatan Hardware.....	37
4.1.1.1.	Hasil dan Pembuatan Alat	37
4.1.2.	Hasil Pengujian Software Tiketing	39
4.1.2.1.	Hasil Pengujian Pembacaan ID E-Ticket	39
4.1.2.2.	Pengujian penyimpanan data E-Tiket Langganan....	41
4.1.2.3.	Pengujian pengeditan data E-tiket Langganan	42
4.1.2.4.	Pengujian penambahan saldo E-tiket Langganan.....	43
4.1.2.5.	Pengujian penyimpanan data E-tiket Harian.....	44
4.1.2.6.	Pengujian pengeditan harga tiket	45
4.1.2.7.	Pengujian gate saldo E-tiket langganan mencukupi.	46

4.1.2.8. Pengujian gate saldo E-tiket langganan tidak mencukupi	49
4.1.2.9. Pengujian gate E-tiket harian sesuai dengan kelas...	51
4.1.2.10. Pengujian gate E-tiket harian sesuai dengan kelas...	53
4.2. Kelebihan dan Kekurangan Alat	55
BAB V KESIMPULAN DAN SARAN	57
1.1 Kesimpulan.....	57
1.2 Saran.....	58
Daftar Pustaka	59
Lampiran	60

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 E-tiket	5
Gambar 2. 2 Arduino Uno	6
Gambar 2. 3 LCD PCF8833	7
<i>Gambar 2. 4</i> RFID Tag	9
Gambar 2. 5 Tag pasif	10
Gambar 2. 6 Tag aktif.....	11
Gambar 2. 7 modul pembaca RFID RC522	14
Gambar 2. 8 Tampilan form Visual Basic 2010.....	15
Gambar 2. 9 Blok diagram sistem tiketing elektronik.....	18
Gambar 2. 10 Diagram alir sistem masuk Stadion melalui Gate elektronik.....	19
Gambar 3. 1 Tahap-tahap metodologi penelitian perancangan sistem tiketing elektronik.....	21
Gambar 3. 2 Blok diagram sistem Tiketing Stadion	23
Gambar 3. 3 Desain Alat Gate Stadion.....	24
Gambar 3. 4 Integrasi modul RFID MFRC522 dengan Arduino UNO	25
Gambar 3. 5 Integrasi LCD dengan Arduino UNO.....	26
Gambar 3. 6 Skematik Keseluruhan alat e-ticket	27
Gambar 3. 7 Layout PCB e-ticket.....	28
Gambar 4. 1 casing Gate stadion	38
Gambar 4. 2 Papan Rangkaian E-Ticket	38
Gambar 4. 3 Hasil akhir pembuatan E-Tiket.....	39
Gambar 4. 4 Proses pengujian pembacaan ID E-Tiket	40
Gambar 4. 5 Data tersimpan pada database Pengunjung Langganan	42
Gambar 4. 6 pengeditan pada database Pengunjung Langganan.....	43
Gambar 4. 7 edit saldo langganan	44
Gambar 4. 8 pengunjung Harian.....	45
Gambar 4. 9 edit harga tiket	46

DAFTAR TABEL

	Halaman
Tabel 3. 1 Penggunaan Pin Input pada Arduino dengan Perangkat Input	29
Tabel 3. 2 Penggunaan Pin Output pada Arduino dengan Perangkat Output	29
Tabel 3. 3 Pengujian Kesesuaian ID kartu dengan data kartu.....	32
Tabel 3. 4 Pengujian penyimpanan data E-tiket Langganan.....	32
Tabel 3. 5 Pengujian pengeditan data E-tiket Langganan.....	33
Tabel 3. 6 Pengujian penambahan saldo E-tiket Langganan	33
Tabel 3. 7 Pengujian penyimpanan data E-tiket Harian.....	34
Tabel 3. 8 Pengujian pengeditan harga tiket	34
Tabel 3. 9 Pengujian gate saldo E-tiket langganan mencukupi	34
Tabel 3. 10 Pengujian gate saldo E-tiket langganan tidak mencukupi	35
Tabel 3. 11 Pengujian gate E-tiket harian sesuai dengan kelas.....	35
Tabel 3. 12 Pengujian gate E-tiket harian sesuai dengan kelas.....	36
Tabel 4. 1 Hasil pengujian Kesesuaian ID kartu dengan data kartu	40
Tabel 4. 2 Pengujian penyimpanan data E-tiket Langganan.....	41
Tabel 4. 3 Pengujian pengeditan data E-tiket Langganan.....	42
Tabel 4. 4 Pengujian penambahan saldo E-tiket Langganan	43
Tabel 4. 5 Pengujian penyimpanan data E-tiket Harian.....	45
Tabel 4. 6 Pengujian pengeditan harga tiket	46
Tabel 4. 7 Pengujian gate saldo E-tiket langganan mencukupi	47
Tabel 4. 8 Pengujian gate saldo E-tiket langganan tidak mencukupi	49
Tabel 4. 9 Pengujian gate E-tiket harian sesuai dengan kelas.....	51
Tabel 4. 10 Pengujian gate E-tiket harian sesuai dengan kelas.....	53

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam sebuah mekanisme pembelian tiket stadion sepak bola para pengunjung masih mendapatkan tiket kertas yang nantinya pada saat masuk masih harus diperiksa lagi oleh petugas yang menjaga di pintu masuk stadion. Cara ini dianggap peneliti masih kurang efisien apa lagi masih menggunakan kertas dan masih memerlukan banyak petugas yang menjaga pintu masuk stadion sehingga bila terjadi penumpukan pengunjung dikhawatirkan akan terjadi kesalahan dalam pengecekan tiket yang dimiliki pengunjung.

Setiap kali pertandingan sepak bola dilaksanakan maka diharapkan banyak penonton yang datang untuk menyaksikan pertandingan langsung di stadion stadion sepak bola. Sehingga pihak pengelola stadion mendapatkan pemasukan yang dapat dipakai untuk biaya perawatan stadion dan peningkatan pelayanan penonton yang lebih baik lagi.

Seperti pada proses pembelian tiket pertandingan sepak bola di stadion yang masih menggunakan cara manual dengan menggunakan kertas, perekapan data yang masih menyulitkan petugas sehingga membutuhkan waktu yang lebih lama. Dan masih banyak penonton pertandingan di stadion khususnya pertandingan sepak bola yang berpotensi masuk tanpa menggunakan tiket sehingga dapat merugikan pihak pengelola stadion sepak bola.

Bagi pihak pengelola stadion diharapkan penonton pertandingan sepak bola di stadion dapat pelayanan yang lebih maksimal sehingga lebih memudahkan untuk melakukan registrasi pembelian tiket dapat berjalan dengan lancar. Data pengunjung stadion sepak bola sangat dibutuhkan oleh pengelola stadion agar dapat mengetahui indeks pengunjung stadion sepak bola dalam sebuah pertandingan sehingga pihak pengelola dapat mengatur dalam segi administrasi dan segi teknis untuk mengembangkan minat pengunjung stadion sepak bola.

Uraian latar belakang diatas menjadi dasar pemikiran peneliti dalam merancang sebuah karya inovatif berupa prototipe sistem tiket otomatis menggunakan RFID berbasis arduino UNO sebagai media mempermudah registrasi pembelian tiket dan mencegah terjadinya penonton yang tidak memiliki tiket dapat masuk ke dalam stadion, serta menyediakan database pengunjung stadion sepak bola agar mempermudah pihak pengelola untuk merekap data penonton dalam sebuah pertandingan.

1.2. Identifikasi Masalah

Berdasarkan uraian pada latar belakang masalah di atas dapat diidentifikasi masalah sebagai berikut :

1. Masih adanya kemungkinan pengunjung yang masuk tidak sesuai dengan tiket yang dibeli
2. Sistem penjualan tiket yang masih kurang efisien sehingga rawan terjadinya kesalahan saat perekapan data pengunjung.

1.3. Pembatasan Masalah

Tampak jelas permasalahan yang terkait dengan topik pemanfaatan RFID sebagai tiket stadion sepak bola berbasis Arduino. Sehingga Peneliti membatasi permasalahan pada alat tiketing stadion sepak bola elektronik sebagai berikut:

1. Perancangan dan pembuatan sistem kontrol menggunakan Arduino.
2. Menggunakan Modul RFID Reader dengan tipe RC522.
3. RFID tag yang digunakan adalah *high frequency tag* (13.56 Mhz) berbentuk kartu.
4. *Output* berupa tampilan di komputer/PC, *Database* dan pintu *Gate* stadion.
5. Jumlah *gate* yang dibuat hanya 3 buah

1.4. Perumusan Masalah

Bagaimana konsep membuat dan menguji prototipe sistem tiket stadion menggunakan RFID berbasis arduino UNO sebagai media pencegahan tindakan penerobosan penonton yang masuk tidak sesuai dengan tiket yang dimiliki dengan aplikasi server yang dibuat dengan Visual Basic 2010 dan menyimpan data pengunjung dalam sebuah database di microsoft Access?

1.5. Tujuan Penelitian

Sesuai dengan masalah yang telah dirumuskan dan diidentifikasi, maka tujuan penelitian kali ini untuk membuat dan menguji prototipe sistem tiket stadion menggunakan RFID berbasis arduino UNO sekaligus memudahkan dalam proses pengunjung masuk ke dalam stadion dan pengelolaan data pengunjung stadion sepak bola. Dengan bahasa pemrograman Arduino sekaligus membangun aplikasi server tiket menggunakan Visual Basic Studio 2010 dan *Database Access* sebagai perangkat pendukung sistem tiket stadion sepak bola

1.6. Kegunaan Penelitian

- a. Meningkatkan kedisiplinan penonton pertandingan sepak bola.
- b. Memudahkan perakapan data pengunjung Stadion ke server pusat.
- c. Mengurangi penggunaan kertas (*paperless*).
- d. Dapat meningkatkan kualitas pelayanan pengunjung pertandingan sepakbola di stadion.

BAB II

KERANGKA TEORETIK, KERANGKA BERFIKIR DAN HIPOTESIS PENELITIAN

2.1. Kerangka Teoritik

2.1.1. Prototipe

Model yang mula-mula (model asli) yang menjadi contoh (kbbi.web.id, prototipe). Definisi prototipe adalah model asli dari suatu produk yang menjadi contoh standar atau contoh baku. Yang artinya untuk menjelaskan bahwa sasaran sudah tercapai secara *intangible* (tidak berwujud) untuk mengenalinya dan merencanakan tindakan-tindakan yang perlu dijalankan untuk menjadikan terealisasi secara *tangible* (berwujud). Menjadikan prototipe pencapaian sasaran dari setiap detail aksi dari apa yang telah dirancang sebelumnya dari pikiran (Andress, 2008).

Dari definisi di atas penulis dapat menyimpulkan prototipe adalah sebuah model menyerupai bentuk aslinya yang berukuran lebih kecil dari sebuah produk sebagai sarana penunjang untuk presentasi dalam suatu promosi, studi dan diskusi yang diperlukan oleh pihak pembuat produk tersebut.

2.1.2. E-Ticketing

E-tiket (elektronik tiket) adalah sebuah tiket dalam bentuk digital. E-ticketing atau electronic ticketing adalah suatu cara untuk mendokumentasikan proses penjualan dari aktifitas pelanggan tanpa harus mengeluarkan dokumen berharga secara fisik ataupun paper tiket. (Kinanti, 2013)

Dari definisi di atas penulis dapat menyimpulkan E-Ticketing adalah suatu cara untuk mendokumentasikan proses penjualan dari aktifitas pelanggan menggunakan kartu elektronik sebagai media penghubung data pelanggan ke dalam database yang disimpan secara elektronik.

Seiring berkembangnya zaman, e-tiket sangat disukai oleh orang karena mereka tidak perlu lagi membawa tiket secara fisik. E-tiket kebanyakan digunakan untuk perusahaan penerbangan, perkereta apiian khususnya kereta listrik, bus trans jakarta. dimana e-tiket tersebut akan berisi tentang data penumpang . Data-data penumpang akan disimpan di sistem komputer berupa database berupa saldo, nama pemilik.

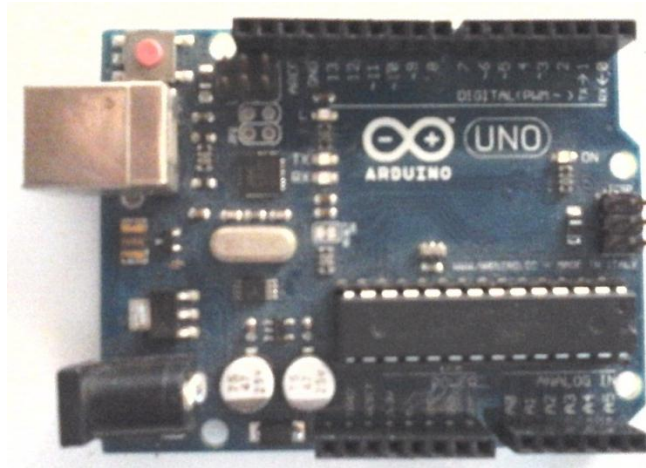


Gambar 2. 1 E-tiket

(sumber :http://www.scheidt-bachmann.com/uploads/pics/eTicket_small_web_04.jpg)

2.1.3. Arduino

Arduino adalah suatu Produk berlabel Arduino yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler Atmega328 (sebuah keping yang secara fungsional bertindak seperti sebuah komputer). Peranti ini dapat dimanfaatkan untuk mewujudkan rangkaian elektronik dari yang sederhana hingga yang kompleks. Pengendalian LED hingga pengontrolan robot dapat diimplementasikan dengan menggunakan papan yang berukuran relatif kecil ini (lihat Gambar 2.1). Bahkan, dengan penambahan komponen tertentu, peranti ini bisa dipakai untuk pemantauan jarak jauh melalui internet, misalnya pemantauan kondisi pasien di rumah sakit dan pengendalian alat-alat di rumah. (Kadir, 2013)



Gambar 2. 2 Arduino Uno

(sumber: dokumentasi pribadi)

Ada berbagai macam jenis – jenis Arduino yaitu :

1. Arduino Duemilanove
2. Arduino UNO
3. Arduino Leonardo
4. Arduino Mega2560
5. Arduino Intel Galile
6. Arduino Pro Micro AT
7. Arduino Nano R3
8. Arduino mini Atmega
9. Arduino Mega ADK
10. Arduino Esplora

Setiap arduino jenis Arduino memiliki fungsi dan penggunaan yang berbeda. Pada umumnya Arduino UNO banyak digunakan dalam mempelajari Arduino dan membuat proyek dengan menggunakan Arduino. Arduino UNO adalah papan mikrokontroler berdasarkan ATmega328. Memiliki 14 digital pin *input/output* (dimana 6 pin dapat

digunakan sebagai *output* PWM), 6 input analog, resonator keramik 16 MHZ, koneksi USB, jack listrik, header ICSP, dan tombol reset. Arduino uno dilengkapi dengan *static random-access memory* (SRAM) berukuran 2 Kb untuk memegang data, *flash memory* berukuran 32kb, dan *erasable programmable read – only memory* (EEPROM) untuk menyimpan program.

Penggunaannya sangat mudah dengan menghubungkan kabel USB dari komputer ke Arduino untuk memprogram Arduino. Perangkat lunak Arduino IDE dipublikasikan secara *open source*. Banyak contoh – contoh program yang dapat diunduh lewat *website* <http://arduino.cc/> . terdapat banyak proyek menggunakan Arduino yang dapat dipelajari di *website* tersebut.

2.1.4. LCD Philips PCF8833



Gambar 2. 3 LCD PCF8833

(sumber: dokumentasi pribadi)

LCD PCF8833 adalah LCD dengan rangkaian tunggal teknologi CMOS sebagai pengendali sehingga daya yang digunakan cukup rendah. Layar yang digunakan adalah layar warna berjenis STN (*Super-Twisted Nematic*) dengan 132 baris pixel dan 132 kolom RGB. Adapun fitur yang terdapat pada LCD PCF8833 sebagai berikut :

- Satu buah *chip* sebagai kontroler dan driver LCD
- Tegangan perintah berkisar antara 1,5V – 3,3V

- Luas layar sebesar 132 *pixel* x 132 *pixel*
- Dapat berkomunikasi dengan cara 8-bit parallel data atau serial data input
- Dilengkapi lampu latar belakang sebagai pencahayaan.

Pada alat ini peneliti menggunakan LCD sebagai output tampilan pada prototipe yang di letakkan pada Gate yang dibuat, sehingga pengguna dapat lebih mudah dalam mengoperasikan e-Ticket.

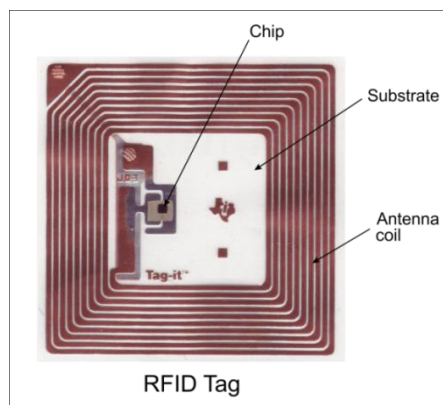
2.1.5. RFID

RFID merupakan salah satu teknologi dari sistem pengidentifikasian suatu objek secara otomatis (Auto ID) selain *barcode Optical character Recognition (OCR)*, *biometric*, dan *Smartcard* (Finkenzeller,2003). RFID (*Radio Frequency Identification*) merupakan sebuah alat yang bekerja dengan memanfaatkan gelombang frekuensi transmisi radio untuk menyampaikan data yang berisi nomor unik. Teknologi ini memiliki kelebihan karena cara penyampaian datanya yang tanpa menggunakan kontak tertentu dan mampu bekerja di setiap kondisi lingkungan (Eridani, 2012). Karena RFID menggunakan gelombang/frekuensi radio untuk sinyal pembawaan informasinya, sehingga RFID terdiri dari dua buah komponen yaitu komponen yang dapat menerima dan mengirim sinyal gelombang/frekuensi radio. Komponen tersebut terbagi menjadi dua buah bagian yaitu bagian penanda dan identitas (*tag*) dan bagian yang mengenali penanda tersebut (*reader*).

2.1.4.1 RFID Tag

RFID tag adalah sebuah benda kecil, misalnya berupa stiker adesif yang dapat ditempelkan pada suatu barang atau produk. RFID tag berisi antena yang memungkinkan peralatan itu menerima dan merespon terhadap suatu query yang dipancarkan oleh suatu RFID *transceiver*. Kebanyakan RFID tag mengandung setidaknya dua bagian: satu adalah sebuah sirkuit terpadu untuk menyimpan dan pengolahan informasi, modulasi dan

demodulasi sebuah frekuensi sinyal radio (RF), dan fungsi khusus lainnya, kedua adalah antena untuk menerima dan mengirimkan sinyal.



Gambar 2. 4 RFID Tag

(Sumber: <https://ikky17.files.wordpress.com/2010/12/rfid.jpg>)

Setiap bagian Tag terdiri dari :

1. *Chip* Ini adalah sebuah Mikroprosesor yang terletak dalam sebuah tag yang berfungsi sebagai penyimpan data.
2. *Antenna Coil* adalah sebuah komponen yang terbuat dari kawat aluminium yang berfungsi sebagai antena yang dapat beroperasi pada frekuensi 13,56 MHz. Jika sebuah tag masuk ke dalam jangkauan reader maka antena ini akan mengirimkan data yang ada pada tag kepada reader terdekat.
3. *Substrate* adalah bahan yang membungkus tag yang terbuat dari bahan kaca.

Terdapat 3 macam dari RFID tag yaitu tag pasif, tag semipasif, dan tag aktif:

1. Tag Pasif

Tag versi paling sederhana adalah tag pasif, yaitu tag yang tidak memiliki catu daya sendiri serta tidak menginisiasi komunikasi dengan reader. Sebagai gantinya, tag merespon frekuensi radio dan menurunkan dayanya dari gelombang-gelombang energi yang dipancarkan oleh reader. Sebuah tag pasif minimum mengandung sebuah identifier unik dari sebuah item yang dipasang tag tersebut. Tag pasif dapat beroperasi

pada frekuensi rendah (low frequency, LF) frekuensi tinggi (*high frequency*, HF), dan frekuensi ultra tinggi, (*ultrahigh frequency*, UHF). . Contoh RFID tag pasif seperti pada **Gambar 2.4**.



Gambar 2. 5 Tag pasif

(sumber: dokumentasi pribadi)

2. Tag Semi-pasif

Tag semi-pasif adalah versi *tag* yang memiliki catu daya sendiri (baterai) tetapi tidak dapat menginisiasi komunikasi dengan reader. Dalam hal ini baterai digunakan oleh tag sebagai catu daya untuk melakukan fungsi yang lain seperti pemantauan keadaan lingkungan dan mencatu bagian elektronik internal tag, serta untuk memfasilitasi penyimpanan informasi. *Tag* versi ini tidak secara aktif memancarkan sinyal ke reader. *Tag* semipasif dapat dihubungkan dengan sensor untuk menyimpan informasi untuk peralatan keamanan kontainer.

3. Tag Aktif

Tag aktif adalah tag yang selain memiliki antena dan chip juga memiliki catu daya dan pemancar serta mengirimkan sinyal kontinyu. *Tag* versi ini biasanya memiliki kemampuan baca tulis, dalam hal ini data tag dapat ditulis ulang atau dimodifikasi. *Tag* aktif dapat menginisiasi komunikasi dan dapat berkomunikasi pada jarak yang lebih

jauh, hingga 750 kaki, tergantung kepada daya baterainya. . Contoh RFID *tag* aktif seperti pada **Gambar 2.5**.



Gambar 2. 6 Tag aktif

Tabel 2. 1 Perbandingan Tag aktif dengan Tag pasif

	Tag Aktif	Tag Pasif
Sumber daya tegangan	Internal pada tag	Daya dikirim menggunakan RF dari reader
Baterai didalam label	Ya	Tidak
Kesediaan daya	Bersifat kontinyu	Hanya pada jangkuan medan <i>reader</i>
Kekuatan sinyal yang dibutuhkan dari reader ke label	Rendah	Tinggi
Ketersediaan kekuatan sinyal dari tag ke reader	Tinggi	Rendah
Jangkuan	100 meter atau lebih	3 meter atau kurang
Pembaca banyak label	Ribuan label dengan kecepatan hingga 120km/jam	Beberapa ratus label, dengan jarak sekitar 3 meter

Pada tabel 2.1 memperlihatkan perbandingan Tag aktif dan Tag pasif dilihat dari sumber daya tegangan yang digunakan, baterai yang ada didalam label, kesedian daya yang digunakan, kekuatan sinyal yang dibutuhkan dari *reader* lelabel kekuatan sinyal dari *tag* ke *reader*, jangkuan yang bekerja pada tag dan pembacaan pada tag.

Pada prototipe sistem pencarian buku perpustakaan menggunakan RFID *reader* berbasis Arduino Nano dan *Visual Basic 6.0*, RFID *tag* yang akan digunakan adalah RFID

tag pasif dalam bentuk lingkaran yang ditempelkan di setiap buku-buku dengan frekuensi kerja 13,56MHz (*high frequency tag*).

2.1.4.2 Jenis-jenis RFID berdasarkan Frekuensi

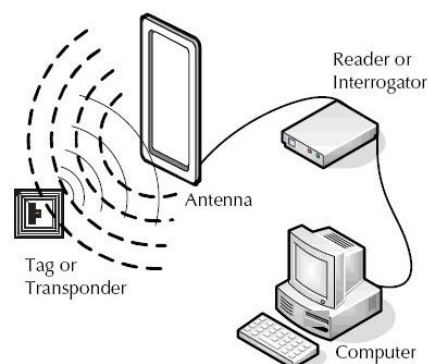
Ada empat macam RFID tag yang sering digunakan bila dikategorikan berdasarkan frekuensi radio, yaitu:

- a. *Low frequency tag* (antara 125 ke 134 khz)
- b. *High frequency tag* (13.56 MHz)
- c. UHF tag (868 sampai 956 MHz), UHF tag tidak bisa digunakan secara global, karena tidak ada peraturan global yang mengatur penggunaannya.
- d. *Microwave tag* (2.45 GHz)

2.1.4.3 Cara Kerja RFID

Terdapat tiga bagian utama pada pada sistem kerja RFID:

- a. Antena.
- b. *Transceiver* untuk men-decode data.
- c. *Transponder* yang telah diprogram dengan informasi tertentu berbentuk tag RFID.



