

**KARTU BERLANGGANAN PERTANDINGAN SEPAK
BOLA MEMANFAATKAN TEKNOLOGI RFID (*RADIO
FREQUENCY IDENTIFICATION*)**

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

HANDINY INDAH P.

NIM. 0710630067-63

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG**

2011

repository.ub.ac.id

**KARTU BERLANGGANAN PERTANDINGAN SEPAK
BOLA MEMANFAATKAN TEKNOLOGI RFID (*RADIO
FREQUENCY IDENTIFICATION*)**

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:
HANDINY INDAH P.
NIM. 0710630067-63

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Agung Darmawansyah, ST., MT.

NIP. 19721218 199903 1 002

Ir. Ponco Siwindarto, M.Eng.Sc

NIP. 19590304 198903 1 001



LEMBAR PENGESAHAN

**KARTU BERLANGGANAN PERTANDINGAN SEPAK
BOLA MEMANFAATKAN TEKNOLOGI RFID (*RADIO
FREQUENCY IDENTIFICATION*)**

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:

**HANDINY INDAH P.
NIM. 0710630067-63**

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 23 Desember 2011

DOSEN PENGUJI

Ir. M. Julius St, MS
NIP. 19540720 198203 1 002

Ir. Nurrusa'adah, MT.
NIP. 19680706 199203 2 001

Mochammad Rif'an, ST., MT.
NIP. 19710301 200012 1 001

Mengetahui
Ketua Jurusan Teknik Elektro

Dr. Ir. Sholeh Hadi Pramono, MS
NIP. 19580728 198701 1 001

PENGANTAR



Puji dan syukur penulis panjatkan kepada Allah SWT, karena dengan rahmat, taufik dan hidayah-Nya lah skripsi ini dapat diselesaikan.

Skripsi berjudul “Kartu Berlangganan Pertandingan Sepak Bola Memanfaatkan Teknologi RFID (*Radio Frequency Identification*)” ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

- Mama, Emak, Ibu Yuyu Sriwachyuni dan Papa Najmudin atas segala doa, nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta telah banyak mendoakan kelancaran penulis hingga terselesaikannya skripsi ini,
- Fahmi Nurul Akbar, yang telah memberikan lebih dari segalanya yang bisa diberikan untuk penulis,
- Keluargaku semua, Adikku Dahak Dhania Komalasari, Tete Drh. Devi Paramitha, Mamih Jubaedah, Mama Tati, Papa Nuris, Mba Ruly, Dek Niar dan seluruh keluarga besar terima kasih atas doa restu selama ini untuk penulis,
- Dr. Ir. Sholeh Hadi Pramono., MS selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya,
- M. Aziz Muslim, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
- Ir. M. Julius St, MS selaku Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya,
- Dr. Agung Darmawansyah, ST., MT. dan Ir. Ponco Siwindarto, M.Eng.Sc. selaku Dosen Pembimbing 1 dan Dosen Pembimbing 2 atas segala bimbingan, nasehat, pengarahan, motivasi, saran dan masukan yang telah diberikan,
- Staf rekording Jurusan Teknik Elektro,
- Ucup, Rian, Wahyu, Galcit, Brian “Gor”, Mbah Kiwil, Mas Jul yang telah ikut berpartisipasi dalam pengerjaan skripsi ini,

- Refita Edna K dan Atika Iqlimah yang selalu menjadi teman, sahabat, saudara dalam suka maupun duka dari maba sampai selamalamanya,
- Teman bermain dan belajar : Inyaz, Hupy, Puput, Cicu, Yayuk, Ita, Nia, Titis, Cici, Turen, Adel, Rebon, Maul, Anok, Kecil, Delu, Utari, Anas, Rizal, Yuli, Indrahamsyong,
- Teman-teman Core yang penuh dengan kemesraan, teman-teman paket B'07 yang selalu memberi motivasi dan dukungan dengan “kompetisi”nya, WS, asisten Sisdig, teman-teman, senior serta semua pihak yang tidak mungkin bagi penulis untuk mencantumkan namanya satu-persatu, terima kasih banyak atas bantuan dan dukungannya.

Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.



Malang,
Desember 2011

Penulis

ABSTRAK

Handiny Indah P., Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Desember 2011, *Kartu Berlangganan Pertandingan Sepak Bola Memanfaatkan RFID(Radio Frequency Identification)*, Dosen Pembimbing: Dr. Agung Darmawansyah, ST., MT. dan Ir. Ponco Siwindarto, M.Eng.Sc.

Indonesia merupakan negara dengan antusiasme terhadap sepak bola yang besar. Setiap klub dari liga domestik memiliki pendukung yang fanatik. Kekompakan dan loyalitas mereka sudah tidak dipertanyakan lagi. Buktinya banyak kelompok suporter yang dianugerahi sebagai Suporter Terbaik. Tetapi kebutuhan dan antusiasme suporter yang sangat besar masih belum didukung dengan fasilitas yang memadai, misalkan sistem penjualan tiket dan sistem pemeriksaan tiket masuk ke stadion. Saat ini penjualan tiket dilakukan dengan cara penyebaran melalui koordinator wilayah (korwil), radio, dan tiket boks. Tetapi hal yang terjadi adalah calo akan memborong tiket dan menjual kembali dengan harga tinggi. Sedangkan karena sistem pemeriksaan tiket yang masih manual, yaitu dengan menggunakan tenaga operator, tiket yang berupa selebar kertas dapat dengan mudah dipalsukan sehingga sulit untuk dibedakan dengan yang asli. Oleh karena itu, dibutuhkan suatu sistem kartu berlangganan pengganti tiket pertandingan yang secara otomatis dapat mengendalikan pintu masuk stadion.

Maka dalam skripsi ini dirancang Kartu Berlangganan Pertandingan Sepak Bola memanfaatkan Teknologi RFID (Radio Frequency Identification). Sistem ini terdiri atas RFID reader, mikrokontroler ATmega162, komputer yang menghubungkan mikrokontroler dengan database MySQL menggunakan program aplikasi Borland Delphi, printer, motor DC dan LCD. RFID reader akan mendeteksi tag RFID dan mengirimkan nomor serial ke mikrokontroler melalui komunikasi serial. Oleh mikrokontroler data tersebut dikirimkan ke komputer untuk diperiksa pada database apakah nomor serial tersebut telah terdaftar atau tidak. Jika tidak terdaftar LCD menampilkan 'Kartu Anda Tidak Terdaftar' sedangkan jika telah terdaftar counter pada mikrokontroler akan aktif dan data dari counter ditampilkan di LCD sebagai nomor tempat duduk. Data nomor tempat duduk dikirimkan ke komputer yang kemudian data tersebut akan digabungkan dengan data tiket pertandingan yang telah dipesan sebelumnya untuk dicetak menggunakan printer. Bersamaan dengan itu driver motor aktif dan palang pintu terbuka selama 5 detik dan akan tertutup secara otomatis.

Dari hasil pengujian, menunjukkan bahwa RFID ID-12 dapat digunakan sebagai sensor identifikasi karena kombinasinya yang berbeda satu sama lain. Sensor ini memberi masukan kepada Mikrokontroler berupa kode ASCII dan kemudian diolah sebagai data oleh Mikrokontroler. RFID ID-12 ini dapat bekerja dengan baik pada jarak pembacaan ± 9 cm saat Tag RFID didekatkan tegak lurus dari RFID Reader, dan ± 5 cm saat posisi Tag RFID bersudut $\pm 45^\circ$ dari posisi RFID Reader tanpa penghalang. Jika menggunakan penghalang berbahan plastik dengan ketebalan ± 3 mm dan jarak ± 2 cm, RFID reader dapat bekerja dengan baik pada jarak pembacaan ± 6 cm saat Tag RFID didekatkan tegak lurus dari RFID Reader, dan ± 3 cm saat posisi Tag RFID bersudut $\pm 45^\circ$.

Kata kunci: RFID, kartu berlangganan, sepak bola, palang pintu.

DAFTAR ISI

PENGANTAR	i
ABSTRAK	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	2
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	5
2.1 RFID (<i>Radio Frequency Identification</i>)	5
2.2 Mikrokontroler ATmega162	7
2.3 Motor DC	10
2.3.1 Prinsip Kerja Motor DC	10
2.3.2 Pengaturan Motor DC	11
2.4 LCD (<i>Liquid Crystal Display</i>)	12
2.5 Borland Delphi 7	13
2.6 MySQL	15
2.6.1 Administrasi <i>User</i>	16
2.6.2 Merancang <i>Database</i>	16
2.7 Komunikasi Serial	19
BAB III METODOLOGI PENELITIAN	21
3.1 Perancangan Alat	21
3.2 Pembuatan Alat	21

3.3 Pengujian Alat	21
BAB IV PERANCANGAN DAN PEMBUATAN ALAT	23
4.1 Penentuan Spesifikasi Alat	23
4.2 Diagram Blok Sistem	23
4.3 Perancangan Perangkat Keras (<i>Hardware</i>)	25
4.3.1 Perancangan Rangkaian RFID <i>Reader</i>	25
4.3.2 Perancangan Rangkaian RS232	27
4.3.3 Perancangan Rangkaian Mikrokontroler	28
4.3.4 Perancangan Rangkaian <i>Driver</i> Motor DC	29
4.3.5 Perancangan Rangkaian LCD	31
4.4 Perancangan Perangkat Lunak (<i>Software</i>)	31
4.4.1 Perancangan Database	31
4.4.2 Perancangan Perangkat Lunak Komputer	32
4.4.2.1 Perancangan Perangkat Lunak Komputer pada Proses Pemesanan Tiket	32
4.4.2.2 Perancangan Perangkat Lunak Komputer di Hari Pertandingan	33
4.4.3 Perancangan Perangkat Lunak Mikrokontroler	35
BAB V PENGUJIAN	38
5.1 Pengujian Program Aplikasi sebagai Penghubung dengan <i>Database</i>	38
5.1.1 Pengujian Program Aplikasi untuk Mencari Data dari dalam <i>Database</i>	38
5.1.2 Pengujian Program Aplikasi untuk Mengubah Data pada <i>Database</i>	42
5.1.3 Pengujian Program Aplikasi untuk Memasukkan Data ke dalam <i>Database</i>	45
5.1.4 Pengujian Program Aplikasi untuk Menghapus Data	50
5.2 Pengujian Rangkaian RFID <i>Reader</i> ID-12 dan RS232	52
5.3 Pengujian LCD	54
5.4 Pengujian <i>Counter</i> Sebagai Penentu Nomor Tempat Duduk	55
5.5 Pengujian Mencetak Data Menggunakan <i>Printer</i>	57
5.6 Pengujian Rangkaian <i>Driver</i> Motor DC	59

5.7 Pengujian Sistem Secara Keseluruhan	60
BAB VI KESIMPULAN DAN SARAN	65
6.1 Kesimpulan	65
6.2 Saran	66
DAFTAR PUSTAKA	67



DAFTAR GAMBAR

Gambar 2.1 Komunikasi Antara *Reader* dan *Transmitter (Tag)* 6

Gambar 2.2 Konfigurasi Pin ID-12 (*RFID Reader*) 7

Gambar 2.3 Konfigurasi Pin Mikrokontroler ATmega162 9

Gambar 2.4 Gambar Motor Secara Umum 11

Gambar 2.5 Rangkaian *Interface* ke LCD Karakter 2x16 13

Gambar 2.6 IDE Delphi 14

Gambar 2.7 Pengiriman Huruf “A” Tanpa bit Paritas 20

Gambar 2.8 Konfigurasi Port Serial 20

Gambar 4.1 Diagram Blok Sistem 24

Gambar 4.2 Rangkaian *RFID Reader* 26

Gambar 4.3 Rangkaian MAX232 yang terdiri atas IC max232 dan 4 buah kapasitor .. 27

Gambar 4.4 Rangkaian Mikrokontroler ATmega162 28

Gambar 4.5 Rangkaian *Driver Motor DC* 29

Gambar 4.6 Rangkaian LCD 31

Gambar 4.7 Hubungan Antara Tabel pada *Database* 32

Gambar 4.8 Diagram Alir Proses Pemesanan Tiket 33

Gambar 4.9 Diagram Alir Perancangan Perangkat Lunak Komputer di Hari
Pertandingan 34

Gambar 4.10 Diagram Alir Perancangan Perangkat Lunak Mikrokontroler 36

Gambar 5.1 Diagram Blok Pengujian Program Aplikasi sebagai Penghubung dengan
Database 38

Gambar 5.2 Tampilan Program Aplikasi Memasukkan Data dalam Kotak Pencarian .. 39

Gambar 5.3 Tampilan Program Aplikasi Saat Data Telah Ditampilkan 40

Gambar 5.4 Tampilan Program Aplikasi Saat Data Telah Ditampilkan 41

Gambar 5.5 Tampilan Program Aplikasi Saat Operator Memasukkan Data yang Salah 42

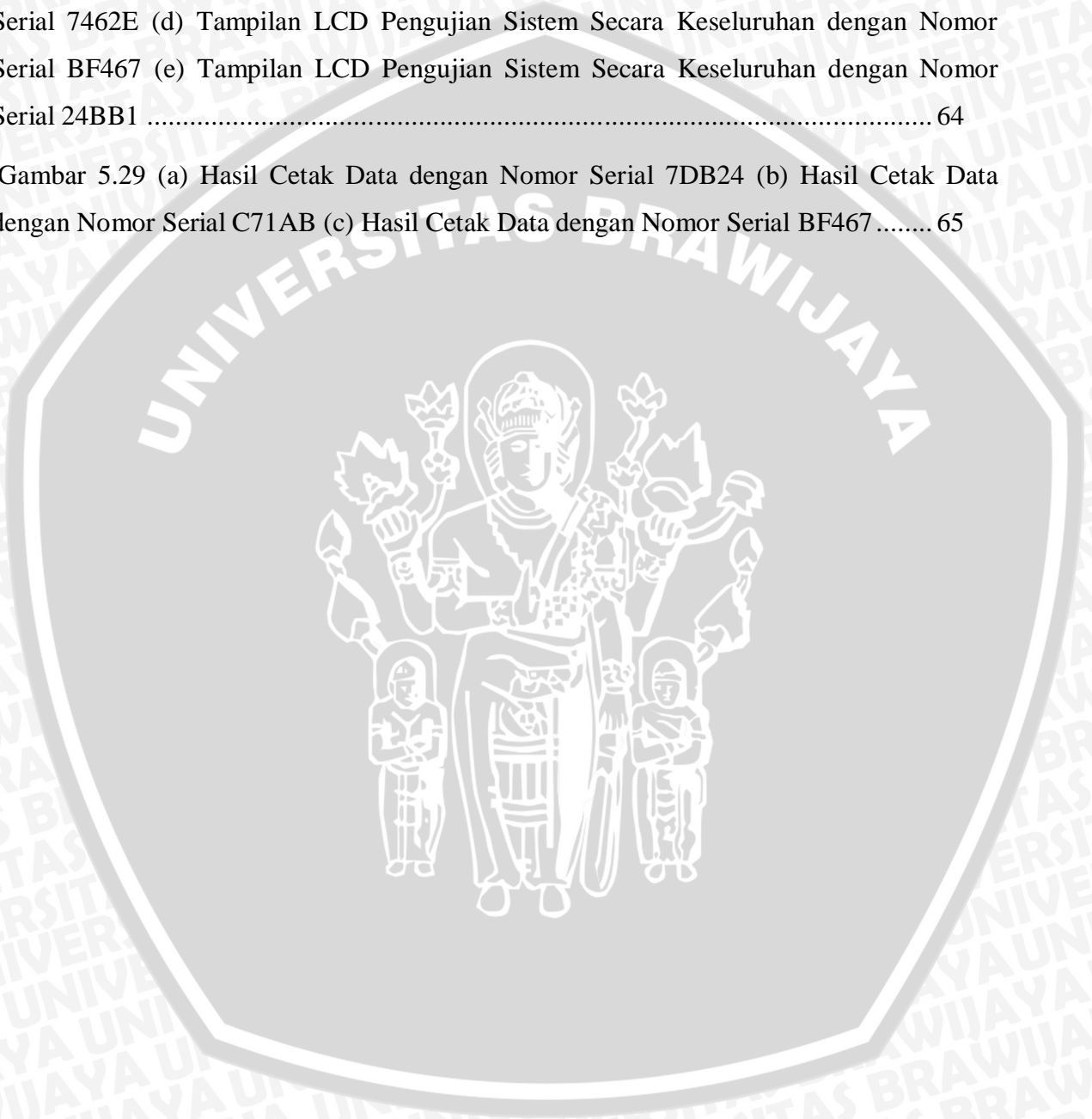
Gambar 5.6 Tampilan Program Aplikasi Sebelum Data Diubah 43