Gambar 2.2 Cara kerja RFID

(sumber: <http://3.bp.blogspot.com/-wHCHKR-DI7o/U38Nc3ELjQI/AAAAAAAAAGY/4tKw2gPr6kU/s1600/5.jpg>)

Antena berfungsi melakukan komunikasi via sinyal dengan tag RFID, dan menyediakan energi bagi RFID tag (hanya pada kasus tag RFID pasif). Ketika tag RFID didekatkan dengan sinyal antena, perangkat ini akan mendeteksi sinyal aktivasi dari antena

dan mengaktifkan *chip* RFID. *Chip* ini akan mengirim informasi untuk diterima antenna untuk kemudian diolah didalam sistem.

2.1.4.4 Modul Pembaca RFID

Modul Pembaca RFID atau biasa disebut dengan RFID *Reader* yaitu sebuah modul yang berfungsi untuk membaca ID atau data yang terdapat pada RFID Tag. Pembaca RFID (RFID *reader*) merupakan penghubung antar *software* aplikasi dengan antenna yang akan meradiasikan gelombang radio ke *tag (transponder)* RFID (Fangan, 2004) Identifikasi objek atau data pada teknologi RFID dilakukan dengan mencocokkan data yang tersimpan dalam memori tag *transponder* dengan data yang dikirimkan oleh *reader*.

Modul pembaca RFID yang akan digunakan dalam alat ini yaitu RC522 buatan NXP *Semiconductors* dengan spesifikasi produk sebagai berikut :

- a. *Chipset: MFRC522 Contactless Reader/Writer IC*
- b. Frekuensi: 13,56 MHz
- c. Jarak pembacaan kartu: < 50 mm
- d. Protokol akses: SPI (*Serial Peripheral Interface*) @ 10 Mbps
- e. Kecepatan transmisi RF: 424 kbps (dua arah / *bi-directional*) / 848 kbps (*unidirectional*)
- f. Mendukung kartu [MIFARE](#) jenis Classic S50 / S70, *UltraLight*, dan *DESFire*
- g. *Framing & Error Detection (parity+CRC)* dengan 64 byte internal I/O *buffer*
- h. Catu Daya: 3,3 Volt
- i. Konsumsi Arus: 13-26 mA pada saat operasi baca/tulis, < 80µA saat modus siaga
- j. Suhu operasional: -20°C s.d. +80°C

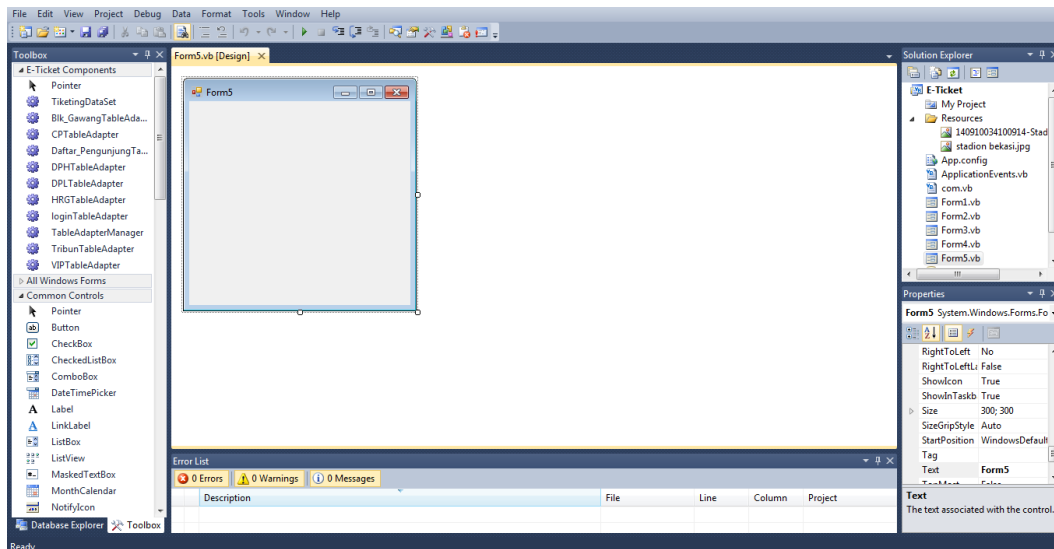


Gambar 2. 7 modul pembaca RFID RC522

(sumber: dokumentasi pribadi)

2.1.6. Visual Basic

Microsoft Visual Basic.NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem. NET *Framework*, dengan menggunakan bahasa *basic*. Dengan menggunakan alat ini, para *programmer* dapat membangun aplikasi Windows Forms, aplikasi web berbasis ASP.NET, dan juga aplikasi *command-Line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*, *Visual C#*, atau *Visual J#*) atau juga dapat diperoleh secara terpadu dalam *Microsoft Visual Studio .NET*. bahasa Visual Basic .NET sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari *Microsoft Visual Basic* versi sebelumnya yang diimplementasikan di atas .NET Framework.



Gambar 2. 8 Tampilan form Visual Basic 2010

2.1.7. Microsoft Access

Access adalah software *Database Management System* (DBMS). Seperti produk DBMS lainnya, *Access* mempunyai kemampuan untuk menyimpan dan menampilkan kembali informasi (yang biasanya sering disebut sebagai data), menampilkan informasi berdasarkan permintaan tertentu dan juga melakukan otomatisasi pada *repetitive task*. Dengan *Access* anda dapat dengan mudah membuat suatu *input form* maupun menampilkan *report* dengan tampilan yang menarik tetapi dengan cara yang mudah. (Susanto dan Hartono, 2000::xi)

Access merupakan *software database* yang tidak serumit *software database* lain, seperti Oracle, SQL Server, Informix tetapi cenderung lebih sederhana dan lebih praktis dalam penggunaannya. Oleh karena *access* merupakan produk dari *Microsoft*, *access* dapat bekerja dengan baik pada *platform windows*, baik itu pada windows 95, Windows 98, Windows ME, Windows NT bahkan sampai dengan Windows XP.

2.1.8. Database Tiketing

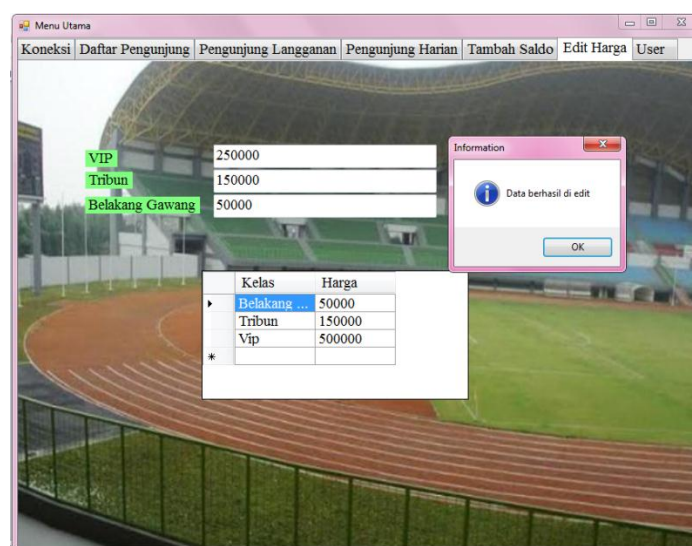
Database adalah koleksi dari data-data yang terkait secara logis dan deskripsi dari data-data tersebut, yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi (Robby, 2004). kumpulan/koleksi data-data yang saling berhubungan dengan cara-

cara tertentu. Selain itu, kumpulan/koleksi data tersebut juga harus diorganisir dan diklasifikasikan dalam format yang terstruktur. Dengan karakteristik ini, *database* dapat digunakan secara efektif dalam pengambilan maupun penyimpanan data.

Pada penelitian ini database digunakan sebagai media penyimpan data pengunjung stadion baik pengunjung langganan maupun pengunjung harian sehingga data yang diinput melalui aplikasi tiketing menjadi tertata. Data yang telah disimpan pada database dapat digunakan pihak manajemen untuk menganalisa profil sebaran pengunjung.

2.1.9. Tiket Stadion Sepak Bola Berbasis Arduino UNO

Tiket stadion sepak bola yaitu sebuah sistem yang dibangun untuk mencegah penonton yang hendak menerobos masuk stadion dengan tiket yang tidak sesuai dengan cara mendesain gate masuk stadion dibuat secara otomatis dengan mikrokontroler berupa arduino uno yang diprogram dengan *soft ware* Arduino dan pada sistem ini juga dilengkapi dengan penyimpanan data penonton stadion dalam sebuah database dengan bantuan *soft ware* Microsoft Access 2010 dan sebuah server dalam bentuk aplikasi yang dibuat peneliti dengan bantuan *soft ware* Visual Basic 2010 agar dapat menghubungkan antara database dengan gate masuk. Tiket yang digunakan berupa RFID Card yang memiliki ID berbeda di setiap kartunya.



Gambar 2. 9 Tampilan aplikasi server

2.2. Kerangka Berfikir

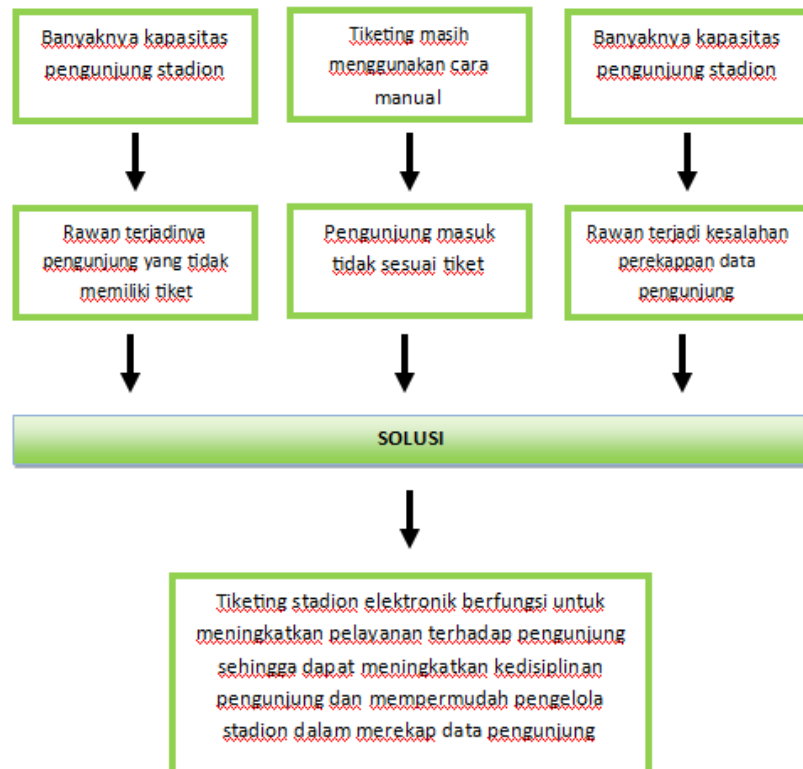
Dengan landasan teori yang telah ditulis pada halaman sebelumnya, pada sistem tiketing elektronik menggunakan RFID berbasis Arduino. Dengan server tiketing yang peneliti buat dengan bantuan perangkat lunak visual basic 2010 dan microsoft access 2010 sebagai database tiketing stadion.

mula-mula server tiketing mendata id kartu RFID dengan menempelkan kartu pada reader RFID server dan memasukan data yang diperlukan kedalam kolom-kolom yang terdapat pada aplikasi server Tiketing. Bila kartu didata sebagai kartu langganan maka ID kartu akan disimpan ke dalam database dengan jumlah saldo yang ditukarkan dan kartu dapat digunakan maksimal 4 kali pembelian tiket selama saldo mencukupi dalam satu kali pertandingan, jika kartu di data sebagai kartu langganan maka kartu hanya memiliki kelas stadion yang bisa diproses satu kali pada gate stadion

Pada Gate terdapat 2 buah input yang pertama input dari RFID reader berupa ID yang diterjemahkan melalui Arduino dan input dari softwae e-Ticket berupa data identitas kartu yang disimpan pada Database Acces. Pengiriman data dari arduino ke PC begitu juga sebaliknya melalui bantuan kabel USB sehingga bila data yang diterima oleh arduino melalui RFID reader sesuai dengan Database maka Arduino dapat memutar servo agar gate dapat terbuka dan LCD dapat menampilkan data yang diterima oleh PC melalui Arduino. Dan ketika gate diputar maka akan menekan limitswitch yang akan memberi inputan ke arduino agar mengembalikan servo ke posisi awal.

2.2.1. Blok Diagram

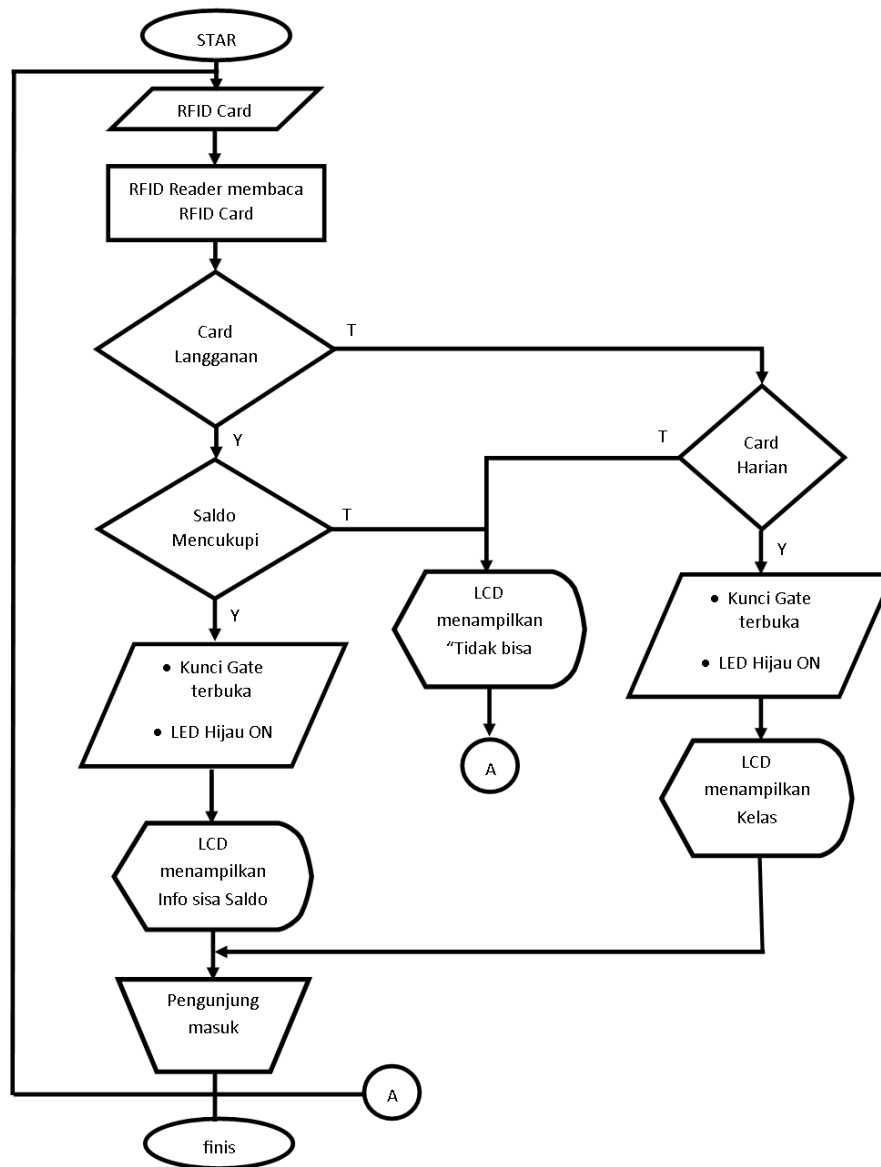
Blok diagram sistem merupakan langkah awal dalam pembuatan alat Tiketing Stadion elektronik. Dalam merancang sebuah sistem hal terpenting adalah menjadikan sistem menjadi tiga bagian besar, pertama *input* (masukan), proses dan *output* (keluaran).



Gambar 2. 10 Blok diagram sistem tiketing elektronik

Pada e-Ticket terdapat 2 buah input yang pertama input dari RFID reader berupa ID yang diterjemahkan melalui Arduino dan input dari software e-Ticket berupa data identitas kartu yang disimpan pada Database Acces. Pengiriman data dari arduino ke PC begitu juga sebaliknya melalui bantuan kabel USB sehingga bila data yang diterima oleh arduino melalui RFID reader sesuai dengan Database maka Arduino dapat memutar servo agar gate dapat terbuka dan LCD dapat menampilkan data yang diterima oleh PC melalui Arduino.

2.2.2. *Flowchart Sistem Kerja Pembelian Tiket Stadion Elektronik*



Gambar 2. 11 Diagram alir sistem masuk Stadion melalui Gate elektronik

BAB III

METODE PENELITIAN

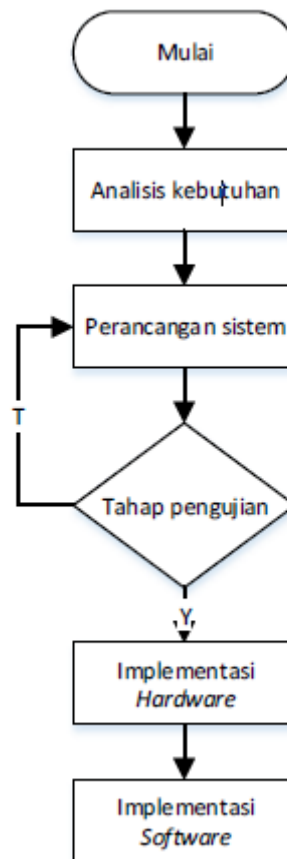
3.1. Tempat dan Waktu Penelitian

Penelitian dilakukan di Lantai 4 Ruang 401, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta. Dalam rentang waktu pada bulan Juli 2015 – Januari 2016.

3.2. Metodologi Penelitian

Metodologi penelitian adalah langkah-langkah yang dilakukan dalam penelitian ilmiah yang bertujuan untuk mendapatkan hasil sehingga tujuan dari penelitian tersebut dapat terpenuhi. Metodologi penelitian yang digunakan dalam menyelesaikan penelitian ini peneliti menggunakan metodologi penelitian dan pengembangan (*Research and Development*) yaitu metode penelitian yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut (Sugiyono, 2009). berikut adalah langkah-langkahnya:

1. Pengumpulan data
2. Desain produk
3. Uji coba produk
4. Revisi produk



Gambar 3. 1 Tahap-tahap metodologi penelitian perancangan sistem tiketing elektronik

3.2.1 Pengumpulan Data

Pengumpulan data adalah tahap dimana peneliti mengumpulkan informasi untuk menentukan kebutuhan dari sistem agar pembelian tiket stadion elektronik dapat digunakan sesuai dengan tujuan penelitian. Untuk memenuhi tujuan tersebut, peneliti menggunakan *software* VB.NET 2010 dalam membentuk aplikasi server e-ticket, serta menggunakan sistem input output yang diproses oleh Arduino dan keduanya dihubungkan melalui modem menggunakan kabel USB

3.2.2 Desain Produk

Dalam tahap perencanaan berisi perencanaan kerangka berpikir peneliti dalam pembuatan sistem pembelian tiket stadion elektronik menggunakan RFID berbasis Arduino dengan bahasa pemrograman Arduino yang dapat dilihat pada blog diagram dan *flowchart*

yang telah dijelaskan di bab 2 pada **Gambar 2.10** untuk blok diagram dan untuk **Gambar 2.11** untuk *flowchart*.

3.2.3 Uji Coba Produk

Pada tahap pengujian peneliti melakukan uji coba, uji coba pertama dengan mengintegrasikan komponen-komponen dengan mikrokontroler Arduino. Pada uji tahap kedua dilakukan untuk menguji pembacaan data e-ticket (*RFID Card*) dan pengiriman data yang telah diproses oleh mikrokontroler ke PC untuk mengetahui data yang diterima sesuai dengan data yang dikirimkan e-ticket (*RFID Reader*).

Pada uji coba tahap akhir adalah pengujian kelayakan alat e-ticket yang digunakan untuk mencatat dan membayar pembelian tiket stadion (kartu langganan), apakah alat e-ticket tersebut benar dapat mencatat dan mengurangi saldo yang ada pada data e-ticket didalam server tiketing stadion sesuai dengan harga tiket yang ditetapkan.

Sedangkan uji coba e-ticket Kartu harian dilakukan dengan mengguakannya untuk mencatat kelas pada database dengan menggunakan ID pada e-tiket, apakah alat tersebut dapat mencatat kelas pada database sesuai dengan tujuan peneliti. Untuk mendapatkan data yang akurat dalam proses pengujian kelayakan alat e-ticket ini, peneliti menggunakan metode observasi, dengan mensimulasikan penjualan tiket sebanyak lima kali dan menganalisis data transaksi penjualan.

3.2.4 Revisi Produk

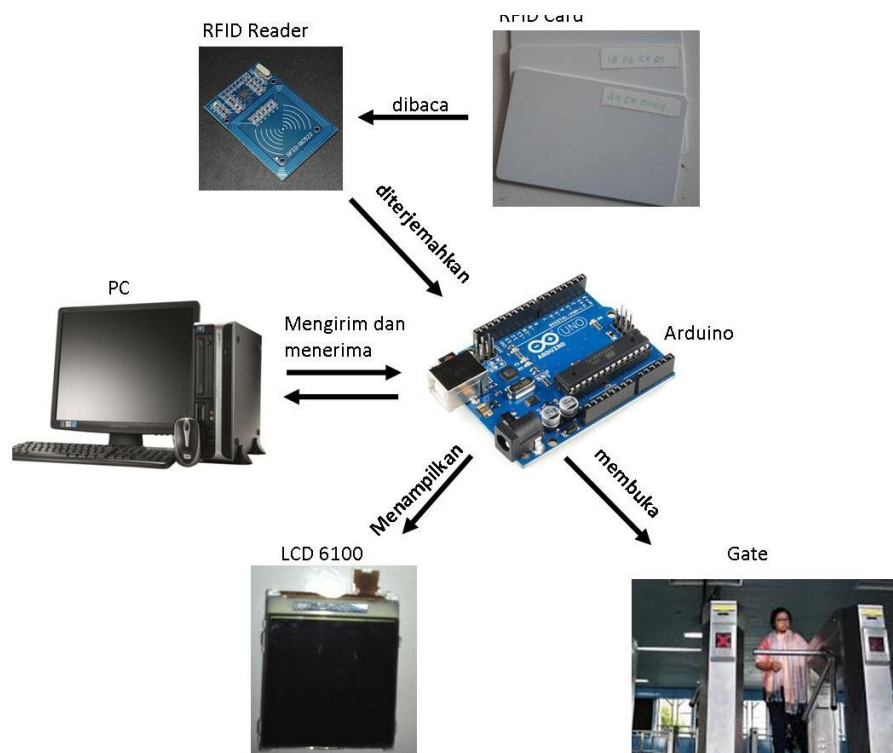
Setelah tahap uji coba maka selanjutnya adalah merevisi sistem tiket stadion dengan kriteria sistem tiket yang dirancang sebelumnya dengan mengacu dari hasil uji coba perangkat keras E-tiket dan perangkat lunak E-tiket yang dibuat peneliti dengan bantuan aplikasi visual basic 2010 sebagai media penghubung antara arduino dengan database microsoft access 2010.

3.3. Rancangan Penelitian

Rancangan penelitian merupakan suatu rencana yang komprehensif dan memiliki tujuan yang terarah dalam melakukan penelitian untuk menghasilkan karya sesuai dengan yang diinginkan. Hal yang dilakukan dalam perancangan sistem tiket elektronik menggunakan RFID ini terdiri dari beberapa tahapan.

3.3.1 Menentukan Diagram Blok Sistem

Dalam menentukan diagram blok sistem peneliti merancang diagram blok program dari sistem untuk menjadi bahan acuan dalam proses penerjemahan dari diagram blok menjadi sebuah alat e-ticket serta sebagai bahan analisa kebutuhan perangkat pendukung baik dari dalam segi *Hardware* maupun *Software* untuk bekerjanya sistem pembelian tiket elektronik. Diagram blok dapat dilihat pada gambar dibawah.