Gambar 5.7 Tampilan Program Aplikasi Setelah Data Diubah	43
Gambar 5.8 Tampilan Program Aplikasi Setelah Data Disimpan	44
Gambar 5.9 Tampilan Program Aplikasi Memasukkan Data	46
Gambar 5.10 Tampilan Program Aplikasi Memasukkan Data	47
Gambar 5.11 Tampilan Program Aplikasi Memasukkan Data	49
Gambar 5.12 (a) Tampilan Program Aplikasi Memasukkan Data dengan Memberi Tanda <i>Checklist</i> (b) Tampilan Program Aplikasi Memasukkan Data dan Menghapus Tanda <i>Checklist</i>	53
Gambar 5.13 Diagram Blok Pengujian Rangkaian RFID <i>Reader</i> ID-12 dan RS232.....	53
Gambar 5.14 Hasil Pengujian Rangkaian RFID <i>Reader</i> ID-12 dan RS232.....	53
Gambar 5.15 Diagram Blok Pengujian Tampilan LCD	55
Gambar 5.16 Hasil Pengujian LCD	55
Gambar 5.17 Diagram Blok Pengujian <i>Counter</i> sebagai Penentu Nomor Tempat Duduk	56
Gambar 5.18 (a) Kondisi Awal Sebelum Ada Masukan <i>Tag</i> RFID (b) Tampilan LCD Saat Nomor Serial <i>Tag</i> RFID Telah Terdaftar pada <i>Database</i> (c) Tampilan LCD Saat Nomor Serial <i>Tag</i> RFID Tidak Terdaftar pada <i>Database</i>	56
Gambar 5.19 Diagram Blok Pengujian Mencetak Data Menggunakan <i>Printer</i>	58
Gambar 5.20 Tampilan Isi Program yang Akan Dicitak	58
Gambar 5.21 (a) <i>Preview</i> Data yang Akan Dicitak (b) Hasil <i>Printout</i> Data	59
Gambar 5.22 Diagram Blok Pengujian Rangkaian <i>Driver</i> Motor DC	60
Gambar 5.23 Diagram Blok Pengujian Rangkaian Keseluruhan Sistem	61
Gambar 5.24 (a) Pengisian Nomor Serial yang Tidak Sesuai (b) Pengisian Nomor Serial yang Sesuai	62
Gambar 5.25 (a) Pemesanan Enam Tiket Pertandingan (b) Saldo pada Kartu Tidak Cukup untuk Melakukan Transaksi (c) Saldo pada Kartu Mencukupi untuk Pemesanan Dua Tiket	62
Gambar 5.26 Saldo Secara Otomatis Berkurang	63

Gambar 5.27 (a) Pengisian Tanggal yang Tidak Sesuai pada Program Aplikasi
 (b) Pengisian Tanggal yang Sesuai pada Program Aplikasi..... 64

Gambar 5.28 (a) Tampilan LCD Pengujian Sistem Secara Keseluruhan dengan Nomor Serial 7DB24 (b) Tampilan LCD Pengujian Sistem Secara Keseluruhan dengan Nomor Serial C71AB (c) Tampilan LCD Pengujian Sistem Secara Keseluruhan dengan Nomor Serial 7462E (d) Tampilan LCD Pengujian Sistem Secara Keseluruhan dengan Nomor Serial BF467 (e) Tampilan LCD Pengujian Sistem Secara Keseluruhan dengan Nomor Serial 24BB1 64

Gambar 5.29 (a) Hasil Cetak Data dengan Nomor Serial 7DB24 (b) Hasil Cetak Data dengan Nomor Serial C71AB (c) Hasil Cetak Data dengan Nomor Serial BF467 65



DAFTAR TABEL

Tabel 2.1 Fungsi dan Format Data Keluaran	7
Tabel 2.2 Rumus Menghitung <i>Baudrate</i>	10
Tabel 2.3 Pin-pin I/O LCD	13
Tabel 2.4 Keterangan Pin-Pin Port Serial	20
Tabel 5.1 Tabel <i>Database</i> Data Diri.....	40
Tabel 5.2 Tabel <i>Database</i> Data Diri	41
Tabel 5.3 Tabel Data Diri pada <i>Database</i> Sebelum Data Diubah	44
Tabel 5.4 Tabel Data Diri pada <i>Database</i> Setelah Data Diubah	44
Tabel 5.5 Tabel Jadwal Pertandingan pada <i>Database</i> Belum Terisi Data	45
Tabel 5.6 Tabel Jadwal Pertandingan pada <i>Database</i>	46
Table 5.7 Tabel Jadwal Pertandingan pada <i>Database</i>	48
Tabel 5.8 Tabel Jadwal Pertandingan pada <i>Database</i>	49
Tabel 5.9 Tabel Jadwal Pertandingan pada <i>Database</i> Sebelum Data Dihapus.....	51
Table 5.10 Tabel Jadwal Pertandingan pada <i>Database</i> Setelah Data Dihapus	52
Tabel 5.11 Hasil Pengujian Jangkauan RFID <i>Reader</i>	54
Table 5.12 Hasil Pengujian Jangkauan RFID <i>Reader</i> dengan Menggunakan Penghalang	54
Tabel 5.13 Data Pengujian <i>Counter</i> Sebagai Penentu Nomor Tempat Duduk	57
Tabel 5.14 Hasil Pengujian Rangkaian <i>Driver</i> Motor DC.....	60
Table 5.15 Hasil Pengujian Terhadap Palang Pintu.....	66

BAB I PENDAHULUAN

1.1 Latar Belakang

Sepak bola merupakan olahraga dengan menggunakan bola yang dimainkan oleh dua tim dengan masing-masing beranggotakan sebelas orang. Olahraga ini sangat terkenal dan dimainkan hampir di seluruh dunia, termasuk di Indonesia.

Antusiasme masyarakat Indonesia terhadap sepak bola sangatlah tinggi. Masyarakat tidak membedakan pertandingan sepak bola tersebut berasal dari liga internasional ataupun liga domestik, semua senang menyaksikannya. Untuk liga domestik, Indonesia memiliki satu liga sepak bola yang dinaungi oleh Persatuan Sepakbola Seluruh Indonesia (PSSI). Dalam liga tersebut, klub-klub dari berbagai daerah di Indonesia bertanding.

Sikap pendukung untuk membela klubnya terlihat saat pertandingan berlangsung, stadion yang digunakan selalu dipadati oleh penonton. Tribun kelas ekonomi, VIP, atau VVIP tidak pernah kosong. Apalagi saat pertandingan besar, tiket dipastikan akan terjual habis. Dengan kebutuhan pendukung yang sangat besar terhadap tiket pertandingan, menyebabkan banyak pihak tidak bertanggung jawab yang memanfaatkan situasi tersebut dengan mengedarkan tiket palsu. Contohnya terjadi pada klub Arema Indonesia.

Kerugian dari pemalsuan tiket paling dirasakan manajemen saat Arema menjamu Persija. Berdasarkan hasil hitungan oleh Dinas Pendapatan Daerah setempat, jumlah penonton berjumlah 32.957 orang. Sedangkan jumlah tiket yang terjual hanya 29.400 lembar sehingga ada selisih 3.197 tiket. Jika dipotong tiket gratis (*freepass*) sebanyak 850 lembar, maka ditaksir penonton masuk tanpa tiket berjumlah 2.347 orang (Tempointeraktif).

Selain pemalsuan tiket, masalah yang merugikan klub-klub di Indonesia adalah banyaknya calo liar yang menjual tiket dengan harga sangat tinggi. Namun demi membela klubnya membutuhkan tiket tersebut, *supporter* akan membelinya walaupun dengan terpaksa.

Kebutuhan dan antusiasme suporter yang sangat besar masih belum didukung dengan fasilitas yang memadai, misalkan sistem penjualan tiket dan sistem pemeriksaan tiket masuk ke stadion. Saat ini penjualan tiket dilakukan dengan cara penyebaran melalui koordinator wilayah (korwil), radio, dan tiket boks. Tetapi hal yang terjadi adalah calo akan memborong tiket dan menjual kembali dengan harga tinggi. Sedangkan karena sistem pemeriksaan tiket yang masih manual, yaitu dengan menggunakan tenaga operator, tiket

yang berupa selembar kertas dapat dengan mudah dipalsukan sehingga sulit untuk dibedakan dengan yang asli. Oleh karena itu, dibutuhkan sentuhan teknologi untuk mengatasi hal-hal tersebut.