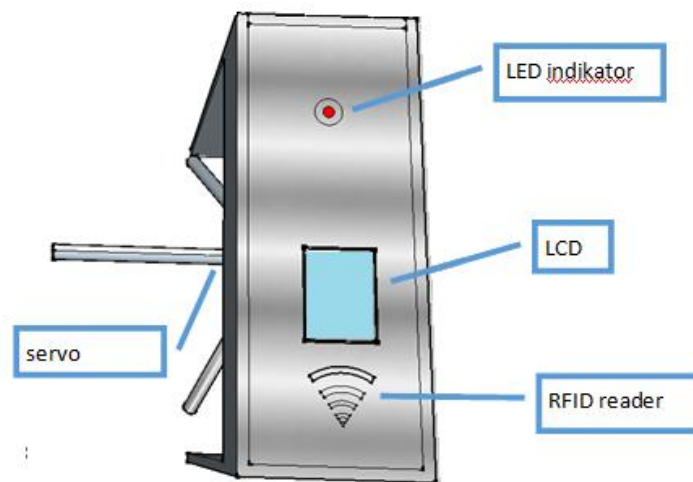


Gambar 3. 2 Blok diagram sistem Tiketing Stadion

Berdasarkan diagram blok pada **Gambar 3.2**, RFID reader akan membaca data ID e-ticket (RFID Card) yang kemudian akan dikirimkan ke Arduino, data dikirimkan melalui kabel USB yang diterima oleh Server tiketing. ID yang diterima server Tiketing kemudian diproses, dalam pemrosesan dilakukan pencatatan pembelian, pengurangan saldo sebesar harga tiket stadion yang harus dibayarkan serta mengirimkan notifikasi kembali ke Arduino (*Gate*).

3.3.2 Perencanaan Desain Alat

Desain alat dibuat dalam bentuk sederhana agar mudah digunakan dan dibawa dengan dilengkapi LCD warna seluas 1,8” sebagai *output* tampilan, serta dilengkapi dengan *port micro* USB untuk memprogram alat dan menghubungkan Arduino (*Gate*) dengan program e-ticket tersebut. Adapun tampilan fisik alat dapat dilihat pada **Gambar 3.1**.



Gambar 3. 3 Desain Alat Gate Stadion

Keterangan gambar :

- Dimensi alat 200 mm × 80 mm × 200 mm
- LCD warna 1,8” sebagai *output* tampilan alat
- Lampu indikator sebagai tanda alat gate terkunci dan tidak terkunci.

- *Port* mikro USB dapat digunakan sebagai pemrogram alat.

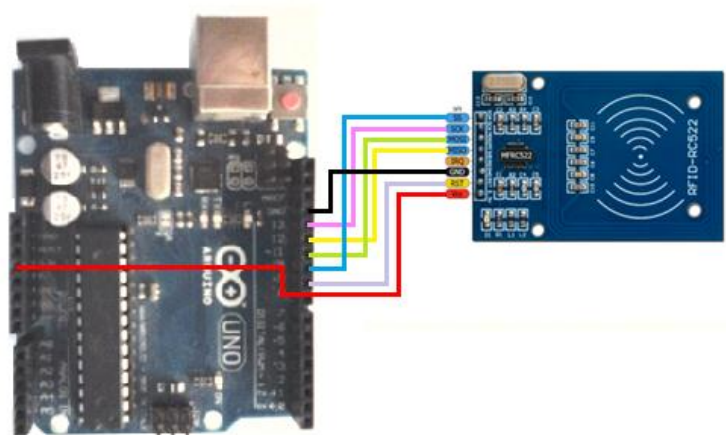
3.3.3 Perancangan Perangkat Keras

Perancangan perangkat keras menentukan keberhasilan suatu sistem. Dalam perancangan perangkat keras e-ticket ini harus dirancang kuat dan stabil agar proses pembelian tiket elektronik dapat berjalan dengan baik.

3.3.4 Perancangan Integrasi komponen

Sebelum melakukan perancangan alat secara keseluruhan, ada baiknya menyiapkan perancangan Integrasi komponen yang akan digunakan dalam alat, guna menentukan *pin input* maupun *output* yang akan digunakan pada Arduino. Apabila perancangan pengintegrasian sudah selesai dilakukan barulah peneliti dapat membuat skematik dan PCB e-ticket. Adapun perancangan tersebut sebagai berikut :

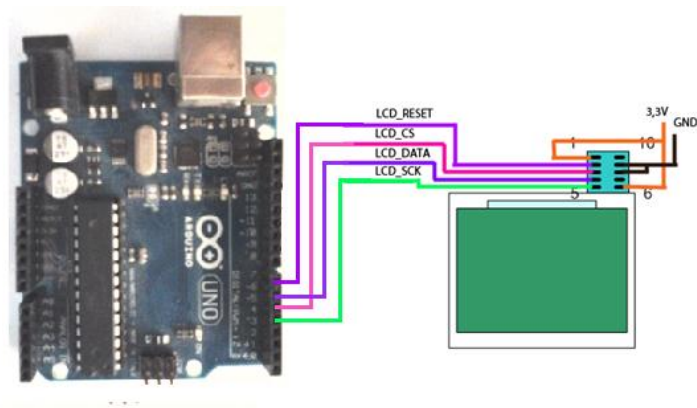
1. Perancangan Pengintegrasian RFID *reader* dengan ATmega328



Gambar 3. 4 Integrasi modul RFID MFRC522 dengan Arduino UNO

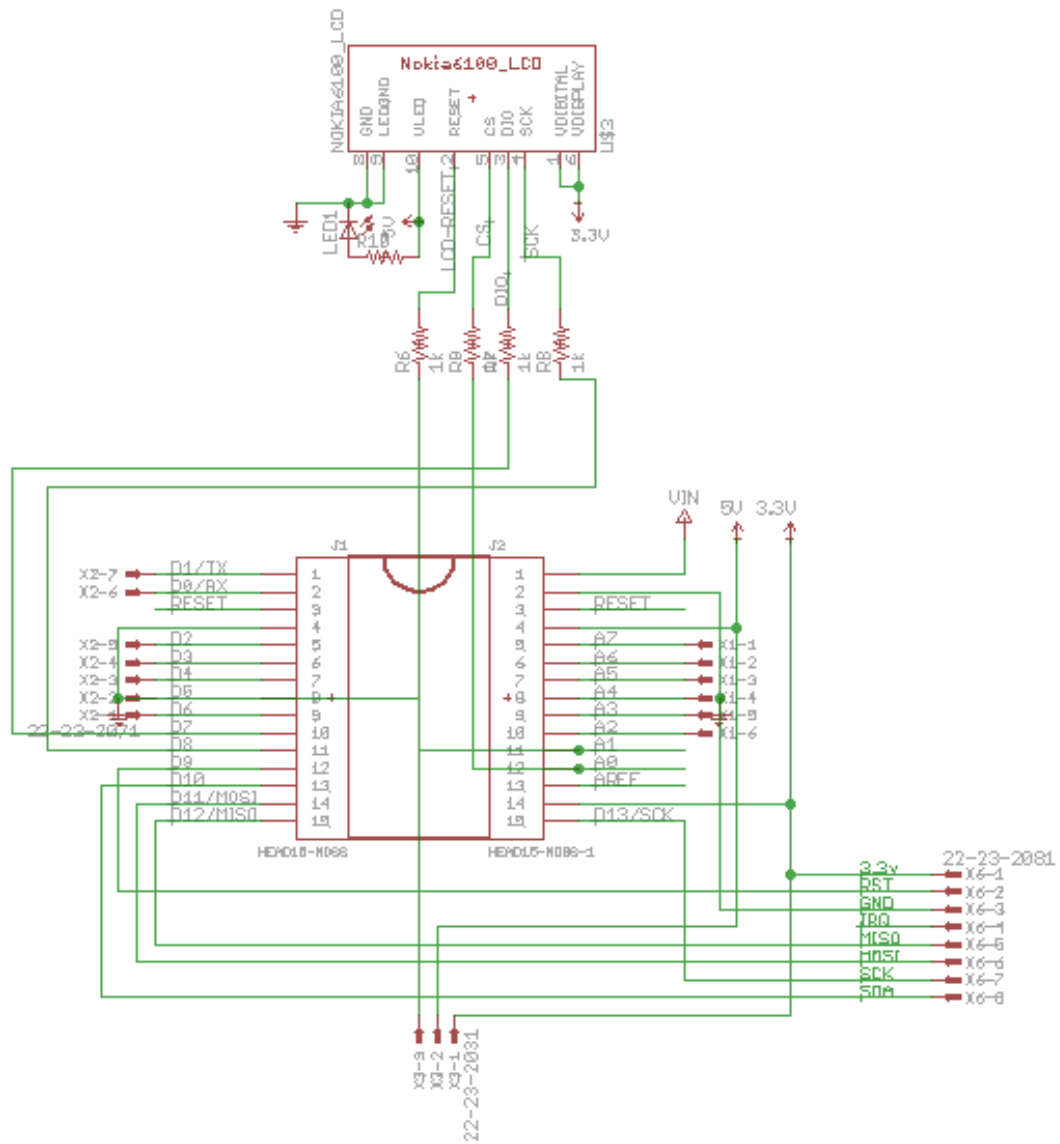
Modul pembaca RFID disambungkan dengan Arduino melalui komunikasi SPI (*Serial Pheriperal Interface*). Komunikasi SPI membutuhkan tiga jalur yaitu MOSI, MISO, dan SCK. Melalui komunikasi ini data dapat saling dikirimkan baik antara mikrokontroler maupun antara peripheral lain di luar mikrokontroler.

2. Perancangan Pengintegrasian LCD dengan Arduino



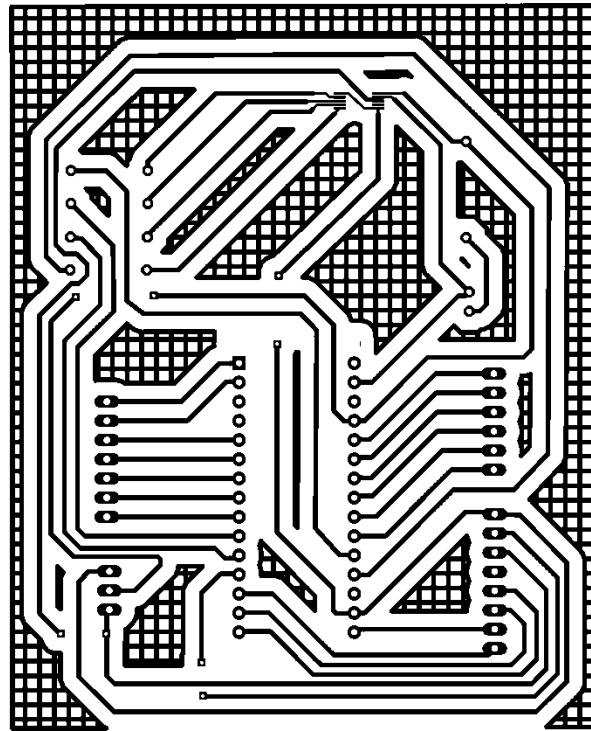
Gambar 3.5 Integrasi LCD dengan Arduino UNO

3. Perancangan Skematik Rangkaian



Gambar 3. 6 Skematik Keseluruhan alat e-ticket

Perancangan Desain *Layout* PCB *e-ticket*



Gambar 3. 7 Layout PCB *e-ticket*

3.3.5 Perencanaan Perangkat Lunak

Perancangan perangkat lunak berupa perancangan program yang dibuat untuk mendukung sistem kerja dari sistem pembelian tiket elektronik yang dapat dihubungkan dengan *software* lain untuk mendapatkan sistem yang baik. Adapun perancangan perangkat lunak yang dibuat sebagai berikut.

3.3.6 Perancangan Program Arduino untuk Arduino Board

Perancangan program Arduino menggunakan *software* IDE Arduino 1.6.5. Perancangan program dibuat berdasarkan prinsip kerja dari sistem pembelian tiket menggunakan RFID berbasis Arduino dengan bahasa pemrograman Arduino. Berikut penggunaan pin *input* pada Arduino dengan perangkat *input*, dapat dilihat pada **Tabel 3.1**.

Tabel 3. 1 Penggunaan Pin Input pada Arduino dengan Perangkat Input

No	Perangkat <i>Input</i>	Pin Perangkat <i>Input</i>	Pin Arduino
1	Limit Swictch	1	D3
2	RFID <i>reader</i>	1	D10
		2	D13
		3	D11
		4	D12
		7	D9

Untuk penggunaan pin *output* pada Arduinodengan perangkat *output*, dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Penggunaan Pin Output pada Arduino dengan Perangkat Output

No	Perangkat <i>Output</i>	Pin Perangkat <i>Output</i>	Pin Arduino
1	LCD PCF8833	2	D7
		3	D5
		4	D4
		5	D6
2	Servo		D3
3	LED Hijau		A4
4	LED Merah		A5

Penggunaan pin *Input* dan *Output* pada Arduino ditetapkan berdasarkan fungsi khusus dari masing-masing pin.

3.3.7 Perancangan Program VB.Net untuk membangun server tiketing

Peneliti menggunakan perangkat lunak Visual Basic 2010 (VB.Net 2010) untuk pembuatan server pendukung sistem pembelian tiket elektronik sekaligus untuk melakukan pengecekan data transaksi pembelian tiket, pendaftaran *e-ticket* dan pengisian saldo dalam kartu *e-ticket* (langganan) dan *e-ticket* (harian). Selain menggunakan VB.Net 2010 peneliti

juga menggunakan *Software* pendukung seperti Microsoft Access sebagai *database* untuk mencatat seluruh data sistem pembelian tiket elektronik dan registrasi *e-ticket*.

3.4. Instrumen Penelitian

Untuk memberikan hasil yang akurat dan presisi saat pengambilan data dalam penelitian ini, digunakan beberapa instrumen penelitian yaitu sebagai berikut :

1. Notebook yang digunakan dalam penelitian dengan spesifikasi sebagai berikut:
 - a. AMD E-450 APU with Radeon™ HD Graphic 1.65 GHz
 - b. Memory 2.00 GB (1,61 GB usable)
 - c. *Solite State Drive (SSD)* 120 GB
 - d. LCD 14"
 - e. Sistem Operasi Windows 7 Pro 32 bit
2. *Software* pendukung terdiri dari:
 - a. Arduino IDE 1.6.5, untuk memprogram ATmega328.
 - b. *Eagle* 6.4.0, untuk membuat skematik dan layout rangkaian.
 - c. *Google Sketchup* 2014, untuk membuat desain perancangan maket.
 - d. *Photoshop* CS5, untuk menggambar integrasi modul dengan ATmega328.
 - e. *Microsoft Office* 2010, yaitu berfungsi untuk penulisan laporan.
3. *Hardware* pendukung yang digunakan terdiri dari:
 - a. Solder listrik.
 - b. Antraktor
 - c. Mini *electric drill* (Bor tangan kecil).
 - d. Multimeter digital.
 - e. Gunting.
 - f. *Cutter*.
 - g. *Lotfat*
 - h. Tang potong

4. Alat ukur AVOMeter Exel digunakan untuk mengukur besar tegangan, mengukur komponen elektronika yang digunakan pada rangkaian, menganalisa kesalahan jalur-jalur tembaga pada PCB dalam rangkaian e-ticket.

3.5. Prosedur Penelitian

Prosedur penelitian sistem tiketing elektronik menggunakan RFID dilakukan melalui beberapa tahap :

1. Pembuatan Rangkaian *E-Ticket*

Rangkaian *e-ticket* adalah rangkaian *Shield* Arduino yang telah diintegrasikan dengan komponen pendukung seperti RFID *reader*, satu buah tombol *input*, satu buah LCD PCF 8833, dan *downloader* dengan *port micro* USB sebagai jalur pemrograman yang seluruhnya dibangun dalam satu potong papan PCB *single layer* dengan dimensi panjang 100 mm dan lebar 70 mm. Kelebihan dari rangkaian *e-ticket* ini yaitu dapat diprogram langsung menggunakan *Software IDE* Arduino melalui *port micro* USB yang telah disediakan.

2. Pembuatan program Arduino Board dengan IDE Arduino

Program Arduino dibuat untuk memproses *input* data dari RFID *reader*, menuliskan kode pelanggaran dan mengirimkan data tersebut ke server tiketing melalui kabel USB. Program dibuat dengan bahasa Arduino dan menggunakan *software IDE* Arduino. Selain itu mikrokontroler juga digunakan untuk memproses data status dari server tiketing yang kemudian ditampilkan melalui LCD.

3. Koneksi Aplikasi Server Tiketing dengan Gate

Agar terjadi komunikasi antara server tiketing dengan alat terlebih dahulu komputer dalam keadaan aktif dan aplikasi server tiketing dalam keadaan *run* atau dalam kondisi sedang berjalan dan terhubung dengan port USB. Koneksi antara server tiketing dengan alat gate melalui kabel USB.

4. Uji Coba Alat Dengan Aplikasi Server Tiketing.

pengujian alat dilakukan setelah perangkat lunak dan perangkat keras diuji.

3.6. Pengujian Alat

Pengujian yang dilakukan peneliti untuk mendapatkan data yang diperlukan pada keseluruhan sistem pembayaran tiket elektronik, kriteria pengujian dilakukan peneliti untuk menyatakan bahwa sistem yang telah dibuat dinyatakan berhasil atau gagal, berikut kriteria pengujian pada penelitian *Sistem pembelian tiket elektronik menggunakan RFID*.

3.6.1.1 Kriteria Pengujian *Hardware* dan *Software*

3.6.1.1 Pengujian Pembacaan Kartu e-Ticket

Dalam pengujian pembacaan e-ticket dengan alat e-ticket (gate) di fokuskan dalam pembacaan ID kartu RFID. Tujuan dari pengamatan ini untuk menghindari perbedaan data pengunjung dengan ID kartu yang di baca.

Tabel 3. 3 Pengujian Kesesuaian ID kartu dengan data kartu

No	Kartu E-ticket	Kriteria Pengujian	Hasil
1	Kartu 1	ID kartu sesuai dengan yang dibaca	
2	Kartu 2	ID kartu sesuai dengan yang dibaca	
3	Kartu 3	ID kartu sesuai dengan yang dibaca	
4	Kartu 4	ID kartu sesuai dengan yang dibaca	
5	Kartu 5	ID kartu sesuai dengan yang dibaca	

3.6.1.2 Pengujian penyimpanan data E-tiket Langganan

Tabel 3. 4 Pengujian penyimpanan data E-tiket Langganan

No	E-ticket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Data tersimpan di Database		
2	Kartu 2	Data tersimpan di Database		
3	Kartu 3	Data tersimpan di Database		
4	Kartu 4	Data tersimpan di Database		

5	Kartu 5	Data tersimpan di Database		
---	---------	----------------------------	--	--

3.6.1.3 Pengujian pengeditan data E-tiket Langgan

Tabel 3. 5 Pengujian pengeditan data E-tiket Langgan

No	E-tiket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Dapat mengedit data sesuai ID di dalam database		
2	Kartu 2	Dapat mengedit data sesuai ID di dalam database		
3	Kartu 3	Dapat mengedit data sesuai ID di dalam database		
4	Kartu 4	Dapat mengedit data sesuai ID di dalam database		
5	Kartu 5	Dapat mengedit data sesuai ID di dalam database		

3.6.1.4 Pengujian penambahan saldo E-tiket Langgan

Tabel 3. 6 Pengujian penambahan saldo E-tiket Langgan

No	E-tiket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Dapat menambahkan saldo sesuai ID E-tiket pada database		
2	Kartu 2	Dapat menambahkan saldo sesuai ID E-tiket pada database		
3	Kartu 3	Dapat menambahkan saldo sesuai ID E-tiket pada database		
4	Kartu 4	Dapat menambahkan saldo sesuai ID E-tiket pada database		
5	Kartu 5	Dapat menambahkan saldo sesuai ID E-tiket pada database		

3.6.1.5 Pengujian penyimpanan data E-tiket Harian

Tabel 3. 7 Pengujian penyimpanan data E-tiket Harian

No	E-tiket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Dapat menyimpan data kelas sesuai ID E-tiket		
2	Kartu 2	Dapat menyimpan data kelas sesuai ID E-tiket		
3	Kartu 3	Dapat menyimpan data kelas sesuai ID E-tiket		
4	Kartu 4	Dapat menyimpan data kelas sesuai ID E-tiket		
5	Kartu 5	Dapat menyimpan data kelas sesuai ID E-tiket		

3.6.1.6 Pengujian pengeditan harga tiket

Tabel 3. 8 Pengujian pengeditan harga tiket

No	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Dapat mengedit harga tiket sesuai inputan server		
2	Dapat mengedit harga tiket sesuai inputan server		
3	Dapat mengedit harga tiket sesuai inputan server		
4	Dapat mengedit harga tiket sesuai inputan server		
5	Dapat mengedit harga tiket sesuai inputan server		

3.6.1.7 Pengujian gate saldo E-tiket langganan mencukupi

Tabel 3. 9 Pengujian gate saldo E-tiket langganan mencukupi

No	E-tiket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Menampilkan data saldo pada layar LCD dan LED hijau On		
2	Kartu 2	Menampilkan data saldo pada layar LCD dan LED hijau On		

3	Kartu 3	Menampilkan data saldo pada layar LCD dan LED hijau On		
4	Kartu 4	Menampilkan data saldo pada layar LCD dan LED hijau On		
5	Kartu 5	Menampilkan data saldo pada layar LCD dan LED hijau On		

3.6.1.8 Pengujian gate saldo E-tiket langganan tidak mencukupi

Tabel 3. 10 Pengujian gate saldo E-tiket langganan tidak mencukupi

No	E-tiket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Menampilkan data saldo pada layar LCD dan LED merah tetap On		
2	Kartu 2	Menampilkan data saldo pada layar LCD dan LED merah tetap On		
3	Kartu 3	Menampilkan data saldo pada layar LCD dan LED merah tetap On		
4	Kartu 4	Menampilkan data saldo pada layar LCD dan LED merah tetap On		
5	Kartu 5	Menampilkan data saldo pada layar LCD dan LED merah tetap On		

3.6.1.9 Pengujian gate E-tiket harian sesuai dengan kelas

Tabel 3. 11 Pengujian gate E-tiket harian sesuai dengan kelas

No	E-tiket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Menampilkan kelas pada layar LCD dan LED hijau On		
2	Kartu 2	Menampilkan kelas pada layar LCD dan LED hijau On		
3	Kartu 3	Menampilkan kelas pada layar LCD dan LED hijau On		
4	Kartu 4	Menampilkan kelas pada layar LCD dan LED hijau On		
5	Kartu 5	Menampilkan kelas pada layar LCD dan LED hijau On		

3.6.1.10 Pengujian gate E-tiket harian sesuai dengan kelas

Tabel 3. 12 Pengujian gate E-tiket harian sesuai dengan kelas

No	E-tiket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Menampilkan kelas pada layar LCD dan LED merah tetap On		
2	Kartu 2	Menampilkan kelas pada layar LCD dan LED merah tetap On		
3	Kartu 3	Menampilkan kelas pada layar LCD dan LED merah tetap On		
4	Kartu 4	Menampilkan kelas pada layar LCD dan LED merah tetap On		
5	Kartu 5	Menampilkan kelas pada layar LCD dan LED merah tetap On		

3.7. Spesifikasi Sistem.

1. E-tiket menggunakan RFID tag *high frequency tag* (13,56 Mhz).
2. Pembaca E-tiket menggunakan modul RFID *Reader* dengan tipe RC522.
3. Tampilan pada Gate menggunakan LCD Philips PCF8833.
4. Gate hanya melewatkan satu orang untuk satu kali tapping.
5. Untuk E-tiket langganan maksimal tapping 4 dalam satu pertandingan.
6. E-tiket harian hanya bisa sekali tapping.
7. Mikrokontroler yang digunakan Arduino Uno.
8. Aplikasi Server dibangun dengan *Software Visual Basic 2010* dan database microsoft Access 2010.
9. Data penonton stadion dan penjualan tiket tersimpan dalam database.

BAB IV

HASIL PENELITIAN

4.1. Hasil Penelitian

Hasil pengujian yang telah dilakukan terhadap indikator-indikator penelitian "Pemanfaatan RFID Sebagai Tiket Stadion Sepak Bola Berbasis Arduino Uno" dapat dikategorikan sebagai berikut :

1. Hasil pengujian perangkat keras (*Hardware*).
2. Hasil pengujian program tiketing otomatis dan program Arduino UNO (*Software*).
3. Hasil pengujian alat tiketing otomatis.

4.1.1. Hasil Pengujian dan Pembuatan Hardware

4.1.1.1. Hasil dan Pembuatan Alat

Alat Pintu e-ticket menggunakan RFID berbasis Arduino dengan bahasa pemrograman Arduino dan server tiketing sebagai sebagai pengolah data tiket stadion terdiri dari *chasing* alat, dan papan rangkaian alat E-ticket. *Chasing* terbuat dari bahan akrilik dengan ketebalan 5 mm. Pembuatan *chasing* alat ini dibuat dengan berpacu pada rancangan desain alat pada bab sebelumnya dengan dimensi panjang 200 mm, lebar 80 mm, dan tebal 200 mm tampak seperti gambar 4.1.



Gambar 4. 1 casing Gate stadion

Untuk papan rangkaian yang telah dibuat dengan *PCB single layer* dengan ukuran panjang 100 mm dan lebar 70 mm tampak pada gambar 4.2.



Gambar 4. 2 Papan Rangkaian E-Ticket

Setelah *chasing* dan papan rangkaian dibuat, peneliti menggabungkan keduanya. Adapun tampak alat e-ticket yang telah selesai digabungkan dapat dilihat pada gambar 4.3.



Gambar 4. 3 Hasil akhir pembuatan E-Tiket

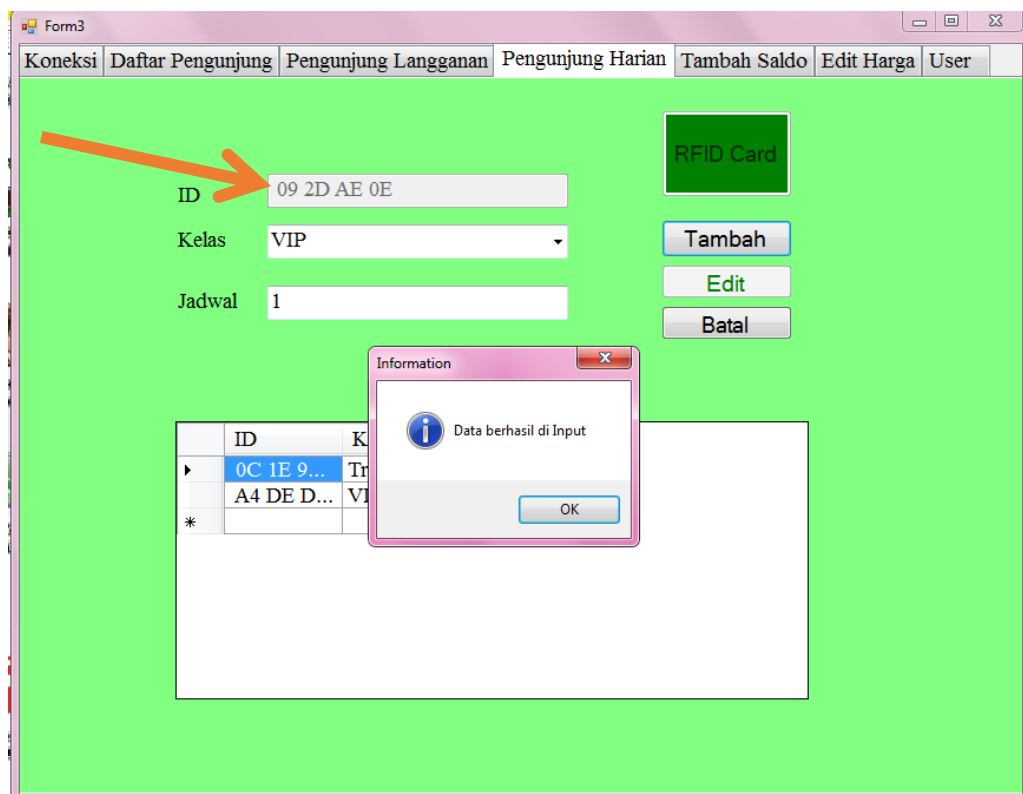
4.1.2. Hasil Pengujian Software Tiketing

4.1.2.1. Hasil Pengujian Pembacaan ID E-Ticket

Dalam pengujian pembacaan e-ticket dengan alat tiketing difokuskan dalam pembacaan ID e-ticket. Tujuan dari pengamatan ini untuk menghindari perbedaan data, terutama pada ID e-ticket yang di baca dengan ID e-ticket yang tertulis pada kartu. Tata cara pengujian kesesuaian ID e-ticket yaitu dengan mencocokkan id yang tertulis pada e-ticket dengan ID e-ticket yang terbaca oleh gate tiketing. Hasil pengujian dapat dilihat pada Tabel 4.1 dan Gambar 4.4.

Tabel 4. 1 Hasil pengujian Kesesuaian ID kartu dengan data kartu

No	Kartu E-Tiket	Kriteria Pengujian	Hasil
1	Kartu 1	ID kartu sesuai dengan yang dibaca	ID card yang dibaca Sesuai dengan yang tertera
2	Kartu 2	ID kartu sesuai dengan yang dibaca	ID card yang dibaca Sesuai dengan yang tertera
3	Kartu 3	ID kartu sesuai dengan yang dibaca	ID card yang dibaca Sesuai dengan yang tertera
4	Kartu 4	ID kartu sesuai dengan yang dibaca	ID card yang dibaca Sesuai dengan yang tertera
5	Kartu 5	ID kartu sesuai dengan yang dibaca	ID card yang dibaca Sesuai dengan yang tertera



Gambar 4. 4 Proses pengujian pembacaan ID E-Tiket

4.1.2.2. Pengujian penyimpanan data E-Tiket Langganan

Dalam pengujian penyimpanan data E-Tiket langganan difokuskan dalam penyimpanan data E_Tiket ke dalam database Access melalui aplikasi yang peneliti untuk meghubungkan Gate dengan Database. Tujuan dari pengamatan ini untuk menghindari data yang disimpan ke dalam Database pengunjung langganan terjadi kesalahan pada proses penyimpanan .dapat dilihat pada Tabel 4.2 dan pada Gambar 4.4.

Tabel 4. 2 Pengujian penyimpanan data E-tiket Langganan

No	E-ticket	Kriteria Pengujian	Hasil Pengujian
1	Kartu 1	Data tersimpan di Database	ID Tersimpan di dalam database
2	Kartu 2	Data tersimpan di Database	ID Tersimpan di dalam database
3	Kartu 3	Data tersimpan di Database	ID Tersimpan di dalam database
4	Kartu 4	Data tersimpan di Database	ID Tersimpan di dalam database
5	Kartu 5	Data tersimpan di Database	ID Tersimpan di dalam database

Form3

Koneksi | Daftar Pengunjung | Pengunjung Langganan | Pengunjung Harian | Tambah Saldo | Edit Harga | User

ID: 84 A1 76 EF

Nama: idris

Tempat Lahir: bekasi

Tanggal Lahir: 25011994

Jenis Kelamin: LAKI_LAKI

Alamat: bekasi

Saldo Awal: 500000

RFID Card

Tambah

Edit

Hapus

Information

Data berhasil di Input

OK

ID	Nama	Tanggal_Lal	Tempat_Lal	Jenis_Kelamin	Alamat	Saldo
09 5B 5...	nurfy	01071993	banten	LAKI_LAKI		
44 5A 00...	hendra	26041995	jakarta	LAKI_LAKI	jakarta	1000000
29 1A 76...	suryana	29021993	cirebon	LAKI_LAKI	cirebon	10000

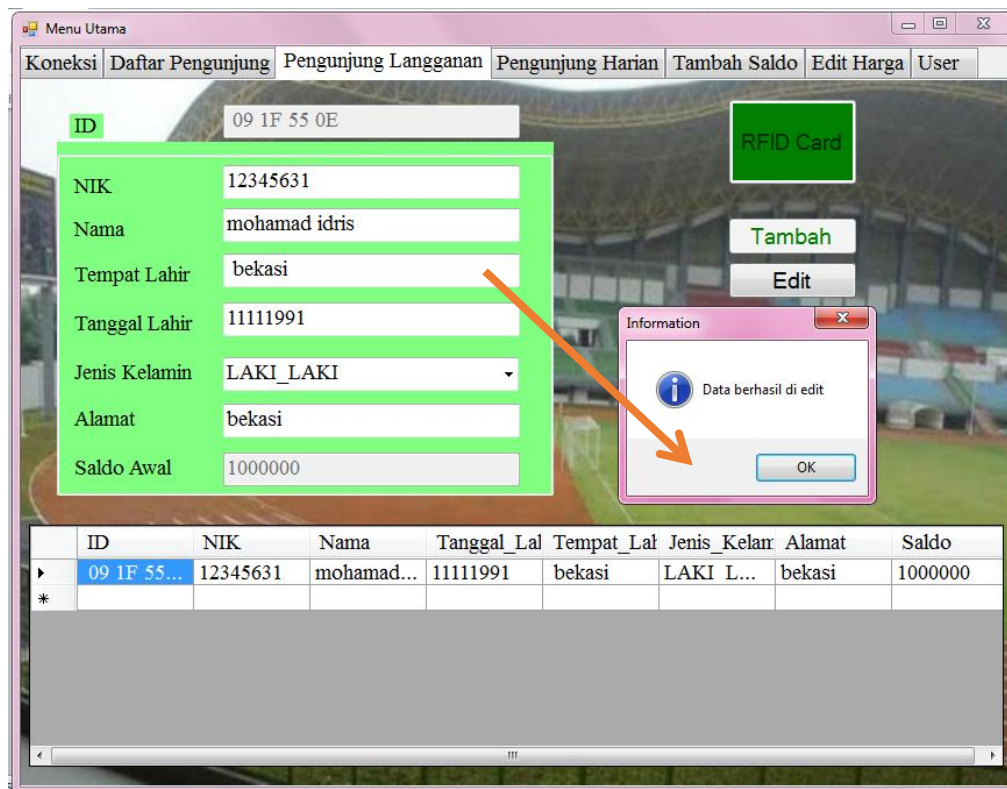
Gambar 4. 5 Data tersimpan pada database Pengunjung Langganan

4.1.2.3. Pengujian pengeditan data E-tiket Langganan

Dalam pengujian pengeditan data E-Tiket langganan difokuskan pada database pengunjung langganan yang telah tersimpan dan memanggil data sesuai dengan ID yang dimiliki E-Tiket. Tujuan dari pengujian ini untuk mengubah data yang telah tersimpan sebelumnya pada database Pengunjung Langganan. Hasil pengujian ini dapat dilihat pada Gambar 4.5 dan pada tabel 4.3

Tabel 4. 3 Pengujian pengeditan data E-tiket Langganan

No	E-ticket	Kriteria Pengujian	Hasil Pengujian
1	Kartu 1	Dapat mengedit data sesuai ID di dalam database	Data berhasil diedit ke dalam database
2	Kartu 2	Dapat mengedit data sesuai ID di dalam database	Data berhasil diedit ke dalam database
3	Kartu 3	Dapat mengedit data sesuai ID di dalam database	Data berhasil diedit ke dalam database
4	Kartu 4	Dapat mengedit data sesuai ID di dalam database	Data berhasil diedit ke dalam database
5	Kartu 5	Dapat mengedit data sesuai ID di dalam database	Data berhasil diedit ke dalam database



Gambar 4. 6 pengeditan pada database Pengunjung Langganan

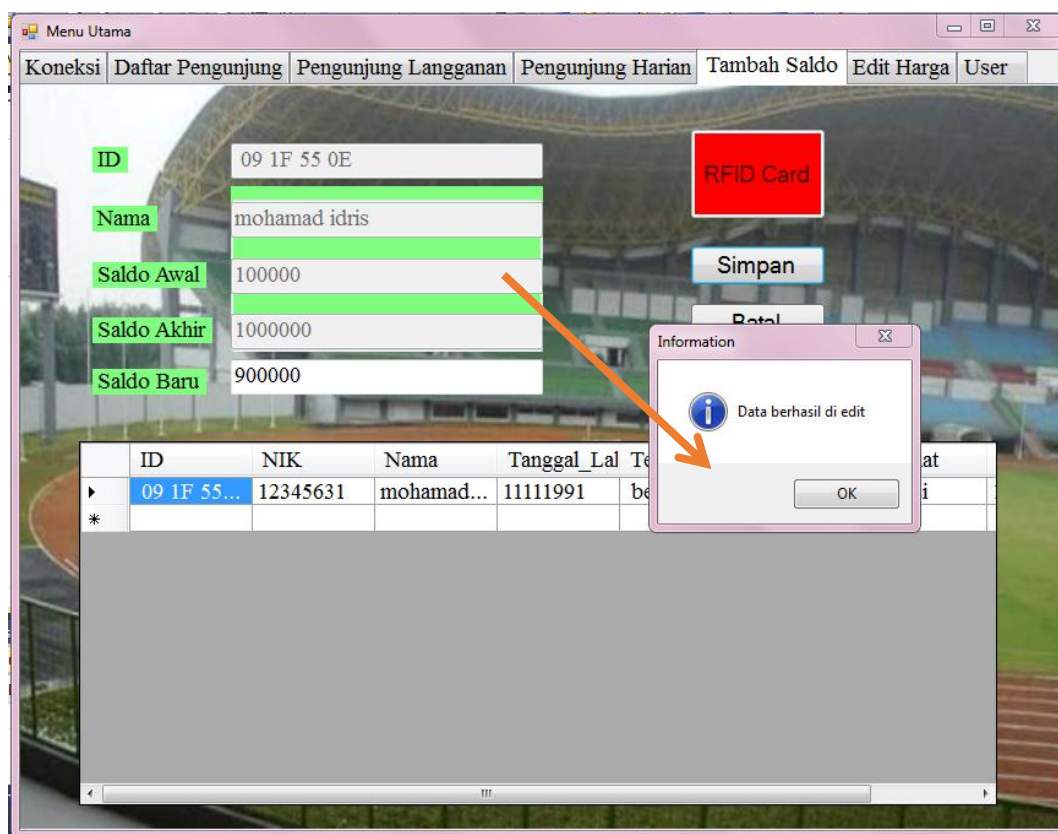
4.1.2.4. Pengujian penambahan saldo E-tiket Langganan

Dalam pengujian penambahan saldo E-Tiket langganan difokuskan pada database penngunjung langganan yang telah tersimpan dan memanggil data saldo sesuai dengan ID yang dimiliki E-Tiket yang telah tersimpan. Tujuan dari pengujian ini untuk menambahkan saldo yang terdapat pada E-Tiket dan memastikan data tersimpan pada database Pengunjung Langganan. Hasil pengujian ini dapat dilihat pada Gambar 4.6 dan tabel 4.4

Tabel 4. 4 Pengujian penambahan saldo E-tiket Langganan

No	E-ticket	Kriteria Pengujian	Hasil Pengujian
1	Kartu 1	Dapat menambahkan saldo sesuai ID E-tiket pada database	Saldo berhasil ditambahkan

2	Kartu 2	Dapat menambahkan saldo sesuai ID E-tiket pada database	Saldo berhasil ditambahkan
3	Kartu 3	Dapat menambahkan saldo sesuai ID E-tiket pada database	Saldo berhasil ditambahkan
4	Kartu 4	Dapat menambahkan saldo sesuai ID E-tiket pada database	Saldo berhasil ditambahkan
5	Kartu 5	Dapat menambahkan saldo sesuai ID E-tiket pada database	Saldo berhasil ditambahkan



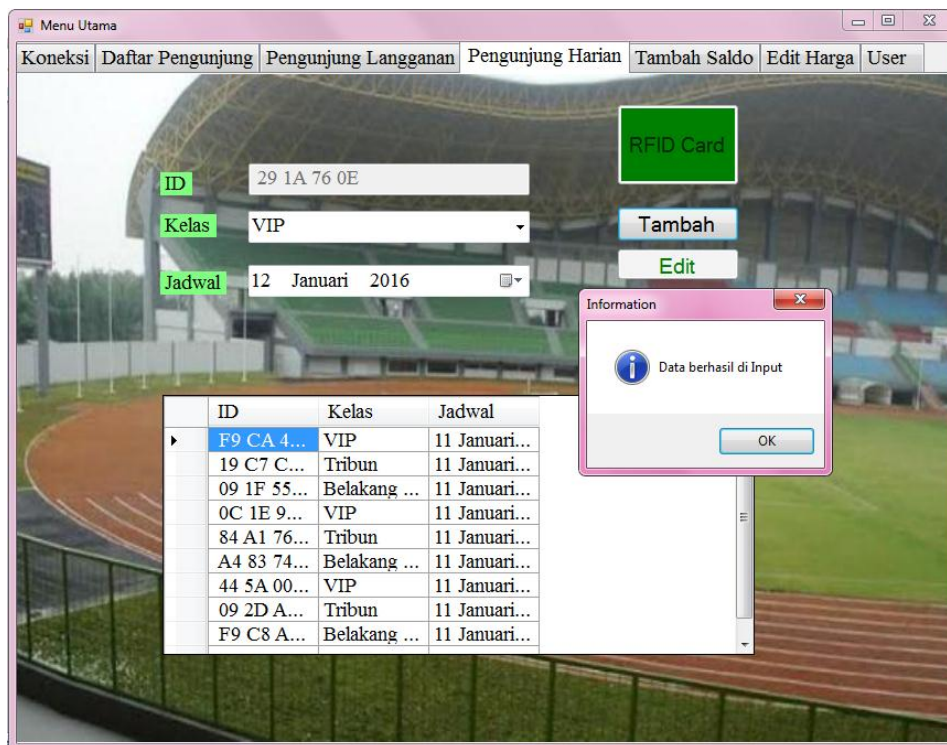
Gambar 4. 7 edit saldo langganan

4.1.2.5. Pengujian penyimpanan data E-tiket Harian

Dalam pengujian penyimpanan data E-Tiket Harian difokuskan dalam penyimpanan data E_Tiket ke dalam database Access melalui aplikasi yang peneliti untuk meghubungkan Gate dengan Database. Tujuan dari pengamatan ini untuk menghindari data yang disimpan ke dalam Database pengunjung Harian terjadi kesalahan pada proses penyimpanan .Tabel 4.5 dan Gambar 4.7

Tabel 4. 5 Pengujian penyimpanan data E-tiket Harian

No	E-tiket	Kriteria Pengujian	Hasil Pengujian
1	Kartu 1	Dapat menyimpan data kelas sesuai ID E-tiket	Data berhasil disimpan ke database pengunjung harian
2	Kartu 2	Dapat menyimpan data kelas sesuai ID E-tiket	Data berhasil disimpan ke database pengunjung harian
3	Kartu 3	Dapat menyimpan data kelas sesuai ID E-tiket	Data berhasil disimpan ke database pengunjung harian
4	Kartu 4	Dapat menyimpan data kelas sesuai ID E-tiket	Data berhasil disimpan ke database pengunjung harian
5	Kartu 5	Dapat menyimpan data kelas sesuai ID E-tiket	Data berhasil disimpan ke database pengunjung harian



Gambar 4. 8 pengunjung Harian

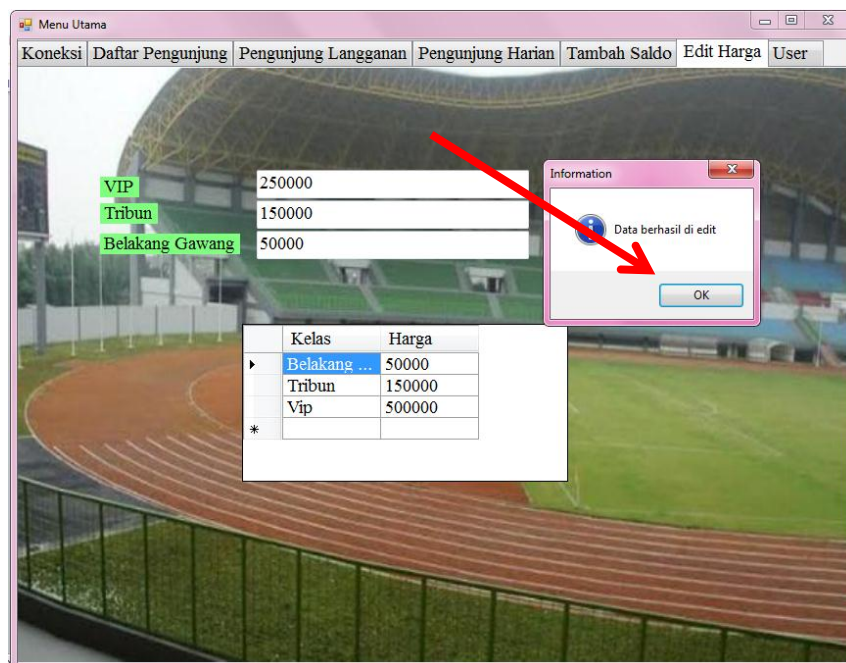
4.1.2.6. Pengujian pengeditan harga tiket

Dalam pengujian pengeditan Harga Tiket difokuskan pada database Harga Tiket Stadion dengan memanggil data sesuai dengan kelas. Tujuan dari pengujian ini untuk mengubah data yang telah tersimpan sebelumnya pada database Harga

Tiket agar sesuai dengan harga yang semestinya. Hasil pengujian ini dapat dilihat pada Gambar 4.8 dan pada tabel 4.6

Tabel 4. 6 Pengujian pengeditan harga tiket

No	Kriteria Pengujian	Hasil Pengujian
1	Dapat mengedit harga tiket sesuai inputan server	Harga tiket berhasil dirubah
2	Dapat mengedit harga tiket sesuai inputan server	Harga tiket berhasil dirubah
3	Dapat mengedit harga tiket sesuai inputan server	Harga tiket berhasil dirubah
4	Dapat mengedit harga tiket sesuai inputan server	Harga tiket berhasil dirubah
5	Dapat mengedit harga tiket sesuai inputan server	Harga tiket berhasil dirubah





Gambar 4. 9 edit harga tiket




4.1.2.7. Pengujian gate saldo E-tiket langganan mencukupi

Dalam pengujian Gate menggunakan E-tiket langganan dengan saldo yang mencukupi untuk melakukan pembelian tiket secara langsung. Proses ini dilakukan dengan gate yang mengirim data ID E-Tiket melalui arduino ke PC dan PC

melalui aplikasi Tiketing mengirimkan data ke arduino untuk menyalakan LED Hijau dan menggerakkan servo apabila Saldo yang dimiliki E-Tiket langganan mencukupi. Tujuan pengujian ini agar E-Tiket yang memiliki Saldo yang mencukupi dapat masuk ke dalam stadion sesuai dengan kelasnya. Hasil pengujian ini dapat dilihat pada tabel 4.7

Tabel 4. 7 Pengujian gate saldo E-tiket langganan mencukupi

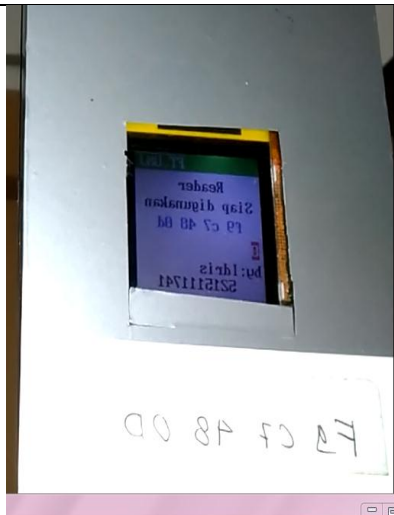
No	E-ticket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Menampilkan data saldo pada layar LCD dan LED hijau On		Led hijau on, LCD menampilkan sisa saldo dan ID card
2	Kartu 2	Menampilkan data saldo pada layar LCD dan LED hijau On		Led hijau on, LCD menampilkan sisa saldo dan ID card