Dalam beberapa tahun terakhir ini teknologi identifikasi berbasis frekuensi radio (*Radio Frequency Identification*) berkembang dengan pesat. Hal ini diakibatkan oleh beberapa hal, salah satu diantaranya kebutuhan yang besar dari aplikasi untuk konsumen dengan menggunakan teknologi ini. *RFID* ternyata mampu menjadi salah satu alternatif solusi untuk mengatasi kelemahan-kelemahan pada sistem penjualan dan pemeriksaan tiket pertandingan sepak bola di Indonesia.

Dengan latar belakang tersebut, maka perlu dirancang suatu sistem yang dapat memecahkan masalah tersebut. Pada skripsi ini dirancang prototipe kartu berlangganan pertandingan sepak bola memanfaatkan teknologi rfid (*radio frequency identification*) dengan tujuan agar dapat dijadikan acuan dalam realisasi alat yang digunakan untuk mempermudah proses pembelian tiket pertandingan sepak bola.

Tag/kartu RFID digunakan sebagai kartu berlangganan yang diperoleh dengan mendaftar ke pihak penyelenggara pertandingan. Dalam *tag* tersebut terdapat data-data berupa nomor serial yang berbeda pada tiap *tag*, nama, dan jadwal pertandingan yang telah dipilih oleh pelanggan untuk ditonton. Jadwal tersebut dipilih sesuai keinginan pelanggan saat mendaftar pada manajemen klub dan pelanggan diharuskan untuk membayar tiket pertandingan saat itu juga. Sistem aktif saat pelanggan mendekati *tag* dengan *reader* RFID, sehingga *reader* dapat mengidentifikasi nomor serial pada *tag* tersebut. Kemudian data yang telah diidentifikasi diolah dan sistem akan mengecek nomor serial pada *tag* RFID telah terdaftar pada pertandingan tersebut atau tidak. Jika telah terdaftar palang pintu masuk akan terbuka dan LCD akan menampilkan nomor tempat duduk kemudian tiket segera dicetak menggunakan printer. Pintu akan tertutup secara otomatis setelah terbuka selama 5 detik. Pada hasil cetak tiket terdapat informasi berupa nomor tempat duduk dan jadwal pertandingan lainnya yang telah didaftarkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, maka rumusan masalah dalam skripsi ini ditekankan pada:

- 1) Bagaimana memanfaatkan *tag* RFID dan *RFID reader* sebagai masukan mikrokontroler pada sistem kartu berlangganan pertandingan sepak bola.

- 2) Bagaimana merancang antarmuka mikrokontroler pada sistem kartu berlangganan pertandingan sepak bola.
- 3) Bagaimana merancang perangkat lunak pada mikrokontroler sebagai media utama pengendali sistem.

1.3 Batasan Masalah

Agar dalam perancangan alat ini dapat sistematis dan terarah, maka ada batasan masalah yaitu:

- 1) Lebih ditekankan membahas sistem kartu berlangganan, bukan pada sistem pendaftaran dan pengisian saldo secara mendetail.
- 2) Menggunakan RFID *reader* dan *tag* RFID yang telah jadi dan tidak membahas rangkaian di dalamnya.
- 3) Tidak membahas teori-teori frekuensi radio.
- 4) Kartu berlangganan hanya berlaku untuk tribun kelas VVIP.

1.4 Tujuan

Tujuan skripsi ini adalah merancang dan membuat prototipe kartu berlangganan pertandingan sepak bola di Indonesia, sehingga pelanggan mendapatkan kemudahan dalam membeli tiket pertandingan, mengurangi pemalsuan tiket pertandingan yang beredar dan juga mengurangi jumlah calo.

1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penulisan skripsi ini adalah sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, ruang lingkup, tujuan, dan sistematika pembahasan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

BAB III Metodologi

Berisi tentang metode-metode yang dipakai dalam melakukan perancangan, pengujian, dan analisis data.

BAB IV Perancangan

Perancangan dan perealisasiian alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja dan realisasi alat.

BAB V Pengujian dan Analisis

Memuat aspek pengujian meliputi penjelasan tentang cara pengujian dan hasil pengujian. Aspek analisis meliputi penilaian atau komentar terhadap hasil-hasil pengujian. Pengujian dan analisis ini terhadap alat yang telah direalisasikan berdasarkan masing-masing blok dan sistem secara keseluruhan.

BAB VI Kesimpulan dan Saran

Memuat intisari hasil pengujian dan menjawab rumusan masalah serta memberikan rekomendasi untuk perbaikan kualitas penelitian dimasa yang akan datang.



BAB II

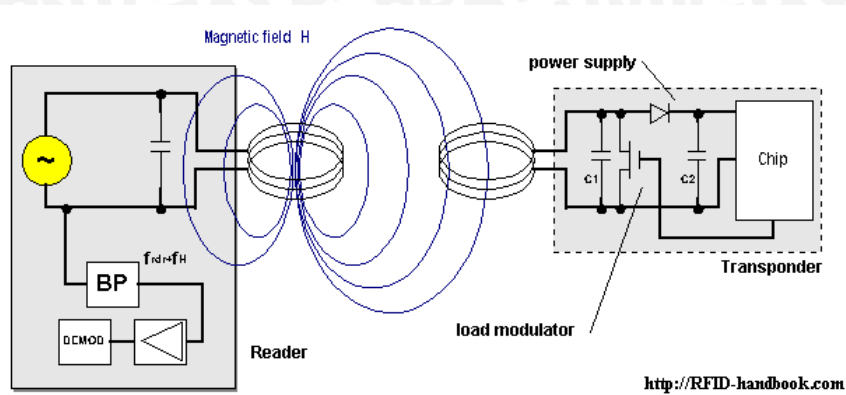
TINJAUAN PUSTAKA

Indonesia merupakan negara dengan antusiasme terhadap sepak bola yang besar. Setiap klub dari liga domestik memiliki pendukung yang fanatik. Kekompakan dan loyalitas mereka sudah tidak dipertanyakan lagi. Buktinya banyak kelompok suporter yang dianugerahi sebagai "Suporter Terbaik". Tetapi kebutuhan dan antusiasme suporter yang sangat besar masih belum didukung dengan fasilitas yang memadai, misalkan sistem penjualan tiket dan sistem pemeriksaan tiket masuk ke stadion.

Penjualan tiket biasanya dilakukan dengan cara menyebarkan melalui koordinator wilayah (korwil), radio, dan tiket boks. Akan tetapi biasanya calo akan memborong tiket dan menjual kembali dengan harga tinggi. Sedangkan karena sistem pemeriksaan tiket yang masih manual, yaitu dengan menggunakan tenaga operator, tiket yang berupa selembar kertas dapat dengan mudah dipalsukan sehingga sulit untuk dibedakan dengan yang asli. Oleh karena itu, dibutuhkan sentuhan teknologi untuk mengatasi hal-hal tersebut.

2.1 **RFID (*Radio Frequency Identification*)**

RFID adalah proses identifikasi seseorang atau objek dengan menggunakan frekuensi transmisi radio. RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah devais kecil yang disebut tag atau *transponder* (*Transmitter + Responder*). Tag RFID akan mengirimkan informasi yang ada di dalam dirinya pada saat berada di dalam gelombang elektromagnetik yang dipancarkan devais yang kompatibel yaitu *pembaca RFID* (RFID Reader). Gelombang elektromagnetik tersebut digunakan sebagai catu daya untuk membangkitkan transmitter sehingga informasi di dalam *tag* dapat terbaca. Jarak pembacaan antara *tag* dan *reader* berkisar ± 12 cm (tergantung jenis tag yang digunakan) serta bekerja pada frekuensi 125 kHz. Hal tersebut ditunjukkan dalam Gambar 2.1.