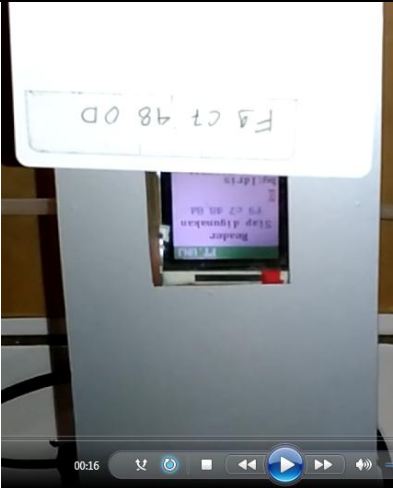

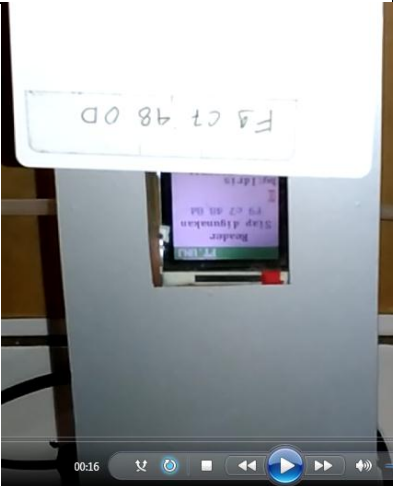
3	Kartu 3	Menampilkan data saldo pada layar LCD dan LED hijau On		Led hijau on, LCD menampilkan sisa saldo dan ID card
4	Kartu 4	Menampilkan data saldo pada layar LCD dan LED hijau On		Led hijau on, LCD menampilkan sisa saldo dan ID card
5	Kartu 5	Menampilkan data saldo pada layar LCD dan LED hijau On		Led hijau on, LCD menampilkan sisa saldo dan ID card


4.1.2.8. Pengujian gate saldo E-tiket langganan tidak mencukupi

Dalam pengujian Gate menggunakan E-tiket langganan dengan saldo yang tidak mencukupi untuk melakukan pembelian tiket secara langsung. Proses ini dilakukan dengan gate yang mengirim data ID E-Tiket melalui arduino ke PC dan PC melalui aplikasi Tiketing mengirimkan data ke arduino untuk menyalakan LED Merah dan tidak menggerakkan servo karena Saldo yang dimiliki E-Tiket langganan tidak mencukupi. Tujuan pengujian ini agar E-Tiket yang memiliki Saldo kurang dari harga tiket tidak dapat masuk ke dalam stadion sesuai dengan kelasnya. Hasil pengujian ini dapat dilihat pada tabel 4.8

Tabel 4. 8 Pengujian gate saldo E-tiket langganan tidak mencukupi

No	E-tiket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Menampilkan data saldo pada layar LCD dan LED merah tetap On		LED merah on, LCD menampilkan ID Card dan sisa saldo


2	Kartu 2	Menampilkan data saldo pada layar LCD dan LED merah tetap On		LED merah on, LCD menampilkan ID Card dan sisa saldo
3	Kartu 3	Menampilkan data saldo pada layar LCD dan LED merah tetap On		LED merah on, LCD menampilkan ID Card dan sisa saldo
4	Kartu 4	Menampilkan data saldo pada layar LCD dan LED merah tetap On		LED merah on, LCD menampilkan ID Card dan sisa saldo


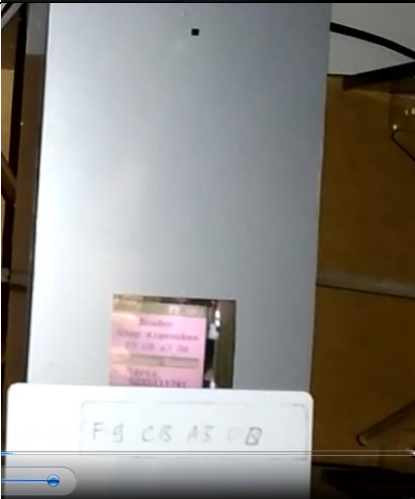

5	Kartu 5	Menampilkan data saldo pada layar LCD dan LED merah tetap On		LED merah on, LCD menampilkan ID Card dan sisa saldo
---	------------	--	--	--

4.1.2.9. Pengujian gate E-tiket harian sesuai dengan kelas

Dalam pengujian gate dengan menggunakan E-Tiket Harian sesuai dengan kelas berfokus pada tampilan pada LCD Gate. Pengujian ini dilakukan dengan menggunakan E-Tiket yang sudah di data pada aplikasi Tiketing dengan mengirimkan ID E-Tiket dari Arduino Ke PC sehingga pada prosesnya harus sesuai dengan intruksi yang diinginkan dengan menyalakan LED Hijau pada gate. Tujuan pengujian ini agar dapat memisahkan E-Tiket dengan E-Tiket yang lain. Hasil pengujian ini dapat dilihat pada tabel 4.9.

Tabel 4. 9 Pengujian gate E-tiket harian sesuai dengan kelas

No	E-tiket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Menampilk an kelas pada layar LCD dan LED hijau On		Led hijau ON, LCD menampilkan ID card dan tipe kelas

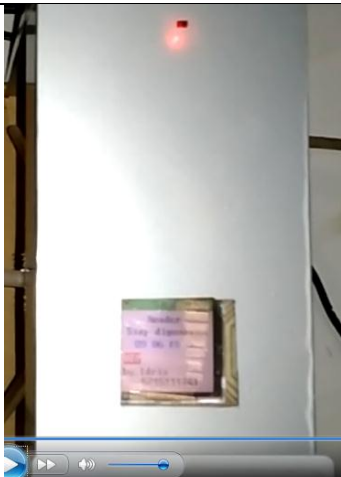
2	Kartu 2	Menampilk an kelas pada layar LCD dan LED hijau On		Led hijau ON, LCD menampilkan ID card dan tipe kelas
3	Kartu 3	Menampilk an kelas pada layar LCD dan LED hijau On		Led hijau ON, LCD menampilkan ID card dan tipe kelas
4	Kartu 4	Menampilk an kelas pada layar LCD dan LED hijau On		Led hijau ON, LCD menampilkan ID card dan tipe kelas




5	Kartu 5	Menampilk an kelas pada layar LCD dan LED hijau On		Led hijau ON, LCD menampilkan ID card dan tipe kelas
---	------------	---	--	---


4.1.2.10. Pengujian gate E-tiket harian sesuai dengan kelas

Dalam pengujian gate dengan menggunakan E-Tiket Harian yang tidak sesuai dengan kelas berfokus pada tampilan pada LCD Gate. Pengujian ini dilakukan dengan menggunakan E-Tiket yang sudah di data pada aplikasi Tiketing dan E-Tiket yang belum didaftarkan pada aplikasi Tiketing dengan mengirimkan ID E-Tiket dari Arduino Ke PC sehingga pada prosesnya harus sesuai dengan intruksi yang diinginkan. Tujuan pengujian ini agar dapat memisahkan E-Tiket dengan E-Tiket yang lain. Hasil pengujian ini dapat dilihat pada tabel 4.10.

Tabel 4. 10 Pengujian gate E-tiket harian sesuai dengan kelas

No	E- ticket	Kriteria Pengujian	Gambar Hasil Pengujian	Hasil Pengujian
1	Kartu 1	Menampilkan kelas pada layar LCD dan LED merah tetap On		Led merah ON, LCD menampilkan ID card dan tipe kelas

2	Kartu 2	Menampilkan kelas pada layar LCD dan LED merah tetap On		Led merah ON, LCD menampilkan ID card dan tipe kelas
3	Kartu 3	Menampilkan kelas pada layar LCD dan LED merah tetap On		Led merah ON, LCD menampilkan ID card dan tipe kelas
4	Kartu 4	Menampilkan kelas pada layar LCD dan LED merah tetap On		Led merah ON, LCD menampilkan ID card dan tipe kelas

5	Kartu 5	Menampilkan kelas pada layar LCD dan LED merah tetap On		Led merah ON, LCD menampilkan ID card dan tipe kelas
---	------------	---	--	--

4.2. Kelebihan dan Kekurangan Alat

4.2.1. Kelebihan Alat

Dari hasil penelitian dan pembahasan, maka alat dan *software* server tiketing yang dibuat memiliki beberapa kelebihan, antara lain:

1. Dengan sistem tiketing elektronik prosedur pembayaran tiket dapat dilakukan lebih efektif karena data di simpan di dalam pada perangkat elektronik.
2. Desain prototipe lebih efektif apabila diaplikasikan dalam bentuk yang sebenarnya untuk mencegah penonton yang hendak menerobos.
3. Pencatatan data pengunjung dilakukan secara elektronik melalui aplikasi Tiketing sehingga dapat lebih efisien.
4. Data E-Tiket disimpan pada database sehingga pengunjung stadion yang hendak masuk tidak sesuai dengan tiket yang dimiliki tidak dapat masuk kedalam stadion.

4.2.2. Kekurangan Alat

Dari beberapa kelebihan diatas, alat dan *software* server Tiketing yang telah dibuat masih memiliki beberapa kekurangan, antara lain:

1. Alat tiketing harus di pasangkan dengan PC yang memiliki prosesor yang baik agar proses transfer data dari pc ke gate lebih cepat.
2. Alat ini harus selalu terhubung dengan PC melalui kabel data
3. Untuk membeleli E-Tiket harian dan menambahkan saldo Langgan masih harus ke loket di stadion karena belum dilengkapi dengan koneksi internet atau website.
4. Tampilan alat masih kurang menarik.

Tampilan *Software* server tiketing masih kurang menarik.

BAB V

KESIMPULAN DAN SARAN

1.1 Kesimpulan

Setelah melakukan perancangan, implementasi, pengujian, dan analisa dapat diambil kesimpulan sebagai berikut :

1. Sistem tiket elektronik menggunakan RFID telah sesuai dengan perencanaan dan tujuan dari penelitian. Sistem pembayaran tiket elektronik memiliki perangkat keras dan perangkat lunak sehingga menjadi satu sistem yang dapat digunakan sebagai gate stadion dan sebagai pencatatan data pengunjung Stadion.
2. Sistem tiketing stadion menggunakan arduino uno sebagai mikrokontroler arduino uno pada gate dengan input berupa RFID *reader* sebagai pembaca ID kartu E-Tiket ID digunakan untuk mengakses data pada server tiketing, sehingga data E-Tiket dapat diolah kedalam *database* dan *limitswitch* sebagai tanda bahwa pengunjung telah melewati gate.
3. Sistem tiketing stadion ini didukung dengan perangkat lunak menggunakan aplikasi Tiketing yang dibuat oleh peneliti dengan menggunakan software Visual Basic 2010 dan *database* menggunakan software sehingga data pengunjung dapat disimpan secara elektronik dan nantinya dapat digunakan pihak manajemen untuk menganalisa profil sebaran pengunjung.

1.2 Saran

Perancangan sistem tiketing menggunakan RFID, memiliki saran diantaranya :

1. Sebaiknya PC yang digunakan memiliki ram minimal 2 GB sehingga proses transmisi data dari arduino ke PC dapat lebih cepat.
2. Sebaiknya suplai daya arduino ditambah apabila gate yang digunakan lebih banyak.
3. Aplikasi tiketing dikemas tampilannya agar lebih menarik lagi dan Aplikasi tiketing dilengkapi dengan fitur website sehingga penambahan saldo tidak harus di stadion dan pembelian tiket dapat dilakukan dimana saja.

Daftar Pustaka

- prototipe*. (2015, November 25). Diambil kembali dari kbbs: <http://kbbs.web.id/prototipe>
- Andress, S. (2008). *Creating Limitless Wealth*. Jakarta: McGraw-hill.
- Dr.Leong, Balboa, M., & Andersen. (2013). ARDUINO RFID MODULE KIT 13.56 MHZ.
- Eridani, D. (2012). *Simulasi Gerbang Tol Menggunakan RFID (Radio Frequency Indetification)*. Semarang: Universitas Diponegoro.
- Kadir, A. (2013). *Panduan Praktis Mempelajari Mikrokontroler dan Pemrogramannya Menggunakan Arduino*. Yogyakarta: Andi.
- Kinanti, F. (2013). ANALISIS DETERMINAN SISTEM INFORMASI E-TICKETING : PENDEKATAN. *jurnal*, 6.
- Robby. (2004). Analisis dan Perancangan Basis Data untuk Mendukung Aplikasi ERP Education pada Bina Nusantara University. 1.
- Sproul, M., & Davenport, P. (t.thn.). SparkFun's Color LCD Shield.
- Sugiyono. (2009). *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung: ALFABETA.
- Ikky (2016, January Kamis)diambil kembali dari <https://ikky17.files.wordpress.com/2010/12/rfid.jpg>

Lampiran

List program VB

```
Imports System.Data.OleDb
Public Class Form2
    Dim CONN As OleDbConnection
    Dim CMD As OleDbCommand
    Dim DS As New DataSet
    Dim DA As OleDbDataAdapter
    Dim RD As OleDbDataReader
    Dim LokasiDB As String
    Sub Koneksi()
        LokasiDB = "Provider =
Microsoft.ACE.OLEDB.12.0;Data
Source = Tiketing.accdb"
        CONN = New
OleDbConnection(LokasiDB)
        If CONN.State =
ConnectionState.Closed Then
CONN.Open()
        End Sub
    Private Sub
TextBox1_KeyPress(ByVal sender
As Object, ByVal e As
System.Windows.Forms.KeyPressEve
ntArgs) Handles
TextBox1.KeyPress
        If e.KeyChar = Chr(13)
Then TextBox2.Focus()
        End Sub
    Private Sub
TextBox2_KeyPress(ByVal sender
As Object, ByVal e As
System.Windows.Forms.KeyPressEve
ntArgs) Handles
TextBox2.KeyPress
        If e.KeyChar = Chr(13)
Then Button1.Focus()
        End Sub
    Private Sub
Button1_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
Button1.Click
        If TextBox1.Text = "" Or
TextBox2.Text = "" Then
            MsgBox("data login
belum lengkap")
            Exit Sub
        Else
            Call Koneksi()
            CMD = New
OleDbCommand("select * from
login where user =' " &
TextBox1.Text & "' and password
=' " & TextBox2.Text & "'", CONN)
```

```
RD =
CMD.ExecuteReader
RD.Read()
If RD.HasRows Then
    Form3.Show()
Else
    MsgBox("Kode
User atau Password salah")
    TextBox1.Text =
""
    TextBox2.Text =
""
        End If
    End If
End Sub

Private Sub
Button2_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
Button2.Click
    Me.Close()
End Sub

Private Sub
Form2_Activated(ByVal sender As
Object, ByVal e As
System.EventArgs) Handles
Me.Activated
    TextBox1.Focus()
End Sub

Private Sub Form2_Load(ByVal
sender As System.Object, ByVal e
As System.EventArgs) Handles
MyBase.Load
    TextBox1.MaxLength = 6
    TextBox2.PasswordChar =
"*"
    TextBox1.Clear()
    TextBox2.Clear()
End Sub
End Class

Imports System.Data.OleDb
Imports System.IO.Ports
Public Class Form3
    Dim CONN As OleDbConnection
    Dim CMD As OleDbCommand
    Dim DS As New DataSet
    Dim DA As OleDbDataAdapter
    Dim RD As OleDbDataReader
    Dim LokasiDB As String
    Dim VA As String
    Dim A As Single
```

```

    Dim b As Single

    Delegate Sub
myMethodDelegate(ByVal [text] As
String)
    Private Delegate Sub
delegateReadData()
    Dim comOpen As Boolean
'Keeps track of the port status.
True = Open; False = Closed
    Dim readbuffer As String
'Buffer of whats read from the
serial port
    Sub k()
        lbinfo.Text = ""
    End Sub
    Sub Koneksi()
        LokasiDB = "Provider =
Microsoft.ACE.OLEDB.12.0;Data
Source = Tiketing.accdb"
        CONN = New
OleDbConnection(LokasiDB)
        If CONN.State =
ConnectionState.Closed Then
CONN.Open()
    End Sub
    Sub kosongkan()
        TextBox1.Clear()
        TextBox2.Clear()
        TextBox3.Clear()
        TextBox4.Clear()
        TextBox5.Clear()
        TextBox6.Clear()
        TextBox7.Clear()
        TextBox8.Clear()
        TextBox9.Clear()
        TextBox10.Clear()
        TextBox11.Clear()
        TextBox12.Clear()
        TextBox17.Clear()
        TextBox18.Clear()
        TextBox19.Clear()
    End Sub
    Sub kondisiawal()
        Koneksi()
        DA = New
OleDbDataAdapter("select * from
DPL", CONN)
        DS = New DataSet
        DS.Clear()
        DA.Fill(DS, "DPL")

        DataGridView2.DataSource
= (DS.Tables("DPL"))
        DataGridView4.DataSource
= (DS.Tables("DPL"))
    End Sub

    Sub kondisiawal1()
        Koneksi()
        DA = New
OleDbDataAdapter("select * from
DPH", CONN)
        DS = New DataSet
        DS.Clear()
        DA.Fill(DS, "DPH")
        DataGridView3.DataSource
= (DS.Tables("DPH"))
    End Sub
    Sub kondisiawal2()
        Koneksi()
        DA = New
OleDbDataAdapter("select * from
login", CONN)
        DS = New DataSet
        DS.Clear()
        DA.Fill(DS, "login")
        DataGridView6.DataSource
= (DS.Tables("login"))
    End Sub
    Sub kondisiawal3()
        Koneksi()
        DA = New
OleDbDataAdapter("select * from
HRG", CONN)
        DS = New DataSet
        DS.Clear()
        DA.Fill(DS, "HRG")
        DataGridView5.DataSource
= (DS.Tables("HRG"))
    End Sub
    Sub kondisiawal4()
        Koneksi()
        DA = New
OleDbDataAdapter("select * from
CP", CONN)
        DS = New DataSet
        DS.Clear()
        DA.Fill(DS, "CP")
        DataGridView5.DataSource
= (DS.Tables("CP"))
    End Sub
    Private Sub
frmDisplay_FormClosing(ByVal
sender As Object, ByVal e As
FormClosingEventArgs) Handles
Me.FormClosing
        'Gracefully disconnect
before form closes
        DoDisconnect()
    End Sub

    Public Sub update1(ByVal
sender As Object, ByVal e As
System.EventArgs)

```



```

        'Update temperature
display as it comes in
        Dim read1 As String
        read1 =
readbuffer.Replace(vbCr,
""").Replace(vbLf, "")
        lbInfo.Text = read1
        End Sub
        Private Sub
SerialPort1_(ByVal sender As
System.Object, ByVal e As
System.IO.Ports.SerialDataReceiv
edEventArgs) Handles
SerialPort1.DataReceived
        If comOpen Then
            Try
                'Send data to a
new thread to update the
temperature display
                readbuffer =
SerialPort1.ReadLine()
                Me.Invoke(New
EventHandler(AddressOf update1))
            Catch ex As
Exception
                'Otherwise show
error. Will display when
disconnecting.

                'MsgBox(ex.Message)
                End Try
            End If
        End Sub
        Public Sub DoDisconnect()
            'Graceful disconnect if
port is open
            If comOpen Then

SerialPort1.DiscardInBuffer()
SerialPort1.Close()

                'Reset our flag and
controls
                comOpen = False

            End If
        End Sub

        Public Sub DoConnect()
            'Setup the serial port
connection
            With SerialPort1
                .PortName =
cbPort.Text
                'Selected Port

                .BaudRate =
CInt(cbBoudRate.Text) 'Baud
Rate. 9600 is default.
                .Parity =
IO.Ports.Parity.None
                .DataBits = 8
                .StopBits =
IO.Ports.StopBits.One
                .Handshake =
IO.Ports.Handshake.None
                .RtsEnable = False

                .ReceivedBytesThreshold = 1
                .NewLine = vbCr
                .ReadTimeout = 10000
            End With
            'Try to open the
selected port...
            Try
                SerialPort1.Open()
                comOpen =
SerialPort1.IsOpen

            Catch ex As Exception
                comOpen = False
                MsgBox("Error Open:
" & ex.Message)
            End Try

            cbPort.Enabled = False
            cbBoudRate.Enabled =
False
            btStart.Text = "Stop"
            btStart.BackColor =
Color.Red
        End Sub
        Private Sub Form3_Load(ByVal
sender As Object, ByVal e As
EventArgs) Handles MyBase.Load
            Call kondisiawal()
            Call kondisiawal1()
            Call kondisiawal2()
            Call kondisiawal3()
            Form1.Hide()
            Dim comPorts As String()
            =
System.IO.Ports.SerialPort.GetPo
rtNames
            If comPorts.Count < 1
Then
                'If there are not
ports connected, show an error
and close the program.
                MsgBox("There are no
com ports available! Closing
program.")
                Me.Close()
            End If
        End Sub
    End Class

```

```

        Else
            cbPort.Items.AddRange(comPorts)
            cbPort.Text =
comPorts(0)
            End If
            Timer2.Start()
        End Sub

        Private Sub
Button6_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button6.Click
            Button5.Enabled = False
            Button10.Enabled = True
            Button8.Enabled = True
            Button7.Enabled = True
            GroupBox1.Enabled = True

        End Sub
        Private Sub
Button9_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button9.Click
            If btStart.Text =
"Start" Then

MessageBox.Show("klik tombol
START pada Koneksi")
                Exit Sub
            ElseIf Button9.BackColor
= Color.Red Then
                MsgBox("Tempelkan
Kartu Pada Reader")
                Try
                    Timer1.Start()
                    Timer2.Start()

                Button9.BackColor = Color.Green
                Button9.Enabled
= False

                Catch ex As
Exception
                End Try
            End If

        End Sub

        Private Sub
Button7_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button7.Click
            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox3.Text = ""
            TextBox4.Text = ""
            TextBox5.Text = ""

            TextBox6.Text = ""
            TextBox7.Text = ""
            ComboBox1.Text = "Jenis
Kelamin"

            Button8.Enabled = False
            Button6.Enabled = False
            Button5.Enabled = False
            Button16.Visible = False
            Button10.Enabled = False
            GroupBox1.Enabled =
False

            Button7.Enabled = False
            Button9.Enabled = True
            Button9.BackColor =
Color.Red
        End Sub
        Private Sub
Button4_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button4.Click
            If CheckBox1.Checked =
False Then
                Koneksi()
                DA = New
OleDbDataAdapter("select * from
Daftar_Pengunjung where waktu
=' ' & DateTimePicker1.Text & ' ',
Kelas = ' ' & ComboBox3.Text &
' ' ' ' , CONN)
                DS = New DataSet
                DS.Clear()
                DA.Fill(DS,
"Daftar_Pengunjung")

                DataGridView1.DataSource =
(DS.Tables("Daftar_Pengunjung"))
            End If
            If CheckBox1.Checked =
True Then
                Koneksi()
                DA = New
OleDbDataAdapter("select * from
Daftar_Pengunjung where waktu
=' ' & DateTimePicker1.Text &
' ' ' ' , CONN)
                DS = New DataSet
                DS.Clear()
                DA.Fill(DS,
"Daftar_Pengunjung")

                DataGridView1.DataSource =
(DS.Tables("Daftar_Pengunjung"))
                ComboBox3.Enabled =
True
            End If
            ComboBox3.Enabled = True

```

```

        CheckBox1.Checked =
False
    End Sub

    Private Sub
Button5_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button5.Click
        Button6.Enabled = False
        GroupBox1.Enabled = True
        Button16.Visible = True
        Button7.Enabled = True
        TextBox6.Enabled = True
    End Sub

    Private Sub
TextBox1_TextChanged(ByVal
sender As Object, ByVal e As
EventArgs) Handles
TextBox1.TextChanged
        If TextBox1.Text =
Chr(13) Then Call Koneksi()
        CMD = New
OleDbCommand("select *from DPL
where ID =' " & TextBox1.Text &
"', CONN)
        RD = CMD.ExecuteReader()
        RD.Read()
        If Not RD.HasRows Then
            TextBox2.Text = ""
            TextBox3.Text = ""
            TextBox4.Text = ""
            ComboBox1.Text =
"Jenis Kelamin"
            TextBox5.Text = ""
            TextBox6.Text = ""
            TextBox7.Text = ""
            TextBox2.Focus()
            Button5.Enabled =
True

            Timer2.Stop()
        Else
            TextBox7.Text =
RD.Item("NIK")
            TextBox2.Text =
RD.Item("Nama")
            TextBox3.Text =
RD.Item("Tempat_Lahir")
            TextBox4.Text =
RD.Item("Tanggal_Lahir")
            TextBox5.Text =
RD.Item("Alamat")
            TextBox6.Enabled =
False
            TextBox6.Text =
RD.Item("Saldo")
            ComboBox1.Text =
RD.Item("Jenis_Kelamin")
            TextBox2.Focus()
            Button6.Enabled =
True
            Button7.Enabled =
True
            Timer2.Stop()
        End If
    End Sub

    Private Sub
Button16_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button16.Click
        If TextBox1.Text = "" Or
TextBox2.Text = "" Or
TextBox3.Text = "" Or
TextBox4.Text = "" Or
TextBox5.Text = "" Or
TextBox6.Text = "" Or
ComboBox1.Text = "Jenis Kelamin"
Then
            MsgBox("Data Belum
Lengkap, Pastikan Semua Data
Sudah Terisi")
            Exit Sub
        Else
            Call Koneksi()
            Dim simpan As String
            = "insert into DPL
(ID,NIK>Nama,Tempat_Lahir,Tangga
l_Lahir,Jenis_Kelamin,Alamat,Sal
do) values (' " & TextBox1.Text &
"', " & TextBox7.Text & "', " &
TextBox2.Text & "', " &
TextBox3.Text & "', " &
TextBox4.Text & "', " &
ComboBox1.Text & "', " &
TextBox5.Text & "', " &
TextBox6.Text & "');"
            CMD = New
OleDbCommand(simpan, CONN)

            CMD.ExecuteNonQuery()
            Dim simpana As
String = "insert into CP
(ID,Saldo,max_taping) values (' "
& TextBox1.Text & "', " &
TextBox6.Text & "', " & "4" &
"');"
            CMD = New
OleDbCommand(simpana, CONN)

            CMD.ExecuteNonQuery()
            Call Koneksi()

```

```

        MsgBox("Data
berhasil di Input",
MsgBoxStyle.Information,
"Information")
        Call kondisiawal()
        Call kosongkan()
        ComboBox1.Text =
"Jenis Kelamin"
        Button7.Enabled =
False
        GroupBox1.Enabled =
False
        Button5.Enabled =
False
        Button6.Enabled =
False
        Button16.Visible =
False
        Button9.Enabled =
True
        Button9.BackColor =
Color.Red
    End If
End Sub

Private Sub
Button10_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button10.Click
    If
MessageBox.Show("Yakin akan
dihapus..?", "",
MessageBoxButtons.YesNo) =
Windows.Forms.DialogResult.Yes
Then
        Dim hapus As String
= "delete * from DPL where ID='"
& TextBox1.Text & "'"
        CMD = New
OleDbCommand(hapus, CONN)

CMD.ExecuteNonQuery()
        Dim hapusa As String
= "delete * from CP where ID='"
& TextBox1.Text & "'"
        CMD = New
OleDbCommand(hapusa, CONN)

CMD.ExecuteNonQuery()
        MsgBox("Data
berhasil di Hapus",
MsgBoxStyle.Information,
"Information")
        Call kosongkan()
        Call kondisiawal()
        Button9.BackColor =
Color.Red

        Button9.Enabled =
True
    Else
        Call kosongkan()
    End If
End Sub

Private Sub
Button8_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button8.Click
    If TextBox1.Text = "" Or
TextBox2.Text = "" Or
TextBox7.Text = "" Or
TextBox3.Text = "" Or
TextBox4.Text = "" Or
TextBox5.Text = "" Or
TextBox6.Text = "" Or
ComboBox1.Text = "Jenis Kelamin"
Then
        MsgBox("Data Belum
Lengkap, Pastikan Semua Data
Sudah Terisi")
        Exit Sub
    Else
        Call Koneksi()
        Dim edit As String =
"Update DPL set NIK ='" &
TextBox7.Text & "',Nama ='" &
TextBox2.Text & "',Tempat_Lahir
='" & TextBox3.Text &
"',Tanggal_Lahir ='" &
TextBox4.Text & "',Jenis_Kelamin
='" & ComboBox1.Text & "',Alamat
='" & TextBox5.Text & "',Saldo
='" & TextBox6.Text & "' where
ID ='" & TextBox1.Text & "'"
        CMD = New
OleDbCommand(edit, CONN)

CMD.ExecuteNonQuery()
        Call Koneksi()
        MsgBox("Data
berhasil di edit",
MsgBoxStyle.Information,
"Information")
        Call kondisiawal()
        Call kosongkan()
        ComboBox1.Text =
"Jenis Kelamin"
        Button8.Enabled =
False
        Button7.Enabled =
False
        Button9.Enabled =
True
        Button9.BackColor =
Color.Red
    End If
End Sub

```

```

GroupBox1.Enabled =
False
    TextBox6.Visible =
True
    Button6.Enabled =
False
        End If
    End Sub

    Private Sub
Button13_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button13.Click
    If TextBox8.Text = "" Or
ComboBox2.Text = "Kelas" Then
        MsgBox("Data Belum
Lengkap, Pastikan Semua Data
Sudah Terisi")
        Exit Sub
    Else
        Call Koneksi()
        Dim edit As String =
"Update DPH set Jadwal =" &
DTP1.Text & ",Kelas =" &
ComboBox2.Text & " where ID =" &
TextBox8.Text & ""
        CMD = New
OleDbCommand(edit, CONN)

CMD.ExecuteNonQuery()
        Dim edita As String
= "Update CP set Saldo =" &
ComboBox2.Text & ",Jadwal =" &
DTP1.Text & " where ID =" &
TextBox8.Text & ""
        CMD = New
OleDbCommand(edita, CONN)

CMD.ExecuteNonQuery()
        Call Koneksi()
        MsgBox("Data
berhasil di edit",
MsgBoxStyle.Information,
"Information")
        Call kondisiawal1()
        TextBox8.Text = ""
        ComboBox2.Text =
"Kelas"
        Button22.BackColor =
Color.Red
        Button22.Enabled =
True
        Button13.Enabled =
False
        Button14.Enabled =
False
    End Sub

Button12.Enabled =
False
    End If
    End Sub
    Private Sub
Button14_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button14.Click
    If TextBox8.Text = "" Or
ComboBox2.Text = "Kelas" Then
        MsgBox("Data Belum
Lengkap, Pastikan Semua Data
Sudah Terisi")
        Exit Sub
    Else
        Call Koneksi()
        Dim simpan As String
= "insert into DPH
(ID,Jadwal,Kelas) values ('" &
TextBox8.Text & "','" &
DTP1.Text & "','" &
ComboBox2.Text & ');"
        CMD = New
OleDbCommand(simpan, CONN)

CMD.ExecuteNonQuery()
        Dim simpanan As
String = "insert into CP
(ID,jadwal,Saldo,max_taping)
values ('" & TextBox8.Text &
"',"' & DTP1.Text & "','" &
ComboBox2.Text & "','" & "1" &
');"
        CMD = New
OleDbCommand(simpanan, CONN)

CMD.ExecuteNonQuery()
        Call Koneksi()
        MsgBox("Data
berhasil di Input",
MsgBoxStyle.Information,
"Information")
        Call kondisiawal1()
        TextBox8.Text = ""
        ComboBox2.Text =
"Kelas"
        Button22.BackColor =
Color.Red
        Button22.Enabled =
True
        Button12.Enabled =
False
        Button14.Enabled =
False
    End If
    End Sub

```

```

        Private Sub
        TextBox8_TextChanged(ByVal
sender As Object, ByVal e As
EventArgs) Handles
        TextBox8.TextChanged
            If TextBox8.Text =
Chr(13) Then Call Koneksi()
                CMD = New
OleDbCommand("select *from DPH
where ID =' " & TextBox8.Text &
"', CONN)
                RD = CMD.ExecuteReader()
                RD.Read()
                If Not RD.HasRows Then
                    DTP1.Text = ""
                    ComboBox2.Text =
"kelas"
                    ComboBox2.Focus()
                    Button14.Enabled =
True
                    Button12.Enabled =
True
                    Timer2.Stop()
                Else
                    DTP1.Text =
RD.Item("Jadwal")
                    ComboBox2.Text =
RD.Item("Kelas")
                    DTP1.Focus()
                    Button12.Enabled =
True
                    Button13.Enabled =
True
                    Timer2.Stop()
                End If
            End Sub

        Private Sub
        TextBox19_TextChanged(ByVal
sender As Object, ByVal e As
EventArgs) Handles
        TextBox19.TextChanged
            If TextBox19.Text =
Chr(13) Then Call Koneksi()
                CMD = New
OleDbCommand("select *from DPL
where ID =' " & TextBox19.Text &
"', CONN)
                RD = CMD.ExecuteReader()
                RD.Read()
                If Not RD.HasRows Then
                    TextBox10.Text = ""
                    TextBox9.Text = ""
                    TextBox11.Text = ""
                    TextBox12.Text = ""
                    TextBox11.Focus()
                    Timer2.Stop()
                Else
                    TextBox10.Text =
RD.Item("Saldo")
                    TextBox11.Text =
RD.Item("Nama")
                    TextBox9.Text =
RD.Item("Saldo")
                    TextBox11.Focus()
                    Timer2.Stop()
                    Button24.Enabled =
True
                End If
            End Sub

        Private Sub
        Button15_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button15.Click
            If TextBox10.Text = ""
Or TextBox12.Text = "" Or
TextBox11.Text = "" Then
                MsgBox("Data Belum
Lengkap, Pastikan Semua Data
Sudah Terisi")
            Exit Sub
            Else
                A = TextBox12.Text
                b = TextBox10.Text
                TextBox9.Text = A +
b
                Call Koneksi()
                Dim edit As String =
"Update DPL set Saldo =' " &
TextBox9.Text & "' where ID =' "
& TextBox19.Text & "' "
                CMD = New
OleDbCommand(edit, CONN)
                CMD.ExecuteNonQuery()
                Dim edita As String
= "Update CP set Saldo =' " &
TextBox9.Text & "' where ID =' "
& TextBox19.Text & "' "
                CMD = New
OleDbCommand(edita, CONN)
                CMD.ExecuteNonQuery()
                Call Koneksi()
                Button11.BackColor =
Color.Red
                MsgBox("Data
berhasil di edit",
MsgBoxStyle.Information,
"Information")
                Call kondisiawal()
                Call kosongkan()
            End If
        End Sub

```

```

False      Button15.Enabled =                TextBox14.Text = ""
False      Button24.Enabled =                TextBox13.Text = ""
False      Button11.Enabled =                End If
True       Button11.BackColor =              End Sub
Color.Red  Button11.BackColor =              Private Sub
          End If                             Button20_Click(ByVal sender As
          End Sub                             Object, ByVal e As EventArgs)
          Private Sub                         Handles Button20.Click
Button18_Click(ByVal sender As               If TextBox18.Text = ""
Object, ByVal e As EventArgs)               Or TextBox17.Text = "" Then
Handles Button18.Click                       MsgBox("Data Belum
          If TextBox13.Text = ""              Lengkap, Pastikan Semua Data
Or TextBox14.Text = "" Or                    Sudah Terisi")
TextBox15.Text = "" Then                     Else
          MsgBox("Data Belum                  Call Koneksi()
Lengkap, Pastikan Semua Data                  Dim simpan As String
Sudah Terisi")                               = "insert into login values ('"
          Exit Sub                             & TextBox18.Text & "','" &
Else                                           TextBox17.Text & ');"
          Call Koneksi()                       CMD = New
          Dim edit As String =                 OleDbCommand(simpan, CONN)
"Update HRG set Harga =" &
TextBox13.Text & "' where Kelas
=" & Label21.Text & "'"
          CMD = New
OleDbCommand(edit, CONN)

CMD.ExecuteNonQuery()
          Dim edita As String
= "Update HRG set Harga =" &
TextBox14.Text & "' where Kelas
=" & Label22.Text & "'"
          CMD = New
OleDbCommand(edita, CONN)

CMD.ExecuteNonQuery()
          Dim editb As String
= "Update HRG set Harga =" &
TextBox15.Text & "' where Kelas
=" & Label23.Text & "'"
          CMD = New
OleDbCommand(editb, CONN)

CMD.ExecuteNonQuery()
          MsgBox("Data
berhasil di edit",
MsgBoxStyle.Information,
"Information")
          Call kondisiawal3()
Button17.Enabled =
True
Button18.Enabled =
False
          TextBox15.Text = ""

```

```

          TextBox14.Text = ""
          TextBox13.Text = ""
          End If
          End Sub
          Private Sub
Button20_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button20.Click
          If TextBox18.Text = ""
Or TextBox17.Text = "" Then
          MsgBox("Data Belum
Lengkap, Pastikan Semua Data
Sudah Terisi")
          Else
          Call Koneksi()
          Dim simpan As String
= "insert into login values ('"
& TextBox18.Text & "','" &
TextBox17.Text & ');"
          CMD = New
OleDbCommand(simpan, CONN)

CMD.ExecuteNonQuery()
          Call Koneksi()
          MsgBox("Data
berhasil di Input",
MsgBoxStyle.Information,
"Information")
          Call kondisiawal2()
          TextBox18.Text = ""
          TextBox17.Text = ""
          End If
          End Sub
          Private Sub
Button21_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button21.Click
          If TextBox18.Text = ""
Or TextBox17.Text = "" Then
          MsgBox("Data Belum
Lengkap, Pastikan Semua Data
Sudah Terisi")
          Else
          Call Koneksi()
          Dim edit As String =
"update login set password =" &
TextBox17.Text & "' where user
=" & TextBox18.Text & ');"
          CMD = New
OleDbCommand(edit, CONN)

CMD.ExecuteNonQuery()
          Call Koneksi()
          MsgBox("Data
berhasil di Input",

```

```

MsgBoxStyle.Information,
"Information")
    Call kondisiawal2()
    TextBox18.Text = ""
    TextBox17.Text = ""
End If
End Sub

Private Sub
Button19_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button19.Click
    If
MessageBox.Show("Yakin akan
dihapus..?", "",
MessageBoxButtons.YesNo) =
Windows.Forms.DialogResult.Yes
Then
    Dim hapus As String
= "delete * from login where
user='" & TextBox18.Text & "'"
    CMD = New
OleDbCommand(hapus, CONN)

CMD.ExecuteNonQuery()
    MsgBox("Data
berhasil di Hapus",
MsgBoxStyle.Information,
"Information")
    Call kosongkan()
    Call kondisiawal2()
    Else
    Call kosongkan()
    End If
End Sub
Sub PortAvailable()
    For Each PortName As
String In
My.Computer.Ports.SerialPortName
s
cbPort.Items.Add(PortName)
    Next
End Sub
Private Sub
btStart_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles btStart.Click
    If btStart.Text =
"Start" Then
    If
cbPort.SelectedItem = Nothing Or
cbBoudRate.Text = Nothing Then

MessageBox.Show("Port/Boudrate
belum dipilih, silahkan pilih
sebelum START")
Exit Sub
End If
Try
DoConnect()
cbPort.Enabled =
False

cbBoudRate.Enabled = False
btStart.Text =
"Stop"

btStart.BackColor = Color.Red
Catch ex As
Exception
    MsgBox("Gagal
dalam konfigurasi Serial")
End Try
ElseIf btStart.Text =
"Stop" Then
    Try
DoDisconnect()
cbPort.Enabled =
True

cbBoudRate.Enabled = True
btStart.Text =
"Start"

btStart.BackColor =
Color.Transparent
    Call k()
    Catch ex As
Exception
    MsgBox("Gagal
dalam menutup Serial")
End Try
End If
End Sub

Private Sub
Button22_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
Button22.Click
    If btStart.Text =
"Start" Then

MessageBox.Show("klik tombol
START pada Koneksi")
Exit Sub
ElseIf
Button22.BackColor = Color.Red
Then
    MsgBox("Tempelkan
Kartu Pada Reader")
    Try
Timer1.Start()

```



```

        Timer2.Start()
Button22.BackColor = Color.Green
        Button22.Enabled
= False
        Catch ex As
Exception
        End Try
        End If
    End Sub

    Private Sub
Button11_Click(ByVal sender As
Object, ByVal e As EventArgs)
Handles Button11.Click
        If btStart.Text =
"Start" Then

MessageBox.Show("klik tombol
START pada Koneksi")
        Exit Sub
        ElseIf
Button11.BackColor = Color.Red
Then
            MsgBox("Tempelkan
Kartu Pada Reader")

        Try
            Timer1.Start()
            Timer2.Start()

TextBox12.Enabled = True

Button11.BackColor = Color.Green
        Button11.Enabled
= False
            Button15.Enabled
= True
            Button24.Enabled
= True

        Catch ex As
Exception
        End Try
        End If
    End Sub

    Sub Timer1_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
Timer1.Tick
        'Call ReadDataArduino()

    End Sub

    Private Sub
Timer2_Tick(ByVal sender As
System.Object, ByVal e As

```

```

System.EventArgs) Handles
Timer2.Tick
        If Button9.BackColor =
Color.Green Then
            TextBox1.Text =
lbinfo.Text
            lbinfo.Text = ""
            'Timer2.Stop()
        ElseIf
Button22.BackColor = Color.Green
Then
            TextBox8.Text =
lbinfo.Text
            lbinfo.Text = ""
            'Timer2.Stop()
        ElseIf
Button11.BackColor = Color.Green
Then
            TextBox19.Text =
lbinfo.Text
            lbinfo.Text = ""
            'Timer2.Stop()
        End If

    End Sub

    Private Sub
Button17_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
Button17.Click
        CMD = New
OleDbCommand("select *from HRG
where Kelas =' " & Label21.Text &
"', CONN)
        RD = CMD.ExecuteReader()
        RD.Read()
        TextBox13.Text =
RD.Item("Harga")
        TextBox13.Focus()
        CMD = New
OleDbCommand("select *from HRG
where Kelas =' " & Label22.Text &
"', CONN)
        RD = CMD.ExecuteReader()
        RD.Read()
        TextBox14.Text =
RD.Item("Harga")
        TextBox14.Focus()
        CMD = New
OleDbCommand("select *from HRG
where Kelas =' " & Label23.Text &
"', CONN)
        RD = CMD.ExecuteReader()
        RD.Read()
        TextBox15.Text =
RD.Item("Harga")

```

```

        TextBox15.Focus()
        Button17.Enabled = False
        Button18.Enabled = True
    End Sub

    Private Sub
    Button12_Click(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs) Handles
    Button12.Click
        TextBox8.Text = ""
        ComboBox2.Text = "Kelas"
        Button14.Enabled = False
        Button13.Enabled = False
        Button12.Enabled = False
        Button22.BackColor =
    Color.Red
        Button22.Enabled = True
    End Sub

    Private Sub
    Button24_Click(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs) Handles
    Button24.Click
        TextBox12.Text = ""
        TextBox19.Text = ""
        GroupBox2.Enabled =
    False
        TextBox12.Enabled =
    False
        Button15.Enabled = False
        Button11.Enabled = True
        Button24.Enabled = False
        Button11.BackColor =
    Color.Red
    End Sub

    Private Sub
    Button25_Click(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs) Handles
    Button25.Click
        End
    End Sub

    Private Sub
    Button2_Click(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs)
        Koneksi()
        DA = New
    OleDbDataAdapter("select * from
    VIP", CONN)
        DS = New DataSet
        DS.Clear()
        DA.Fill(DS, "VIP")
        DataGridView1.DataSource
        = (DS.Tables("VIP"))
    End Sub

    Private Sub
    Button3_Click(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs)
        Koneksi()
        DA = New
    OleDbDataAdapter("select * from
    Tribun", CONN)
        DS = New DataSet
        DS.Clear()
        DA.Fill(DS, "Tribun")
        DataGridView1.DataSource
        = (DS.Tables("Tribun"))
    End Sub

    Private Sub
    Button1_Click(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs) Handles
    Button1.Click
        Form1.Show()
    End Sub

    Private Sub
    Button23_Click(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs)
        SerialPort1.Write("1seddfn")
    End Sub

    Private Sub
    Button26_Click_1(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs) Handles
    Button26.Click
        SerialPort1.Write("A")
    End Sub

    Private Sub
    Button23_Click_1(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs) Handles
    Button23.Click
        SerialPort1.Write("1seddfn")
    End Sub

    Private Sub
    Button2_Click_1(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs) Handles
    Button2.Click

```

```

        Koneksi()
        DA = New
OleDbDataAdapter("select * from
Daftar_Pengunjung", CONN)
        DS = New DataSet
        DS.Clear()
        DA.Fill(DS,
"Daftar_Pengunjung")

DataGridView1.DataSource =
(DS.Tables("Daftar_Pengunjung"))
    End Sub

    Private Sub
CheckBox1_CheckedChanged(ByVal
sender As System.Object, ByVal e
As System.EventArgs) Handles
CheckBox1.CheckedChanged
        If CheckBox1.Checked =
True Then ComboBox3.Enabled =
False
    End Sub

    Private Sub
TabPage7_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
TabPage7.Click

    End Sub

    Private Sub
Label19_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
Label19.Click
        Button23.Visible = True
        Button26.Visible = True
    End Sub
End Class

Imports System.Data.OleDb
Public Class Form1
    Dim comOpen1 As Boolean
    'Keeps track of the port status.
    True = Open; False = Closed
    Dim comOpen2 As Boolean
    'Keeps track of the port status.
    True = Open; False = Closed
    Dim comOpen3 As Boolean
    'Keeps track of the port status.
    True = Open; False = Closed
    Dim readbuffer1 As String
    'Buffer of whats read from the
    serial port

        Dim readbuffer2 As String
    'Buffer of whats read from the
    serial port
        Dim readbuffer3 As String
    'Buffer of whats read from the
    serial port
        Dim CONN As OleDbConnection
        Dim CMD As OleDbCommand
        Dim DS As New DataSet
        Dim DA As OleDbDataAdapter
        Dim RD As OleDbDataReader
        Dim coba As String
        Dim LokasiDB As String
        Dim c As Single
        Dim c1 As Single
        Dim c2 As Single
        Dim d As Single
        Dim e As Single
        Dim TK As String
        Dim Tb As String
        Dim Tc As String
        Dim cc As Single
        Dim dd As Single
        Dim bb As Single
        Delegate Sub
myMethodDelegate(ByVal [text] As
String)
        Private Delegate Sub
delegateReadData()
        Sub Koneksi()
            LokasiDB = "Provider =
Microsoft.ACE.OLEDB.12.0;Data
Source = Tiketing.accdb"
            CONN = New
OleDbConnection(LokasiDB)
            If CONN.State =
ConnectionState.Closed Then
CONN.Open()
        End Sub
        Sub kondisiawal()
            Call Koneksi()
            DA = New
OleDbDataAdapter("select * from
CP", CONN)
            DS = New DataSet
            DS.Clear()
            DA.Fill(DS, "CP")
            DataGridView1.DataSource
= (DS.Tables("CP"))
        End Sub

        Private Sub form1_Load(ByVal
sender As Object, ByVal e As
EventArgs) Handles MyBase.Load
            'Get all connected
            serial ports

```

```

        Dim comPorts As String()
    =
    System.IO.Ports.SerialPort.GetPortNames
        If comPorts.Count < 1
    Then
        'If there are not
    ports connected, show an error
    and close the program.
        MsgBox("There are no
    com ports available! Closing
    program.")
        Me.Close()
        Else

    cmbPorts1.Items.AddRange(comPorts)
        cmbPorts1.Text =
    comPorts(0)

    cmbPorts2.Items.AddRange(comPorts)
        cmbPorts2.Text =
    comPorts(0)

    cmbPorts3.Items.AddRange(comPorts)
        cmbPorts3.Text =
    comPorts(0)
        End If
        Call kondisiawal()

        CMD = New
    OleDbCommand("select *from HRG
    Where Kelas=' & "Belakang
    Gawang" & "'", CONN)
        RD = CMD.ExecuteReader()
        RD.Read()
        d = RD.Item("Harga")
        CMD = New
    OleDbCommand("select *from HRG
    Where Kelas=' & "Tribun" & "'",
    CONN)
        RD = CMD.ExecuteReader()
        RD.Read()
        e = RD.Item("Harga")
        End Sub

        Private Sub
    btnConnect1_Click(ByVal sender
    As Object, ByVal e As EventArgs)
    Handles btnConnect1.Click
        If cmbPorts1.Text =
    cmbPorts2.Text Or cmbPorts2.Text
    = cmbPorts3.Text Or
    cmbPorts1.Text = cmbPorts3.Text
    Then

```

```

    MessageBox.Show("Pastikan port
    sudah dipilih dengan benar")
        Exit Sub
        End If
        'Conenct to serial port
    DoConnect()
    Timer1.Start()
        End Sub

        Private Sub
    frmDisplay_FormClosing(ByVal
    sender As Object, ByVal e As
    FormClosingEventArgs) Handles
    Me.FormClosing
        'Gracefully disconnect
    before form closes
        DoDisconnect()
        SerialPort1.Write("A")
        SerialPort2.Write("A")
        SerialPort3.Write("A")
        End Sub
        Private Sub
    btnDisconnect1_Click(ByVal
    sender As Object, ByVal e As
    EventArgs) Handles
    btnDisconnect1.Click
        'Disconnect the serial
    port
        DoDisconnect()
        End Sub

        Private Sub
    SerialPort1_DataReceived(ByVal
    sender As System.Object, _
    ByVal e As
    System.IO.Ports.SerialDataReceiv
    edEventArgs) _
    Handles SerialPort1.DataReceived
        If comOpen1 Then

            Try
                'Send data to a
            new thread to update the
            temperature display
                readbuffer1 =
            SerialPort1.ReadLine()
                Me.Invoke(New
            EventHandler(AddressOf update1))
            Catch ex As
            Exception
                'Otherwise show
            error. Will display when
            disconnecting.