Gambar 2.1 Komunikasi Antara Reader dan Transmitter (Tag)
Sumber: <http://RFID-handbook.com>

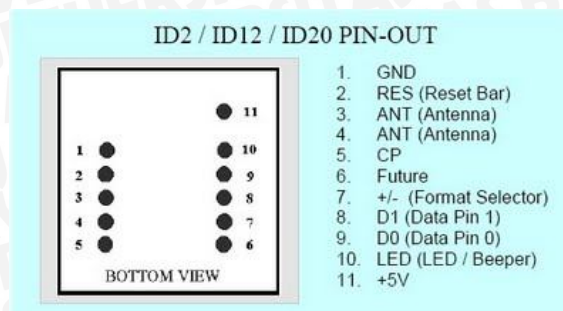
RFID dapat disediakan dalam devais yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read/Write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan. Karena teknologi ini sulit untuk dipalsukan, maka RFID dapat menyediakan keamanan yang tinggi. Pada perindustrian, RFID bisa sangat membantu mulai dari proses manufacturing, distribusi, penyimpanan, hingga penjualan, barang dapat dilacak.

Pada sistem RFID umumnya, *tag* atau *transponder* ditempelkan pada suatu objek atau subjek. Setiap *tag* dapat membawa informasi yang unik, diantaranya: nomor serial, nama, tujuan, kelas, dan data lain dari objek atau subjek tersebut. Ketika *tag* ini melalui medan yang dihasilkan oleh RFID *reader* yang kompatibel, *tag* akan mentransmisikan informasi yang ada pada *tag* kepada RFID *reader*, sehingga proses identifikasi objek dapat dilakukan.

Sistem RFID terdiri atas empat komponen, yaitu:

- Tag*: Ini adalah devais yang menyimpan informasi untuk identifikasi objek. *Tag RFID* disebut juga *transponder*. Tipe *tag RFID* yang digunakan pada perancangan ini adalah ISO Card GK4001.
- Antena: untuk mentransmisikan sinyal frekuensi radio antara pembaca RFID dengan *tag RFID*.
- Pembaca RFID: adalah devais yang kompatibel dengan *tag RFID* yang akan berkomunikasi secara *wireless* dengan *tag*. Tipe yang digunakan adalah ID-12.
- Software* Aplikasi: adalah aplikasi pada sebuah *workstation* atau PC yang dapat membaca data dari *tag* melalui *RFID reader*. Baik *tag* dan *RFID reader* dilengkapi dengan antena sehingga dapat menerima dan memancarkan gelombang elektromagnetik.

RFID reader tipe ID-12 memiliki 11 pin. Konfigurasi pin RFID reader ditunjukkan dalam Gambar 2.2 dan Tabel 2.1 menunjukkan fungsi serta format data keluarannya.



Gambar 2.2 Konfigurasi Pin ID-12 (RFID Reader)

Sumber: <http://panduwicaksana.blogspot.com>

Tabel 2.1 Fungsi dan Format Data Keluaran

Pin No.	Description	ASCII	Magnet Emulation	Wiegand26
Pin 1	Zero Volts and Tuning Capacitor Ground	GND 0V	GND 0V	GND 0V
Pin 2	Strap to +5V	Reset Bar	Reset Bar	Reset Bar
Pin 3	To External Antenna and Tuning Capacitor	Antenna	Antenna	Antenna
Pin 4	To External Antenna	Antenna	Antenna	Antenna
Pin 5	Card Present	No Functions	Card Present	No Functions
Pin 6	Future	Future	Future	Future
Pin 7	Format Selector (+/-)	Strap to GND	Strap to pin 10	Strap to +5V
Pin 8	Data 1	CMOS	Clock	One Output
Pin 9	Data 0	TTL Data	Data	Zero Output
Pin 10	3.1 KHz Logic	Beeper/Led	Beeper/Led	Beeper/Led
Pin 11	DC Voltage Supply	+5V	+5V	+5V

Sumber: <http://panduwicaksana.blogspot.com>

2.2 Mikrokontroler ATmega162

Sebagai sentral/pusat dari pengontrolan sistem ini digunakan mikrokontroler ATmega162. Mikrokontroler ini yang bertugas membaca data dari RFID reader yaitu format ASCII dan mengolah data tersebut sehingga menjadi format desimal.

ATmega162 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan. ATmega162 memiliki fasilitas komunikasi serial untuk melakukan pengiriman/penerimaan data dengan komputer. IC jenis ini berorientasi pada kontrol yang dapat diprogram ulang. Mikrokontroler ATmega162 mempunyai karakteristik utama sebagai berikut:

- CPU dengan lebar data 8 bit.
- Empat buah I/O 8 bit dan sebuah I/O 3 bit.
- Memori Flash sebesar 16 kilo *byte*.

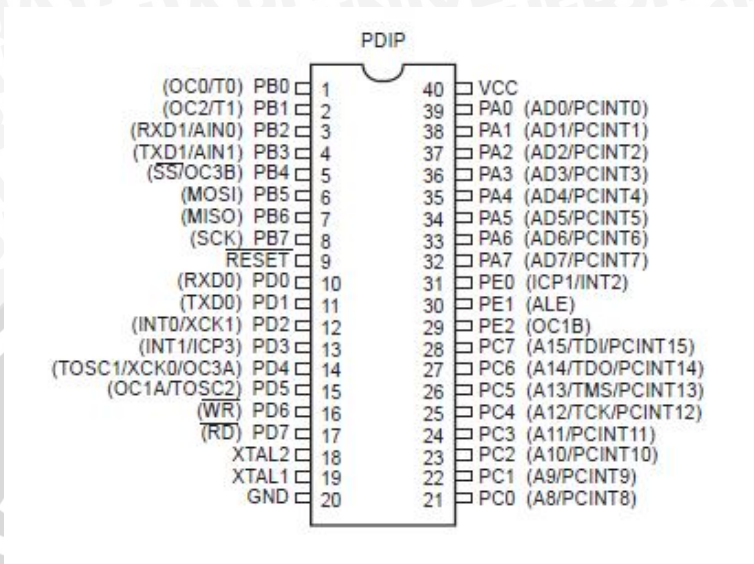
- SRAM sebesar 1 kilo *byte*.
- EEPROM sebesar 512 *byte* untuk memori program pada *chip*.
- Dua buah *timer/counter* 8 bit dan 16 bit.
- Enam buah saluran PWM.
- Dua buah *port* USART untuk komunikasi serial.
- Interupsi internal dan eksternal.

Masing-masing kaki dalam mikrokontroler ATmega162 mempunyai fungsi tersendiri. Dengan mengetahui fungsi masing-masing kaki mikrokontroler ATmega162, perancang aplikasi mikrokontroler ATmega162 akan lebih mudah merencanakan dan membuat sistem yang dirancang. ATmega162 mempunyai 40 pin, susunan masing-masing pin ditunjukkan dalam Gambar 2.3.

Fungsi pin-pin secara keseluruhan dari mikrokontroler ATmega162 adalah sebagai berikut:

- Port A (Pin A0-7), berfungsi sebagai masukan analog ke ADC internal pada mikrokontroler ATmega162, selain itu juga berfungsi sebagai port I/O dwi-arah 8-bit, jika ADC-nya tidak digunakan. Masing-masing pin menyediakan resistor *pull-up* internal yang bisa diaktifkan untuk masing-masing bit.
- Port B (Pin B0-7), berfungsi sebagai port I/O dwi-arah 8-bit. Masing-masing pin menyediakan resistor *pull-up* internal yang bisa diaktifkan untuk masing-masing bit. Port B juga memiliki pin fungsi khusus, yaitu SPI, Timer/Counter, komparator analog, dan USART1.
- Port C (Pin C0-7), berfungsi sebagai port I/O dwi-arah 8-bit. Masing-masing pin menyediakan resistor *pull-up* internal yang bisa diaktifkan untuk masing-masing bit. Port C juga memiliki pin fungsi khusus, yaitu JTAG, PCINT.
- Port D (Pin D0-7), berfungsi sebagai port I/O dwi-arah 8-bit. Masing-masing pin menyediakan resistor *pull-up* internal yang bisa diaktifkan untuk masing-masing bit. Port D juga memiliki pin fungsi khusus, yaitu *Timer/Counter*, interupsi eksternal, dan USART0.
- Port E (Pin E0-2), berfungsi sebagai port I/O dwi-arah 3-bit. Port E memiliki pin fungsi khusus, yaitu *Timer/Counter*, interupsi eksternal.
- /RESET, merupakan masukan reset. Level rendah pada pin ini selama lebih dari lama waktu minimum yang ditentukan akan menyebabkan reset, walaupun *clock* tidak dijalankan.

- XTAL1, merupakan masukan ke penguat osilator terbalik (*inverting*) dan masukan ke rangkaian *clock* internal.
- XTAL2, merupakan luaran dari penguat osilator terbalik



Gambar 2.3 Konfigurasi Pin Mikrokontroler ATmega162
Sumber: Atmel, 2009:2

Mikrokontroler ATmega162 memiliki dua buah *port* USART untuk komunikasi serial, yaitu USART0 and USART1. USART0 dan USART1 digunakan untuk komunikasi antara ATmega162 dengan RFID dan komputer. Fasilitas komunikasi serial USART Mikrokontroler ini memiliki fitur sebagai berikut:

- Komunikasi *full-duplex* dengan register serial untuk penerima dan pengirim data.
- Dapat dioperasikan pada mode komunikasi sinkron dan asinkron.
- Pada operasi sinkron *clock* berasal dari *master* atau *slave*.
- Mempunyai resolusi tinggi untuk generator *baudrate*.
- Layanan pengiriman data terdiri dari 5, 6, 7, 8 dan 9 bit dan 1 atau 2 bit stop.
- Paritas genap atau ganjil dan didukung dengan pengecekan paritas oleh *hardware*.
- Pendeteksi pengiriman kelebihan data.
- Pendeteksi kesalahan pada format data yang dikirim.
- Memiliki filter *noise* yang terdiri atas pendeteksi kesalahan bit start dan low pass filter.
- Memiliki 3 layanan *interrupt* yaitu TX *complate*, TX *data empty*, dan RX *complate*.
- Mode komunikasi *multi processor*.
- Mode komunikasi asinkron dengan dua kecepatan.

Untuk menghitung baudrate komunikasi serial digunakan rumus dalam Tabel 2.2.

Tabel 2.2 Rumus Menghitung Baudrate

Operating Mode	Equation for Calculating Baud Rate ⁽¹⁾	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{osc}}{16(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{osc}}{8(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{osc}}{2(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2BAUD} - 1$

Sumber: Atmel, 2009:169

Keterangan:

BAUD = Baudrate (dalam *bits per second*, bps)

f_{osc} = Frekuensi *clock* dari sistem osilator

UBRR = *Register baudrate* yang terdiri dari UBRRH dan UBRRL

2.3 Motor DC

2.3.1 Prinsip Kerja Motor DC

Segulung kawat yang dialiri arus listrik dan ditempatkan di dalam suatu medan magnet akan mengalami gaya yang sebanding dengan arus dan kekuatan medan magnetnya. Gaya yang ditimbulkan disebut dengan Gaya *Lorentz* [Soemarwanto, 1997:28] yang dapat dirumuskan sebagai berikut:

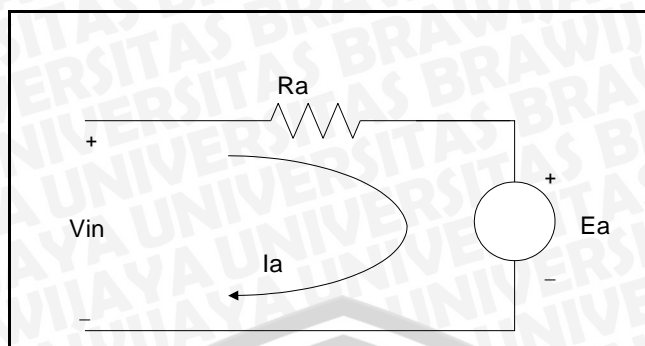
$$F = B I L \quad (2.1)$$

Dalam hal ini B adalah kerapatan *fluks* magnet, I adalah arus yang mengalir dan L adalah panjang kawat.

Persamaan 2.1 merupakan prinsip dasar sebuah motor, dimana terjadinya proses perubahan energi listrik (I) akan menimbulkan gaya mekanik (F). Jika motor mempunyai jari-jari sebesar r , maka akan menimbulkan torsi sebesar:

$$\Gamma = F.r = B.I.L.r \quad (2.2)$$

Pada saat dibangkitkan, konduktor akan bergerak di dalam medan magnet dan akan menimbulkan gaya gerak listrik (ggl) yang merupakan reaksi lawan terhadap tegangan penyebabnya. Proses konversi energi listrik menjadi energi mekanik dapat berlangsung jika tegangan sumber lebih besar dari gaya gerak listrik lawan.



Gambar 2.4 Gambar Motor Secara Umum
Sumber: Fitzgerald, 1992 : 269

Motor dapat berputar jika tegangan masukan motor lebih besar dari ggl yang timbul. Hubungan antara tegangan sumber dan ggl lawan seperti ditunjukkan dalam Gambar 2.4, dapat dirumuskan sebagai berikut:

$$E_a = V_{in} - I_a.R_a \quad (2.3)$$

Dalam hal ini E_a adalah tegangan pada jangkar, V_{in} adalah tegangan masukan, I_a adalah arus jangkar dan R_a adalah tahanan jangkar, sedangkan induksi yang timbul adalah:

$$E_a = C n \Phi \quad (2.4)$$

Dengan C adalah konstanta, n adalah kecepatan motor, dan Φ adalah fluks magnetik yang besarnya sebanding dengan arus penguatan torsi. Torsi pada motor juga sebanding dengan fluks magnetik dan arus. Hal ini ditunjukkan pada Persamaan 2.5 berikut:

$$\Gamma = C \Phi I_a \quad (2.5)$$

Jika diketahui kecepatan sudut ω adalah:

$$\omega = 2\pi \frac{n}{60} \quad (2.6)$$

Maka hubungan torsi dan kecepatan motor adalah:

$$\Gamma = \frac{P}{\omega} \quad (2.7)$$

$$\Gamma = \frac{P}{2\pi \frac{n}{60}}$$

Dengan P adalah daya motor.

2.3.2 Pengaturan Motor DC

Apabila Persamaan 2.3 disubstitusikan dengan Persamaan 2.4 akan didapatkan rumus kecepatan motor (n) sebagai berikut:

$$n = \frac{V_{in} - I_a R_a}{C\Phi} \quad (2.8)$$

Dari Persamaan 2.8 terlihat bahwa kecepatan motor sebanding dengan tegangan masukan (V_{in}) [Zuhal, 1993: 91]. Jadi apabila tegangan masukan besar maka kecepatan motor akan cepat, demikian pula sebaliknya, jika tegangan masukan kecil maka kecepatan motor akan lambat.

2.4 LCD (*Liquid Crystal Display*)

Liquid Crystal Display (LCD) merupakan komponen elektronika yang digunakan untuk menampilkan karakter baik berupa karakter angka, huruf, atau karakter lainnya, sehingga tampilan tersebut dapat dilihat secara visual. Pemakaian LCD sebagai indikator tampilan banyak digunakan, karena daya yang digunakan LCD relatif kecil.

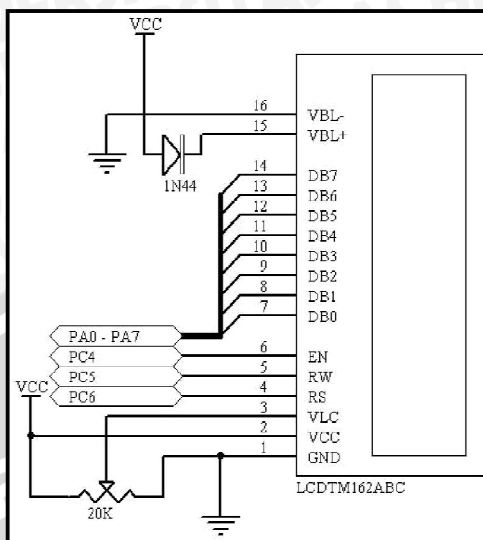
LCD terdiri atas tumpukan tipis atau sel dari dua lembar kaca dengan pingsiiran tertutup rapat. Antara dua lembar kaca tersebut diberi bahan kristal cair (*liquid crystal*) yang tembus cahaya. Permukaan luar masing-masing keping kaca mempunyai lapisan tembus cahaya seperti oksida timah (*tin oxide*) atau oksida indium (*indium oxide*).