```

```

'MsgBox(ex.Message)
    End Try
    End If
    End Sub
    Private Sub
SerialPort2_DataReceived(ByVal
sender As System.Object, _

ByVal e As
System.IO.Ports.SerialDataReceiv
edEventArgs) _

Handles SerialPort2.DataReceived
    If comOpen2 Then

        Try
            'Send data to a
            new thread to update the
            temperature display
            readbuffer2 =
SerialPort2.ReadLine()
            Me.Invoke(New
            EventHandler(AddressOf update2))
            Catch ex As
            Exception
                'Otherwise show
                error. Will display when
                disconnecting.

'MsgBox(ex.Message)
    End Try
    End If
    End Sub
    Private Sub
SerialPort3_DataReceived(ByVal
sender As System.Object, _

ByVal e As
System.IO.Ports.SerialDataReceiv
edEventArgs) _

Handles SerialPort3.DataReceived
    If comOpen3 Then

        Try
            'Send data to a
            new thread to update the
            temperature display
            readbuffer3 =
SerialPort3.ReadLine()
            Me.Invoke(New
            EventHandler(AddressOf update3))
            Catch ex As
            Exception

                'Otherwise show
                error. Will display when
                disconnecting.

'MsgBox(ex.Message)
    End Try
    End If
    End Sub
    Public Sub update1(ByVal
sender As Object, ByVal e As
System.EventArgs)
        'Update temperature
        display as it comes in
        Dim read1 As String
        read1 =
readbuffer1.Replace(vbCr,
        "").Replace(vbLf, "")
        TextBox1.Text = read1
    End Sub
    Public Sub update2(ByVal
sender As Object, ByVal e As
System.EventArgs)
        'Update temperature
        display as it comes in
        Dim read2 As String
        read2 =
readbuffer2.Replace(vbCr,
        "").Replace(vbLf, "")
        TextBox2.Text = read2
    End Sub

    Public Sub update3(ByVal
sender As Object, ByVal e As
System.EventArgs)
        'Update temperature
        display as it comes in
        Dim read3 As String
        read3 =
readbuffer3.Replace(vbCr,
        "").Replace(vbLf, "")
        TextBox3.Text = read3
    End Sub
    Public Sub DoDisconnect()
        SerialPort1.Write("A")
        SerialPort2.Write("A")
        SerialPort3.Write("A")
        'Graceful disconnect if
        port is open
        If comOpen1 Then

SerialPort1.DiscardInBuffer()
        SerialPort1.Close()

        'Reset our flag and
        controls
        comOpen1 = False

```

```

btnDisconnect1.Enabled = False
    btnConnect1.Enabled
= True
    txtBaudRate1.Enabled
= True
    cmbPorts1.Enabled =
True
    End If
    'Graceful disconnect if
port is open
    If comOpen2 Then

SerialPort2.DiscardInBuffer()
    SerialPort2.Close()

    'Reset our flag and
controls
    comOpen2 = False
    txtBaudRate2.Enabled
= True
    cmbPorts2.Enabled =
True
    End If
    'Graceful disconnect if
port is open
    If comOpen3 Then

SerialPort3.DiscardInBuffer()
    SerialPort3.Close()

    'Reset our flag and
controls
    comOpen3 = False
    txtBaudRate3.Enabled
= True
    cmbPorts3.Enabled =
True
    End If
End Sub

Public Sub DoConnect()

    'Setup the serial port
connection
    With SerialPort1
        .PortName =
cmbPorts1.Text
        'Selected Port
        .BaudRate =
CInt(txtBaudRate1.Text)
        'Baud Rate. 9600 is default.
        .Parity =
IO.Ports.Parity.None
        .DataBits = 8
        .StopBits =
IO.Ports.StopBits.One
        .Handshake =
IO.Ports.Handshake.None
        .RtsEnable = False

        .ReceivedBytesThreshold = 1
        .NewLine = vbCr
        .ReadTimeout = 10000
    End With

    'Try to open the
selected port...
    Try
        SerialPort1.Open()
        comOpen1 =
SerialPort1.IsOpen
        SerialPort2.Open()
        .Handshake =
IO.Ports.Handshake.None
        .RtsEnable = False

        .ReceivedBytesThreshold = 1
        .NewLine = vbCr
        .ReadTimeout = 10000
    End With

    With SerialPort2
        .PortName =
cmbPorts2.Text
        'Selected Port
        .BaudRate =
CInt(txtBaudRate2.Text)
        'Baud Rate. 9600 is default.
        .Parity =
IO.Ports.Parity.None
        .DataBits = 8
        .StopBits =
IO.Ports.StopBits.One
        .Handshake =
IO.Ports.Handshake.None
        .RtsEnable = False

        .ReceivedBytesThreshold = 1
        .NewLine = vbCr
        .ReadTimeout = 10000
    End With

    With SerialPort3
        .PortName =
cmbPorts3.Text
        'Selected Port
        .BaudRate =
CInt(txtBaudRate3.Text)
        'Baud Rate. 9600 is default.
        .Parity =
IO.Ports.Parity.None
        .DataBits = 8
        .StopBits =
IO.Ports.StopBits.One
        .Handshake =
IO.Ports.Handshake.None
        .RtsEnable = False

        .ReceivedBytesThreshold = 1
        .NewLine = vbCr
        .ReadTimeout = 10000
    End With

```

```

        comOpen2 =
SerialPort1.IsOpen
        SerialPort3.Open()
        comOpen3 =
SerialPort1.IsOpen
SerialPort1.Write("A")
SerialPort2.Write("A")
SerialPort3.Write("A")
    Catch ex As Exception
        comOpen1 = False
        comOpen2 = False
        comOpen3 = False
        MsgBox("Error Open:
" & ex.Message)
    End Try

    btnDisconnect1.Enabled =
True
    btnConnect1.Enabled =
False
    txtBaudRate1.Enabled =
False
    cmbPorts1.Enabled =
False
    txtBaudRate2.Enabled =
False
    cmbPorts2.Enabled =
False
    txtBaudRate3.Enabled =
False
    cmbPorts3.Enabled =
False
    TextBox1.Enabled = False
    TextBox2.Enabled = False
    TextBox3.Enabled = False

    End Sub
    Private Sub
TextBox1_TextChanged(ByVal
sender As System.Object, ByVal e
As System.EventArgs) Handles
TextBox1.TextChanged
        Dim l As Single
        Dim cc As Single
        CMD = New
OleDbCommand("select *from HRG
Where Kelas='" & "VIP" & "'",
CONN)
        RD = CMD.ExecuteReader()
        RD.Read()
        c = RD.Item("Harga")
        If TextBox1.Text =
Chr(12) Then Call Koneksi()

        CMD = New
OleDbCommand("select *from CP
Where ID ='" & TextBox1.Text &
"'", CONN)
        RD = CMD.ExecuteReader()
        RD.Read()
        Gate1.BackColor =
Color.Transparent
        Timer1.Start()
        If Not RD.HasRows Then
            Label1.Text = "tidak
terdaftar"
            Label1.Focus()
            Gate1.BackColor =
Color.Red
        Else
            Label1.Text =
RD.Item("Saldo")
            Label1.Focus()
            If Label1.Text =
"Tribun" Then
                Gate1.BackColor
= Color.Red
            ElseIf Label1.Text =
"Belakang Gawang" Then
                Gate1.BackColor
= Color.Red
            ElseIf Label1.Text =
"VIP" Then
                Gate1.BackColor
= Color.Green
            TK = "HARIAN"
            Else : l =
Label1.Text
                Call Koneksi()
                CMD = New
OleDbCommand("select *from CP
Where ID ='" & TextBox1.Text &
"'", CONN)
                RD =
CMD.ExecuteReader()
                RD.Read()
                cc =
RD.Item("max_taping")
                Call
kondisiawal()
                If l >= c And cc
>= 1 Then
                    Call
Koneksi()
                    Dim edit As
String = "Update CP set
max_taping ='" & cc - 1 & "'",
Saldo = '" & l - c & "' where ID
= '" & TextBox1.Text & "'"

```

```

        CMD = New
OleDbCommand(edit, CONN)
CMD.ExecuteNonQuery()
        Call
Koneksi()
Gate1.BackColor = Color.Green
        Label1.Text
= 1 - c
        TK =
"LANGGANAN"
        ElseIf 1 < c Or
cc <= 0 Then
Gate1.BackColor = Color.Red
        Label1.Text
= 1
        End If
        End If
    End Sub

    Private Sub
TextBox2_TextChanged(ByVal
sender As System.Object, ByVal e
As System.EventArgs) Handles
TextBox2.TextChanged
        Dim k As Single
        Dim dd As Single
        CMD = New
OleDbCommand("select *from HRG
Where Kelas='" & "Belakang
Gawang" & "'", CONN)
        RD = CMD.ExecuteReader()
        RD.Read()
        c1 = RD.Item("Harga")
        If TextBox2.Text =
Chr(12) Then Call Koneksi()
        CMD = New
OleDbCommand("select *from CP
Where ID='" & TextBox2.Text &
'", CONN)
        RD = CMD.ExecuteReader()
        Timer2.Start()
        RD.Read()
        If Not RD.HasRows Then
            Label2.Text = "tidak
terdaftar"
            Label2.Focus()
            Gate2.BackColor =
Color.Red
        Else
            Label2.Text =
RD.Item("Saldo")
            Label2.Focus()

                If Label2.Text =
"Tribun" Then
                    Gate2.BackColor
= Color.Red

                ElseIf Label2.Text =
"Belakang Gawang" Then
                    Gate2.BackColor
= Color.Green
                    Tb = "HARIAN"
                ElseIf Label2.Text =
"VIP" Then
                    Gate2.BackColor
= Color.Red
                Else : k =
Label2.Text
                    CMD = New
OleDbCommand("select *from CP
Where ID='" & TextBox2.Text &
'", CONN)
                    RD =
CMD.ExecuteReader()
                    RD.Read()
                    dd =
RD.Item("max_taping")
                    Call
kondisiawal()
                    If k >= c1 And
dd >= 1 Then
                        Call
Koneksi()
                        Dim edit As
String = "Update CP set
max_taping =' " & dd - 1 & "'
where ID =' " & TextBox2.Text &
'"
                        CMD = New
OleDbCommand(edit, CONN)
                        CMD.ExecuteNonQuery()
                        Call
Koneksi()
                        Gate2.BackColor = Color.Green
                        Label2.Text
= k - c1
                        Tb =
"LANGGANAN"
                    ElseIf k < c1 Or
dd < 1 Then
                        Gate2.BackColor = Color.Red
                        Label2.Text
= k
                    End If
                End If
            End If
        End Sub

```



```

        End If
    End Sub

    Private Sub
    TextBox3_TextChanged(ByVal
    sender As System.Object, ByVal e
    As System.EventArgs) Handles
    TextBox3.TextChanged
        Dim m As Single
        Dim bb As Single
        CMD = New
        OleDbCommand("select *from HRG
        Where Kelas='" & "Tribun" & "'",
        CONN)
        RD = CMD.ExecuteReader()
        RD.Read()
        c2 = RD.Item("Harga")
        If TextBox3.Text =
        Chr(12) Then Call Koneksi()
        CMD = New
        OleDbCommand("select *from CP
        Where ID='" & TextBox3.Text &
        "'", CONN)
        RD = CMD.ExecuteReader()
        RD.Read()
        Timer3.Start()
        If Not RD.HasRows Then
            Label3.Text = "tidak
            terdaftar"
            Label3.Focus()
            Gate3.BackColor =
            Color.Red
        Else
            Label3.Text =
            RD.Item("Saldo")
            Label3.Focus()
            If Label3.Text =
            "Tribun" Then
                Gate3.BackColor
                = Color.Green
                Tc = "HARIAN"
            ElseIf Label3.Text =
            "Belakang Gawang" Then
                Gate3.BackColor
                = Color.Red
            ElseIf Label3.Text =
            "VIP" Then
                Gate3.BackColor
                = Color.Red
            Else : m =
            Label3.Text
            CMD = New
            OleDbCommand("select *from CP
            Where ID='" & TextBox3.Text &
            "'", CONN)
            RD =
            CMD.ExecuteReader()
            RD.Read()
            RD.Read()
            bb =
            RD.Item("max_taping")
            Call
            kondisiawal()
            If m >= c2 And
            bb > 0 Then
                Call
                Koneksi()
                Dim edit As
                String = "Update CP set
                max_taping='" & bb - 1 & "'
                where ID='" & TextBox3.Text &
                "'"
                CMD = New
                OleDbCommand(edit, CONN)
                CMD.ExecuteNonQuery()
                Call
                Koneksi()
                Gate3.BackColor = Color.Green
                Label3.Text
                = m - c2
                Tc =
                "LANGGANAN"
            ElseIf m < c2 Or
            bb <= 0 Then
                Gate3.BackColor = Color.Red
                Label3.Text
                = m
            End If
        End If
    End Sub

    Private Sub
    Timer1_Tick(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs) Handles
    Timer1.Tick
        Dim waktu As String
        Dim g1 As Single
        waktu = DateString
        If Gate1.BackColor =
        Color.Green Then
            CMD = New
            OleDbCommand("select *from
            Daftar_Pengunjung Where ID='" &
            TextBox1.Text & "'", CONN)
            RD =
            CMD.ExecuteReader()
            RD.Read()

```

```

        If Not RD.HasRows
Then
            g1 = 0
        Else
            g1 =
RD.Item("jumlah_taping")
            Call Koneksi()
        End If
        If TK = "HARIAN"
Then
            Dim hapusa As
String = "delete * from CP where
ID='" & TextBox1.Text & "'"
            CMD = New
OleDbCommand(hapusa, CONN)

CMD.ExecuteNonQuery()
            Call
kondisiawal()
        End If
        If TK = "LANGGANAN"
And g1 > 0 Then
            Call Koneksi()
            Dim editd As
String = "Update DPL set Saldo
='" & Label1.Text & "' where ID
='" & TextBox1.Text & "'"
            CMD = New
OleDbCommand(editd, CONN)

CMD.ExecuteNonQuery()
            Call Koneksi()
            Dim editc As
String = "Update CP set
max_taping =' & cc - 1 &
'",Saldo =' & Label1.Text & "'
where ID =' & TextBox1.Text &
'"
            CMD = New
OleDbCommand(editc, CONN)

CMD.ExecuteNonQuery()
            Call Koneksi()
            Dim edit As
String = "Update
Daftar_Pengunjung set
jumlah_taping =' & g1 + 1 & "'
where ID =' & TextBox1.Text &
'"
            CMD = New
OleDbCommand(edit, CONN)

CMD.ExecuteNonQuery()
            Call Koneksi()
            Dim edita As
String = "Update VIP set
jumlah_taping =' & g1 + 1 & "'

```

```

where ID =' & TextBox1.Text &
'"
            CMD = New
OleDbCommand(edita, CONN)

CMD.ExecuteNonQuery()
            Call Koneksi()
        End If
        If g1 = 0 Then
            Dim simpana As
String = "insert into
Daftar_Pengunjung
(ID,Kelas,Tipe_Kartu,Waktu,jumla
h_taping) values (' &
TextBox1.Text & ',' & "VIP" &
',' & TK & ',' & waktu &
',' & "1" & ');"
            CMD = New
OleDbCommand(simpana, CONN)

CMD.ExecuteNonQuery()
            Call Koneksi()
            Dim simpan As
String = "insert into VIP
(ID,Tipe_Kartu,waktu) values ('
& TextBox1.Text & ',' & TK &
',' & waktu & ');"
            CMD = New
OleDbCommand(simpan, CONN)

CMD.ExecuteNonQuery()
            Call Koneksi()
        End If

SerialPort1.Write("1" +
Label1.Text)

TimeSpan.FromSeconds(5)
        TextBox1.Text = ""
        Label1.Text = ""
        Gate1.BackColor =
Color.Transparent
        Timer1.Stop()

        ElseIf Gate1.BackColor =
Color.Red Then
            If comOpen1 = False
Then SerialPort1.Open()

SerialPort1.Write("0" +
Label1.Text)

TimeSpan.FromSeconds(5)
        TextBox1.Clear()
        Label1.Text = ""
        Gate1.BackColor =
Color.Transparent

```

```

        Timer1.Stop()
    End If

End Sub

Private Sub
Button1_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
Button1.Click
    Me.Hide()
End Sub

Private Sub
Timer2_Tick(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
Timer2.Tick
    Dim waktu As String
    Dim g2 As Single
    waktu = DateString
    If Gate2.BackColor =
Color.Green Then
        CMD = New
OleDbCommand("select *from
Daftar_Pengunjung Where ID='" &
TextBox2.Text & "'", CONN)
        RD =
CMD.ExecuteReader()
        RD.Read()
        If Not RD.HasRows
Then
            g2 = 0
        Else
            g2 =
RD.Item("jumlah_taping")
            Call Koneksi()
        End If
        If Tb = "HARIAN"
Then
            Dim hapusa As
String = "delete * from CP where
ID='" & TextBox2.Text & "'"
            CMD = New
OleDbCommand(hapusa, CONN)

CMD.ExecuteNonQuery()
            Call
kondisiawal()
        End If
        If Tb = "LANGGANAN"
And g2 > 0 Then
            Call Koneksi()
            Dim editd As
String = "Update DPL set Saldo
= '" & Label2.Text & "' where ID
= '" & TextBox2.Text & "'"
            CMD = New
OleDbCommand(editd, CONN)

CMD.ExecuteNonQuery()
            Call Koneksi()
            Dim editc As
String = "Update CP set
max_taping = '" & dd - 1 & "'
'",Saldo = '" & Label2.Text & "'
where ID = '" & TextBox2.Text &
"'"
            CMD = New
OleDbCommand(editc, CONN)

CMD.ExecuteNonQuery()
            Call Koneksi()
            Dim edit As
String = "Update
Daftar_Pengunjung set
jumlah_taping = '" & g2 + 1 & "'
where ID = '" & TextBox2.Text &
"'"
            CMD = New
OleDbCommand(edit, CONN)

CMD.ExecuteNonQuery()
            Call Koneksi()
            Dim edita As
String = "Update Blk_Gawang set
jumlah_taping = '" & g2 + 1 & "'
where ID = '" & TextBox2.Text &
"'"
            CMD = New
OleDbCommand(edita, CONN)

CMD.ExecuteNonQuery()
            Call Koneksi()

SerialPort2.Write("1" +
Label2.Text)

TimeSpan.FromSeconds(5)
            TextBox2.Text =
""
            Label2.Text = ""
            Gate2.BackColor
= Color.Transparent
            Timer2.Stop()
        End If
        If g2 = 0 Then
            Dim simpana As
String = "insert into
Daftar_Pengunjung
(ID,Kelas,Tipe_Kartu,Waktu,jumla
h_taping) values ('" &

```

```

TextBox2.Text & "',' &
"Belakang Gawang" & "',' & Tb &
 "',' & waktu & "',' & "1" &
 "'');";
        CMD = New
OleDbCommand(simpana, CONN)

CMD.ExecuteNonQuery()
        Call Koneksi()
        Dim simpan As
String = "insert into Blk_Gawang
(ID,Tipe_Kartu,waktu) values ('"
& TextBox2.Text & "',' & Tb &
 "',' & waktu & "')";"
        CMD = New
OleDbCommand(simpan, CONN)

CMD.ExecuteNonQuery()
        Call Koneksi()

SerialPort2.Write("1" +
Label2.Text)

TimeSpan.FromSeconds(5)
        TextBox2.Text =
""
        Label2.Text = ""
        Gate2.BackColor
= Color.Transparent
        Timer2.Stop()
        End If
        ElseIf Gate2.BackColor =
Color.Red Then
            If comOpen2 = False
Then SerialPort2.Open()

SerialPort2.Write("0" +
Label2.Text)

TimeSpan.FromSeconds(5)
        TextBox2.Text = ""
        Label2.Text = ""
        Gate2.BackColor =
Color.Transparent
        Timer2.Stop()
        End If
    End Sub

    Private Sub
Timer3_Tick(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
Timer3.Tick
        Dim waktu As String
        Dim g2 As Single
        waktu = DateString

        If Gate3.BackColor =
Color.Green Then
            CMD = New
OleDbCommand("select *from
Daftar_Pengunjung Where ID='" &
TextBox3.Text & "'", CONN)
            RD =
CMD.ExecuteReader()
            RD.Read()
            If Not RD.HasRows
Then
                g2 = 0
            Else
                g2 =
RD.Item("jumlah_taping")
                Call Koneksi()
                End If
                If Tc = "HARIAN"
Then
                    Dim hapusa As
String = "delete * from CP where
ID='" & TextBox3.Text & "'"
                    CMD = New
OleDbCommand(hapusa, CONN)

CMD.ExecuteNonQuery()
                    Call
kondisiawal()
                    End If
                    If Tc = "LANGGANAN"
And g2 > 0 Then
                        Call Koneksi()
                        Dim editd As
String = "Update DPL set Saldo
='" & Label3.Text & "' where ID
='" & TextBox3.Text & "'"
                        CMD = New
OleDbCommand(editd, CONN)

CMD.ExecuteNonQuery()
                            Call Koneksi()
                            Dim editc As
String = "Update CP set
max_taping =' & bb - 1 &
"',Saldo =' & Label3.Text & "'
where ID =' & TextBox1.Text &
'"
                            CMD = New
OleDbCommand(editc, CONN)

CMD.ExecuteNonQuery()
                                Call Koneksi()
                                Dim edit As
String = "Update
Daftar_Pengunjung set
jumlah_taping =' & g2 + 1 & '"

```

```

where ID =' ' & TextBox3.Text &
""
        CMD = New
OleDbCommand(edit, CONN)

CMD.ExecuteNonQuery()
        Call Koneksi()
        Dim edita As
String = "Update Tribun set
jumlah_taping =' ' & g2 + 1 & ' '
where ID =' ' & TextBox3.Text &
""
        CMD = New
OleDbCommand(edita, CONN)

CMD.ExecuteNonQuery()
        Call Koneksi()
        End If
        If g2 = 0 Then
        Dim simpana As
String = "insert into
Daftar_Pengunjung
(ID,Kelas,Tipe_Kartu,Waktu,jumla
h_taping) values (' ' &
TextBox3.Text & ', ' & "Tribun"
& ', ' & Tc & ', ' & waktu &
', ' & "1" & ');"
        CMD = New
OleDbCommand(simpana, CONN)

CMD.ExecuteNonQuery()
        Call Koneksi()
        Dim simpan As
String = "insert into Tribun
(ID,Tipe_Kartu,waktu) values (' '
& TextBox3.Text & ', ' & Tc &
', ' & waktu & ');"
        CMD = New
OleDbCommand(simpan, CONN)

CMD.ExecuteNonQuery()
        Call Koneksi()
        End If

SerialPort3.Write("1" +
Label3.Text)

TimeSpan.FromSeconds(5)
        TextBox3.Text = ""
        Label3.Text = ""
        Gate3.BackColor =
Color.Transparent
        Timer3.Stop()
        ElseIf Gate3.BackColor =
Color.Red Then
        If comOpen3 = False
Then SerialPort3.Open()