Pada perancangan sistem ini memakai LCD modul M1632 yang merupakan sebuah modul LCD dot matrik yang membutuhkan daya kecil. LCD modul M1632 dilengkapi panel LCD dengan tingkat kontras yang cukup tinggi serta pengendali LCD CMOS yang telah terpasang dalam modul tersebut. LCD modul M1632 mempunyai spesifikasi sebagai berikut:

- Memiliki 16 karakter dan 2 baris tampilan yang terdiri atas 5 X 7 dot matrik ditambah dengan kursor.
- Memerlukan catu daya +5 volt.
- Otomatis reset saat catu daya dinyalakan.
- Display data RAM 80 X 8 (max 80 karakter).
- Menggunakan 4 bit data dan 3 bit control.

LCD ini mempunyai konsumsi daya yang relatif rendah, dan terdapat sebuah controller CMOS di dalamnya. Controller tersebut sebagai pembangkit dari karakter ROM/RAM dan display data RAM. Semua fungsi tampilan dikontrol oleh suatu instruksi dan modul LCD dapat dengan mudah untuk di-*interface*-kan dengan mikrocontroller. Masukan yang diperlukan untuk mengendalikan modul ini berupa bus data yang masih *multiplex* dengan bus alamat serta 3 bit sinyal kontrol. Sementara pengendalian dot matrix LCD dilakukan secara internal oleh controller yang sudah ada pada modul LCD.

Gambar 2.5 menunjukkan rangkaian *interface* ke LCD Karakter 2x16. dan Tabel 2.3 menunjukkan pin-pin I/O LCD.



Gambar 2.5 Rangkaian *Interface* ke LCD Karakter 2x16

Tabel 2.3 Pin-pin I/O LCD

No	Simbol	Level	Fungsi
1	Vss		GND
2	Vcc		Power Supply 5 volt
3	Vee		LCD drive
4	RS	H/L	H: Data Input, L: Ins Input
5	R/W	H/L	H: Read,L: Write
6	E	H	Enable Signal
7	DB0	H/L	Data Bus
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	
15	V+BL		Power Supply 4-4.2V
16	V-BL		GND

Sumber: LMB 16 Specification

2.5 Borland Delphi 7

Borland Delphi atau biasa disebut Delphi merupakan perangkat lunak pengembangan aplikasi yang sangat populer di lingkungan Linux maupun Windows. Delphi merupakan bahasa pemrograman yang mempunyai cakupan kemampuan yang luas dan canggih. Berbagai jenis aplikasi dapat dibuat dengan Delphi, termasuk aplikasi untuk mengolah teks, grafik, angka, *database*, dan aplikasi web.

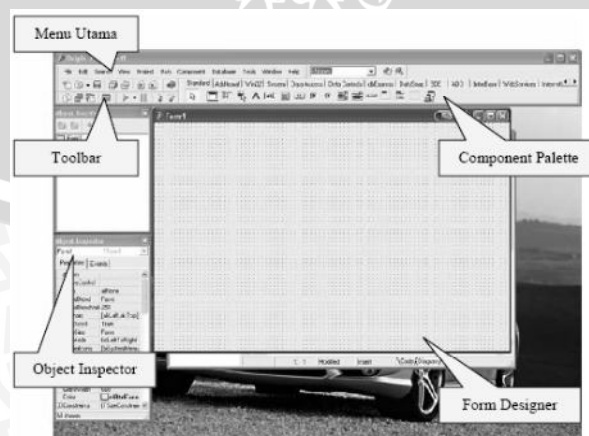
Secara umum, kemampuan Delphi adalah menyediakan komponen-komponen dan bahasa pemrograman yang handal, sehingga memungkinkan untuk membuat program aplikasi sesuai dengan keinginan, dengan tampilan dan kemampuan yang canggih.

Untuk mempermudah pemrogram dalam membuat program aplikasi, Delphi menyediakan fasilitas pemrograman yang sangat lengkap. Fasilitas pemrograman tersebut dibagi dalam dua kelompok, yaitu *object* dan bahasa pemrograman. Secara ringkas, *object* adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat (*visual*). *Object* biasanya dipakai untuk melakukan tugas-tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa pemrograman secara singkat dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu. Delphi menggunakan struktur bahasa pemrograman *Object Pascal* yang sudah sangat dikenal di kalangan pemrograman profesional.

Khusus untuk pemrograman *database*, Delphi menyediakan *object* yang kuat, canggih, dan lengkap, sehingga memudahkan pemrogram merancang, membuat, dan menyelesaikan aplikasi *database* yang diinginkan. Delphi dapat menangani data dalam berbagai format *database*, misalnya format MS-Access, Oracle, MySQL, MS SQL, DB2, dan lain-lain.

Delphi dapat digunakan untuk membuat berbagai jenis program, diantaranya aplikasi, *ActiveX*, *WebService*, dan lain-lain. Delphi menyediakan *Integrated Development Environment* (IDE) untuk memudahkan perancangan tampilan untuk pemakai (antarmuka pemakai) secara visual dan penulisan kode.

IDE Delphi menyediakan berbagai jendela yang akan sering dilibatkan dalam pengembangan aplikasi, antara lain menu utama, *SpeedBar*, Jendela Form, *Object Inspector*, dan Komponen *Palette*. Gambar 2.6 menunjukkan tampilan IDE Delphi.



Gambar 2.6 IDE Delphi
Sumber: MADCOMS, 2006: 2

2.6 MySQL

Secara sederhana *database* (basis data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses secara mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data, seperti menambah dan menghapus data.

DBMS singkatan dari *Database Management System*. DBMS merupakan perangkat lunak yang dirancang secara khusus untuk memudahkan pengelolaan *database*. DBMS yang populer dewasa ini berupa RDBMS (*Relational Database Management System*), yang menggunakan model basis data relasional atau dalam bentuk tabel-tabel yang saling terhubung.

MySQL merupakan salah satu contoh produk RDBMS yang sangat populer di lingkungan Linux, tetapi juga tersedia pada Windows. MySQL merupakan *database server* yang dikembangkan oleh sebuah perusahaan Swedia yang bernama MySQL AB, yang pada saat itu bernama T.c.X DataKonsult AB, di mulai sejak sekitar tahun 1994-1995.

Banyak kelebihan yang dimiliki oleh MySQL, antara lain:

- MySQL merupakan *database* yang memiliki kecepatan yang tinggi dalam melakukan pemrosesan data, dapat diandalkan, dan mudah digunakan serta mudah dipelajari. Hal ini disebabkan MySQL telah banyak digunakan di belahan bumi manapun sehingga jika kita mempunyai masalah dengan *database* tersebut, kita dapat bertanya kepada banyak orang (pengguna yang lain) melalui internet maupun orang di sekitar kita yang siap membantu menyelesaikan masalah tersebut serta dukungan manual maupun referensi yang banyak bertebaran di internet.
- MySQL mendukung banyak bahasa pemrograman seperti C, C++, *Perl*, *Python*, *Java*, dan PHP. Bahasa pemrograman tersebut dapat digunakan untuk berinteraksi maupun berkomunikasi dengan *MySQL server*, atau dapat juga digunakan sebagai komponen pembentuk antarmuka (*interface*) dari suatu *database* MySQL.
- Koneksi, kecepatan, dan keamanan membuat MySQL sangat cocok diterapkan untuk mengakses *database* melalui internet, dengan menggunakan bahasa pemrograman *Perl* atau PHP sebagai *interface*-nya.
- MySQL dapat melakukan koneksi dengan *client* menggunakan *protocol* TCP/IP, *Unix socket* (Unix), atau *Named Pipes* (NT).
- MySQL dapat menangani *database* dengan skala yang sangat besar dengan jumlah *record* mencapai lebih dari 50 juta, dapat menampung 60 ribu tabel, dan juga bisa

menampung 5 milyar baris data. Selain itu, batas indeks pada tiap table dapat menampung mencapai 32 indeks.

- Dalam hal relasi antartabel pada suatu *database*, MySQL menerapkan metode yang sangat cepat, yaitu dengan menggunakan metode *one-sweep multijoin*. MySQL sangat efisien dalam mengelola informasi yang kita minta yang berasal dari banyak tabel sekaligus.
- *Multiuser*, yaitu dalam satu *database server* pada MySQL dapat diakses oleh beberapa user dalam waktu yang sama tanpa mengalami konflik atau *crash*.
- *Security* yang dimiliki *database* MySQL dikenal baik, karena memiliki lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan system perizinan yang khusus serta *password* yang dimiliki setiap *user* dalam bentuk data terenkripsi.
- MySQL merupakan *software database* yang bersifat *free* atau gratis, jadi kita tidak perlu susah-susah mengeluarkan biaya untuk membayar lisensi kepada pembuat *software*. Hal ini sangat berbeda jika kita menggunakan *software database* seperti IBM DB2 ataupun Oracle, karena kita harus membayar mahal untuk mendapatkan lisensinya.