```

```

SerialPort3.Write("0" +
Label3.Text)

TimeSpan.FromSeconds(5)
        TextBox3.Text = ""
        Label3.Text = ""
        Gate3.BackColor =
Color.Transparent
        Timer3.Stop()
        End If
        End Sub
End Class

```

List program arduino

Arduino A

```

#include <SparkFunColorLCDShield.h>
#include <ColorLCDShield.h>
#include <Servo.h>
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN 9
#define SS_PIN 10
#define BACKGROUND WHITE
LCDShield lcd;
MFRC522 mfrc522(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;
boolean loopbutton = true;
boolean vb = true;
boolean ab = true;
#define button A0
const int ledpinh = A5;
const int ledpinm = A4;
int buttonstate=0;
Servo myservo;
String a;
String b;
String c;
String d;

```

```

String e;
String f;
String baca;
String bace;
char* data;
char* data1;
char* data2;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Initialize serial
communications with the PC
  while (!Serial); // Do nothing if no serial
port is opened (added for Arduinos based on
ATMEGA32U4)
  SPI.begin(); // Init SPI bus
  mfr522.PCD_Init();

  pinMode(button,INPUT_PULLUP);
  pinMode (ledpinh,OUTPUT);
  pinMode (ledpinm,OUTPUT);
  myservo.attach(3);

  lcd.init(PHILIPS);
  lcd.contrast(60);
  lcd.clear(BACKGROUND);
}

void loop() {
  // put your main code here, to run
repeatedly
  vb = true;
  digitalWrite(ledpinm , HIGH);
  loopbutton = true;
  f=Serial.readString();
  if (f.equals("A")) {
    lcd.clear(BACKGROUND);
    ab = true;
  }
  while (ab){
    lcd.clear(BACKGROUND);
    lcd.setStr("Tidak Ada", 40,30, WHITE,
RED);
    lcd.setStr("Koneksi ke PC", 57,14,
WHITE, RED);
    lcd.setRect(0,0,18,77,1,GREEN);
    lcd.setRect(0,120,18,132,1,GREEN);
    lcd.setStr("FT.UNJ", 0,77, WHITE,
GREEN);
    lcd.setStr("by:Idris", 93,2, BLACK,
BACKGROUND);
    lcd.setStr("5215111741", 107,29,
BLACK, BACKGROUND);
    myservo.write(70);
  }
  f=Serial.readString();
  if (f.equals("A")) {
    lcd.clear(BACKGROUND);
    ab = false;
  }
}

  lcd.setStr("Reader", 20,37, BLACK,
BACKGROUND);
  lcd.setStr("Siap digunakan", 37,8,
BLACK,BACKGROUND );
  lcd.setRect(0,0,18,77,1,GREEN);
  lcd.setRect(0,120,18,132,1,GREEN);
  lcd.setStr("FT.UNJ", 0,77, WHITE,
GREEN);
  lcd.setStr("by:Idris", 93,2, BLACK,
BACKGROUND);
}

```

```

    lcd.setStr("5215111741", 107,29,
BLACK, BACKGROUND);
if ( ! mfrc522.PICC_IsNewCardPresent() ) {
    return;
}

// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial() ) {
    return;
}

// Dump debug info about the card;
PICC_HaltA() is automatically called
    for (byte i = 0; i < mfrc522.uid.size; i++)
    {

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

Serial.print(mfrc522.uid.uidByte[i], HEX);
        }
// Halt PICC
Serial.println();
    a= String(mfrc522.uid.uidByte[0] < 0x10 ?
" 0" : " ");
    a+= String(mfrc522.uid.uidByte[0], HEX);
    b= String(mfrc522.uid.uidByte[1] < 0x10
? " 0" : " ");
    b+= String(mfrc522.uid.uidByte[1], HEX);
    c= String(mfrc522.uid.uidByte[2] < 0x10 ?
" 0" : " ");
    c+= String(mfrc522.uid.uidByte[2], HEX);
    d= String(mfrc522.uid.uidByte[3] < 0x10
? " 0" : " ");
    d+= String(mfrc522.uid.uidByte[3], HEX);

    e=a+b+c+d+" ";
    char *baru;
// e.toCharArray(baru,e.length()+1);
    baru= &e[0u];

    lcd.setStr(baru,55,12,BLUE,BACKGROUN
D);

while(vb){
    f=Serial.readString();
    bace=String(f[0]);
    for(byte i = 1; i < f.length();i++){
        baca+=String(f[i]); //data2=data2+data[i]
    }
// baca.toCharArray(data,baca.length()+1);
    data = &baca[0u];

    lcd.setStr(data,75,4,BLUE,BACKGROUND
);
    if (f.length()>1){
        vb = false;
    }
}
    if (bace.equals("1")) {
        lcd.setStr(data,75,4,WHITE,GREEN);
        digitalWrite(ledpinh , HIGH);
        digitalWrite(ledpinm , LOW);
        myservo.write(50);
        delay(200);
        while (loopbutton){
            buttonstate=digitalRead(button);
            if(buttonstate==LOW){
                digitalWrite(ledpinm , HIGH);
                myservo.write(70);
                delay(200);
                lcd.clear(BACKGROUND);

```

```

digitalWrite(ledpinh , LOW);

f="";
baca="";
loopbutton = false;
}
}
}
else if (data1==0) {
  lcd.setStr(data,75,4,WHITE,RED);
  myservo.write(70);
  delay(2000);
  lcd.clear(BACKGROUND);
  f="";
  baca="";
}

mfr522.PICC_HaltA();
mfr522.PCD_StopCrypto1();
}

Arduino B
#include <SparkFunColorLCDShield.h>

#include <ColorLCDShield.h>

#include <Servo.h>

#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN 9
#define SS_PIN 10
#define BACKGROUND WHITE

LCDShield lcd;
MFRC522 mfr522(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;

boolean loopbutton = true;
boolean vb = true;
boolean ab = true;
#define button A0
const int ledpinh = A5;
const int ledpinm = A4;
int buttonstate=0;
Servo myservo;

String a;
String b;
String c;
String d;
String e;
String f;
String baca;
String bace;
char* data;
char* data1;
char* data2;

void setup() {
  // put your setup code here, to run once:

  Serial.begin(9600); // Initialize serial
communications with the PC

  while (!Serial); // Do nothing if no serial
port is opened (added for Arduinos based on
ATMEGA32U4)

  SPI.begin(); // Init SPI bus
mfr522.PCD_Init();

  pinMode(button,INPUT_PULLUP);
  pinMode (ledpinh,OUTPUT);
  pinMode (ledpinm,OUTPUT);

```



```

myservo.attach(3);

lcd.init(PHILIPS);
lcd.contrast(60);
lcd.clear(BACKGROUND);
}

void loop() {
  // put your main code here, to run
  repeatedly
  vb = true;
  digitalWrite(ledpinm , HIGH);
  loopbutton = true;
  f=Serial.readString();
  if (f.equals("A")) {
    lcd.clear(BACKGROUND);
    ab = true;
  }
  while (ab){
    lcd.clear(BACKGROUND);

    lcd.setStr("Tidak Ada", 40,30, WHITE,
RED);

    lcd.setStr("Koneksi ke PC", 57,14,
WHITE, RED);

    lcd.setRect(0,0,18,77,1,GREEN);
    lcd.setRect(0,120,18,132,1,GREEN);

    lcd.setStr("FT.UNJ", 0,77, WHITE,
GREEN);

    lcd.setStr("by:Idris", 93,2, BLACK,
BACKGROUND);

    lcd.setStr("5215111741", 107,29,
BLACK, BACKGROUND);

    myservo.write(162);

    f=Serial.readString();

    if (f.equals("A")) {
      lcd.clear(BACKGROUND);
      ab = false;
    }
  }

  lcd.setStr("Reader", 20,37, BLACK,
BACKGROUND);
  lcd.setStr("Siap digunakan", 37,8,
BLACK,BACKGROUND );
  lcd.setRect(0,0,18,77,1,GREEN);
  lcd.setRect(0,120,18,132,1,GREEN);
  lcd.setStr("FT.UNJ", 0,77, WHITE,
GREEN);
  lcd.setStr("by:Idris", 93,2, BLACK,
BACKGROUND);
  lcd.setStr("5215111741", 107,29,
BLACK, BACKGROUND);
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }

  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }

  // Dump debug info about the card;
  PICC_HaltA() is automatically called
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ?
"0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
  }
}

```

```

// Halt PICC
Serial.println();

a= String(mfrc522.uid.uidByte[0] < 0x10 ?
" 0" : " ");
a+= String(mfrc522.uid.uidByte[0], HEX);
b= String(mfrc522.uid.uidByte[1] < 0x10
? " 0" : " ");
b+= String(mfrc522.uid.uidByte[1], HEX);
c= String(mfrc522.uid.uidByte[2] < 0x10 ?
" 0" : " ");
c+= String(mfrc522.uid.uidByte[2], HEX);
d= String(mfrc522.uid.uidByte[3] < 0x10
? " 0" : " ");
d+= String(mfrc522.uid.uidByte[3], HEX);

e=a+b+c+d+ " ";
char *baru;
// e.toCharArray(baru,e.length()+1);
baru= &e[0u];

lcd.setStr(baru,55,12,BLUE,BACKGROUN
D);

while(vb){
Serial.flush();
f=Serial.readString();
bace=String(f[0]);
for(byte i = 1; i < f.length();i++){
baca+=String(f[i]); //data2=data2+data[i]
}
//
baca.toCharArray(data,baca.length()+1);
data = &baca[0u];
//
lcd.setStr(data,75,4,BLUE,BACKGROUND
);
if (f.length())>1){
vb = false;
}
}
if (bace.equals("1")) {
lcd.setStr(data,75,4,WHITE,GREEN);
digitalWrite(ledpinh , HIGH);
digitalWrite(ledpinm , LOW);
myservo.write(145);
delay(200);
while (loopbutton){
buttonstate=digitalRead(button);
if(buttonstate==LOW){
digitalWrite(ledpinm , HIGH);
myservo.write(162);
delay(200);
lcd.clear(BACKGROUND);
digitalWrite(ledpinh , LOW);
f="";
baca="";
loopbutton = false;
}
}
}
else if (data1==0) {
lcd.setStr(data,75,4,WHITE,RED);
myservo.write(162);
delay(2000);
lcd.clear(BACKGROUND);
f="";
baca="";
}
mfrc522.PICC_HaltA();
mfrc522.PCD_StopCrypto1();

```

```

}

Arduino C
#include <SparkFunColorLCDShield.h>

#include <ColorLCDShield.h>

#include <Servo.h>

#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN 9
#define SS_PIN 10
#define BACKGROUND WHITE

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

boolean loopbutton = true;
boolean vb = true;
boolean ab = true;

#define button A0
const int ledpinh = A5;
const int ledpinm = A4;
int buttonstate=0;
Servo myservo;

String a;
String b;
String c;
String d;

String e;
String f;
String baca;
String bace;
char* data;
char* data1;
char* data2;

void setup() {
    // put your setup code here, to run once:

    Serial.begin(9600); // Initialize serial
    communications with the PC

    while (!Serial); // Do nothing if no serial
    port is opened (added for Arduinos based on
    ATMEGA32U4)

    SPI.begin(); // Init SPI bus
    mfrc522.PCD_Init();

    pinMode(button,INPUT_PULLUP);
    pinMode(ledpinh,OUTPUT);
    pinMode(ledpinm,OUTPUT);
    myservo.attach(3);

    lcd.init(PHILIPS);
    lcd.contrast(60);
    lcd.clear(BACKGROUND);
}

void loop() {
    // put your main code here, to run
    repeatedly

    vb = true;
    digitalWrite(ledpinm , HIGH);
    loopbutton = true;
    f=Serial.readString();
    if (f.equals("A")) {
        lcd.clear(BACKGROUND);

```

```

    ab = true;

}

while (ab){
    lcd.clear(BACKGROUND);

    lcd.setStr("Tidak Ada", 40,30, WHITE,
RED);

    lcd.setStr("Koneksi ke PC", 57,14,
WHITE, RED);

    lcd.setRect(0,0,18,77,1,GREEN);
    lcd.setRect(0,120,18,132,1,GREEN);

    lcd.setStr("FT.UNJ", 0,77, WHITE,
GREEN);

    lcd.setStr("by:Idris", 93,2, BLACK,
BACKGROUND);

    lcd.setStr("5215111741", 107,29,
BLACK, BACKGROUND);

myservo.write(145);

f=Serial.readString();
if (f.equals("A")) {
    lcd.clear(BACKGROUND);
    ab = false;
}
}

    lcd.setStr("Reader", 20,37, BLACK,
BACKGROUND);

    lcd.setStr("Siap digunakan", 37,8,
BLACK,BACKGROUND );

    lcd.setRect(0,0,18,77,1,GREEN);
    lcd.setRect(0,120,18,132,1,GREEN);

    lcd.setStr("FT.UNJ", 0,77, WHITE,
GREEN);

    lcd.setStr("by:Idris", 93,2, BLACK,
BACKGROUND);

    lcd.setStr("5215111741", 107,29,
BLACK, BACKGROUND);

if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
}

// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
}

// Dump debug info about the card;
PICC_HaltA() is automatically called
for (byte i = 0; i < mfrc522.uid.size; i++)
{
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ?
" 0" : " ");

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

}

// Halt PICC
Serial.println();

a= String(mfrc522.uid.uidByte[0] < 0x10 ?
" 0" : " ");

a+= String(mfrc522.uid.uidByte[0], HEX);

b= String(mfrc522.uid.uidByte[1] < 0x10
? " 0" : " ");

b+= String(mfrc522.uid.uidByte[1], HEX);

c= String(mfrc522.uid.uidByte[2] < 0x10 ?
" 0" : " ");

c+= String(mfrc522.uid.uidByte[2], HEX);

d= String(mfrc522.uid.uidByte[3] < 0x10
? " 0" : " ");

d+= String(mfrc522.uid.uidByte[3], HEX);

e=a+b+c+d+" ";

char *baru;

```

```

// e.toCharArray(baru,e.length()+1);
baru= &e[0u];

lcd.setStr(baru,55,12,BLUE,BACKGROUND);

while(vb){
  f=Serial.readString();
  bace=String(f[0]);
  for(byte i = 1; i < f.length();i++){
    baca+=String(f[i]); //data2=data2+data[i]
  }
//
baca.toCharArray(data,baca.length()+1);
  data = &baca[0u];
//
lcd.setStr(data,75,4,BLUE,BACKGROUND);
  if (f.length()>1){

    vb = false;
  }
}

if (bace.equals("1")) {
  lcd.setStr(data,75,4,WHITE,GREEN);
  digitalWrite(ledpinh , HIGH);
  digitalWrite(ledpinm , LOW);
  myservo.write(130);
  delay(200);
  while (loopbutton){
    buttonstate=digitalRead(button);
    if(buttonstate==LOW){
      digitalWrite(ledpinm , HIGH);
      myservo.write(145);
      delay(200);

      lcd.clear(BACKGROUND);
      digitalWrite(ledpinh , LOW);
      f="";
      baca="";
      loopbutton = false;
    }
  }
}
else if (data1==0) {
  lcd.setStr(data,75,4,WHITE,RED);
  myservo.write(145);
  delay(2000);
  lcd.clear(BACKGROUND);
  f="";
  baca="";
}
mfr522.PICC_HaltA();
mfr522.PCD_StopCrypto1();
}

Arduino server
/*
 * MFRC522 - Library to use ARDUINO RFID
MODULE KIT 13.56 MHZ WITH TAGS SPI W
AND R BY COOQROBOT.
 * The library file MFRC522.h has a wealth
of useful info. Please read it.
 * The functions are documented in
MFRC522.cpp.
 *
 * Based on code Dr.Leong (
WWW.B2CQSHOP.COM )
 * Created by Miguel Balboa
(circuitito.com), Jan, 2012.
 * Rewritten by Søren Thing Andersen
(access.thing.dk), fall of 2013 (Translation to

```

English, refactored, comments, anti collision, cascade levels.)

* Released into the public domain.
*
* This sample shows how to setup a block on a MIFARE Classic PICC to be in "Value Block" mode.
* In Value Block mode the operations Increment/Decrement/Restore and Transfer can be used.

*

----- empty_skull

* Aggiunti pin per arduino Mega
* add pin configuration for arduino mega
* <http://mac86project.altervista.org/>

----- Nicola Coppola

* Pin layout should be as follows:

* Signal	Pin	Pin	Pin
* Arduino Uno		Arduino Mega	
MFRC522 board			

* -----

* Reset	9	5	RST
* SPI SS	10	53	SDA
* SPI MOSI	11	51	MOSI
* SPI MISO	12	50	MISO
* SPI SCK	13	52	SCK

*
* The reader can be found on eBay for around 5 dollars. Search for "mf-rc522" on ebay.com.

*/

#include <SPI.h>

#include <MFRC522.h>

```
#define SS_PIN 10
#define RST_PIN 9

MFRC522 mfrc522(SS_PIN, RST_PIN);
    // Create MFRC522 instance.

String z="";

boolean buka=true;

void setup() {
    Serial.begin(9600);    //
Initialize serial communications with the PC

    SPI.begin();        // Init
SPI bus

    mfrc522.PCD_Init();    // Init
MFRC522 card

    //Serial.println("Scan a MIFARE
Classic PICC to demonstrate Value Blocks.");
}

void loop() {
    // Look for new cards

    z=Serial.readString();

    if (z=="buka"){
        buka=true;
    }

// Serial.print(z);

    if (buka){

        if ( !
mfrc522.PICC_IsNewCardPresent()) {

            return;

        }

    }

    else{

        buka = false;

    }
}
```

```

        // Select one of the cards
        if ( !
mfr522.PICC_ReadCardSerial() {
            return;
        }
        // Now a card is selected. The UID
and SAK is in mfr522.uid.

        // Dump UID
        //Serial.print("Card UID:");
        for (byte i = 0; i < mfr522.uid.size;
i++) {
            Serial.print(mfr522.uid.uidByte[i]
< 0x10 ? " 0" : " ");

            Serial.print(mfr522.uid.uidByte[i],
HEX);
        }
        //Serial.println(z);
        Serial.println();

/*

        // Dump PICC type
        byte piccType =
mfr522.PICC_GetType(mfr522.uid.sak);
        Serial.print("PICC type: ");
        Serial.println(mfr522.PICC_GetType
eName(piccType));
        if ( piccType !=
MFRC522::PICC_TYPE_MIFARE_MINI
            && piccType !=
MFRC522::PICC_TYPE_MIFARE_1K
            && piccType !=
MFRC522::PICC_TYPE_MIFARE_4K) {
            Serial.println("This sample
only works with MIFARE Classic cards.");
        }
    }
}

return;

// Prepare key - all keys are set to
FFFFFFFFFFFFh at chip delivery from the
factory.
MFRC522::MIFARE_Key key;
for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
}

// In this sample we use the second
sector (ie block 4-7).
byte sector
= 1;
byte valueBlockA = 5;
byte valueBlockB = 6;
byte trailerBlock = 7;

// Authenticate using key A.
Serial.println("Authenticating using
key A...");
byte status;
status =
mfr522.PCD_Authenticate(MFRC522::PICC
_CMD_MF_AUTH_KEY_A, trailerBlock,
&key, &(mfr522.uid));
if (status != MFRC522::STATUS_OK)
{
    Serial.print("PCD_Authenticate()
failed: ");
    Serial.println(mfr522.GetStatusCo
deName(status));
    return;
}
}
}

```

```

        // We need a sector trailer that
        defines blocks 5 and 6 as Value Blocks and
        enables key B.

        byte trailerBuffer[] = {
        255,255,255,255,255,255, 0,0,0, 0,
        255,255,255,255,255,255}; // Keep default
        keys.

        // g1=6 => Set block 5 as value
        block. Must use Key B to write & increment,
        A or B can be used for derement.

        // g2=6 => Same thing for block 6.

        // g3=3 => Key B must be used to
        modify the Sector Trailer. Key B becomes
        valid.

        mfr522.MIFARE_SetAccessBits(&t
        railerBuffer[6], 0, 6, 6, 3);

        // Now we read the sector trailer
        and see if it is like we want it to be.

        Serial.println("Reading sector
        trailer...");

        byte buffer[18];

        byte size = sizeof(buffer);

        status =
        mfr522.MIFARE_Read(trailerBlock, buffer,
        &size);

        if (status != MFRC522::STATUS_OK)
        {

        Serial.print("MIFARE_Read() failed:
        ");

        Serial.println(mfr522.GetStatusCo
        deName(status));

        return;

        }

        if (    trailerBuffer[6] !=
        &&    trailerBuffer[7] !=
        &&    trailerBuffer[8] !=
        trailerBuffer[8]) {

```

```

        Serial.println("Writing
        new sector trailer...");

        status =
        mfr522.MIFARE_Write(trailerBlock,
        trailerBuffer, 16);

        if (status !=
        MFRC522::STATUS_OK) {

        Serial.print("MIFARE_Write()
        failed: ");

        Serial.println(mfr522.GetStatusCo
        deName(status));

        return;

        }

        }

        // Authenticate using key B.

        Serial.println("Authenticating again
        using key B...");

        status =
        mfr522.PCD_Authenticate(MFRC522::PICC
        _CMD_MF_AUTH_KEY_B, trailerBlock,
        &key, &(mfr522.uid));

        if (status != MFRC522::STATUS_OK)
        {

        Serial.print("PCD_Authenticate()
        failed: ");

        Serial.println(mfr522.GetStatusCo
        deName(status));

        return;

        }

        // Value blocks has a 32 bit signed
        value stored three times and an 8 bit
        address stored 4 times.

        // Make sure blocks valueBlockA
        and valueBlockB has that format.

        formatBlock(valueBlockA);

```



```

formatBlock(valueBlockB);

// Add 1 to the value of
valueBlockA and store the result in
valueBlockA.

Serial.print("Adding 1 to value of
block "); Serial.println(valueBlockA);

status =
mfrc522.MIFARE_Increment(valueBlockA,
1);

if (status != MFRC522::STATUS_OK)
{

Serial.print("MIFARE_Increment()
failed: ");

Serial.println(mfrc522.GetStatusCo
deName(status));

return;

}

status =
mfrc522.MIFARE_Transfer(valueBlockA);

if (status != MFRC522::STATUS_OK)
{

Serial.print("MIFARE_Transfer()
failed: ");

Serial.println(mfrc522.GetStatusCo
deName(status));

return;

}

// Dump the result

mfrc522.PICC_DumpMifareClassicS
ectorToSerial(&(mfrc522.uid), &key, sector);
*/

// Halt PICC

mfrc522.PICC_HaltA();

// Stop encryption on PCD

mfrc522.PCD_StopCrypto1();

}

// while (buka){
// }

}

/*()

void formatBlock(byte blockAddr) {

Serial.print("Reading block ");
Serial.println(blockAddr);

byte buffer[18];

byte size = sizeof(buffer);

byte status =
mfrc522.MIFARE_Read(blockAddr, buffer,
&size);

if (status != MFRC522::STATUS_OK)
{

Serial.print("MIFARE_Read() failed:
");

Serial.println(mfrc522.GetStatusCo
deName(status));

return;

}

if ( (buffer[0] ==
(byte)~buffer[4])
&& (buffer[1] ==
(byte)~buffer[5])
&& (buffer[2] ==
(byte)~buffer[6])
&& (buffer[3] ==
(byte)~buffer[7])
&& (buffer[0] ==
buffer[8])

```

```

        && (buffer[1] ==
buffer[9])
        && (buffer[2] ==
buffer[10])
        && (buffer[3] ==
buffer[11])

        && (buffer[12] ==
(byte)~buffer[13])
        && (buffer[12] ==
buffer[14])
        && (buffer[12] ==
(byte)~buffer[15])) {
            Serial.println("Block has
correct Block Value format.");
        }
        else {
            Serial.println("Writing
new value block...");

            byte valueBlock[] = {
0,0,0,0, 255,255,255,255, 0,0,0,0,
blockAddr,~blockAddr,blockAddr,~blockAdd
r };

            status =
mfrc522.MIFARE_Write(blockAddr,
valueBlock, 16);

            if (status !=
MFRC522::STATUS_OK) {

                Serial.print("MIFARE_Write()
failed: ");

                Serial.println(mfrc522.GetStatusCo
deName(status));
            }
        }
    } // End formatBlock()

*/

```