2.6.1. Administrasi User

Administrasi *user* adalah pengolahan *user* sehingga dapat menggunakan MySQL. Untuk itu diperlukan adanya *account*, yang didefinisikan sebagai suatu identitas unik dari *user*. Identitas inilah yang memungkinkan untuk melakukan koneksi ke *database server* dan memberikan operasi-operasi *database*.

Secara garis besar, *user* di dalam MySQL dikategorikan menjadi dua bagian, yaitu super *user* (*root*) dan *user*. Super *user* bertindak sebagai administrator yang bertanggung jawab terhadap segala administrasi sistem MySQL, termasuk salah satunya adalah mengelola *user-user*. Kategori kedua adalah *user* 'biasa', yaitu *user* yang dapat menggunakan *database* di dalam MySQL, sesuai dengan hak akses miliknya.

2.6.2. Merancang Database

Database atau basis data merupakan koleksi data terstruktur yang diletakkan di dalam suatu komputer. Di dalam aspek kehidupan, *database* memegang peranan yang sangat penting sekali, terutama di bidang aktivitas komputasi. Banyak sekali perangkat

lunak yang memanfaatkan *database* sebagai jantungnya, baik itu perangkat lunak berskala kecil maupun besar, berbasis web maupun non-web, dan sebagainya.

Database adalah suatu kumpulan data yang saling berhubungan dan terorganisasi sedemikian rupa hingga mudah untuk digunakan kembali. *Database* merupakan salah satu komponen yang penting sekali dalam sistem informasi, karena merupakan dasar untuk menyediakan informasi bagi para pemakai.

❖ **DDL (Data Definiton Language)**

DDL adalah sebuah Metode *Query* SQL yang berguna untuk mendefinisikan data pada sebuah *database*, adapun *Query* yang dimiliki adalah:

- *CREATE* : Digunakan untuk melakukan pembuatan tabel dan *database*.
- *DROP* : Digunakan untuk melakukan penghapusan tabel maupun *database*.
- *ALTER* : Digunakan untuk melakukan pengubahan struktur tabel yang telah dibuat baik menambah *Field* (*add*), mengganti nama *Field* (*change*) ataupun menamakan kembali (*rename*), serta menghapus (*drop*).

❖ **DML (Data Manipulation Language)**

DML adalah sebuah metode *Query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *Query* ini adalah untuk melakukan pemanipulasian *database* yang telah ada atau telah dibuat sebelumnya. Adapun *Query* yang termasuk di dalamnya adalah:

- *INSERT*: Digunakan untuk melakukan penginputan atau pemasukan data pada tabel *database*.
- *UPDATE*: Digunakan untuk melakukan pengubahan atau peremajaan terhadap data yang ada pada tabel.
- *DELETE*: Digunakan untuk melakukan penghapusan data pada tabel. Penghapusan ini dapat dilakukan secara sekaligus (seluruh isi tabel) maupun hanya beberapa *Recordset*.

Kedua bentuk *Query* SQL tersebut dapat digunakan pada semua model basis data yang mendukungnya, baik yang berbasis Unix maupun Windows misalnya, anda dapat mengimplementasikannya pada program penghasil *database* seperti Microsoft Acces atau pada semua *database* bawaan program visual seperti Database Desktop yang bernama Paradox.

❖ Membuat Database

Database di dalam MySQL diimplementasikan sebagai sesuatu direktori berisi *file-file* yang sesuai dengan tabel pada *database*. Pembuatan *database* dilakukan menggunakan pernyataan *CREATE DATABASE* dan diikuti dengan nama *database* yang akan dibuat. Oleh karena tidak ada tabel-tabel pada saat suatu *database* diciptakan, maka pernyataan untuk membuat *database* hanya menciptakan direktori kosong sesuai dengan nama yang diberikan.

Nama *database* di dalam MySQL dapat ditulis dengan panjang maksimal 64 byte. Semua karakter diperbolehkan untuk memberikan nama *database*, kecuali tanda slash (/), *backslash* (\), dan titik. Hal ini berarti bahwa diharuskan mengawali nama *database* dengan angka, akan tetapi tidak boleh semuanya berupa angka. Sintaks dari pernyataan *CREATE DATABASE* di dalam MySQL adalah seperti berikut.

```
create database [if not exists] nama_database
```

❖ Membuat Tabel

Tabel merupakan salah satu objek yang ada pada *database*, oleh sebab itu di dalam pembuatan tabel harus memilih *database* terlebih dahulu dan menjadikannya sebagai *database* aktif. Jika tidak memilih *database*, perintah tidak akan dilaksanakan dengan baik, karena MySQL tidak mengetahui akan diletakkan pada *database* mana tabel baru tersebut.

Pembuatan tabel pada MySQL dapat dilakukan melalui beberapa cara, misalnya dengan memberikan perintah langsung pada *command prompt* atau mengeksekusi *file* yang berisi pernyataan SQL pembuatan tabel.

Pembuatan *Field* atau kolom dalam *database* sebenarnya memiliki dua kondisi yaitu harus dimasuki data atau bisa dikosongkan. Dalam hal ini kolom yang diisi tidak boleh kosong atau NOT NULL.

Dalam pembuatan *database*, *record* yang datanya tidak boleh sama dengan *record* yang lain disebut *primary key* atau kunci primer. Untuk membuat sebuah kunci primer, hanya boleh dibuat satu kali, jadi kunci primer sebenarnya diciptakan untuk menjadi kolom utama sebuah tabel.

Dengan bertambahnya popularitas MySQL, banyak *tool* yang dikembangkan untuk memudahkan proses manajemen agar *database* MySQL lebih mudah digunakan. Pada saat ini sudah banyak MySQL *client* yang berbasis GUI (*Graphical UserInterface*). MySQL-Front merupakan salah satu *software* yang dibuat untuk antarmuka *database* MySQL.

MySQL-Front memungkinkan kita untuk manajemen *database* MySQL dengan mudah melalui antarmuka Windows. Ini lebih mudah digunakan daripada kita menggunakan *command line* atau *console* bawaan dari MySQL.

MySQL-Front adalah *tool* yang dibuat menggunakan Delphi yang merupakan salah satu pemrograman berbasis visual sehingga dapat menampilkan tampilan grafik yang cukup baik. Kelebihan *tool* ini adalah kemampuannya untuk mengakses ODBC (*Open Database Connectivity*, protokol standar yang digunakan untuk mengakses informasi antar-*database* pada *platform* Microsoft Windows). Namun *tool* ini hanya berjalan pada *platform* Windows, sistem operasi yang menjadi sponsor utama ODBC.

MySQL-Front banyak mendukung fungsi-fungsi dari MySQL, diantaranya adalah *create/drop databases*, *create/drop tables*, *edit/add/delete fields*, *edit/insert/delete record*, *view* dan *kill* untuk suatu proses pada pengguna lain, mengeksekusi SQL *scripts* yang cukup besar, menampilkan properti tabel secara *advanced*, *replication*, *copy* tabel ke nama tabel yang baru, manajemen *user*, *flush host/logs/privileges/table*. Kita juga dapat menggunakan MySQL-Front untuk menulis SQL *query* dengan sintaks yang kompleks dan *export* serta *import* data dari ODBC *datasources*.

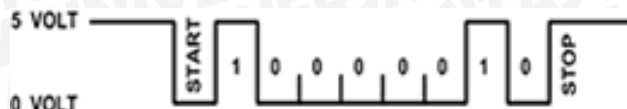
2.6 Komunikasi Serial

Ada 2 macam cara komunikasi data serial yaitu Sinkron dan Asinkron. Pada komunikasi data serial sinkron, clock dikirimkan bersama sama dengan data serial, tetapi clock tersebut dibangkitkan sendiri-sendiri baik pada sisi pengirim maupun penerima. Sedangkan pada komunikasi serial asinkron tidak diperlukan clock karena data dikirimkan dengan kecepatan tertentu yang sama baik pada pengirim/penerima.

Pada UART, kecepatan pengiriman data (atau yang sering disebut dengan Baud Rate) dan fase clock pada sisi transmitter dan sisi receiver harus sinkron. Untuk itu diperlukan sinkronisasi antara Transmitter dan Receiver. Hal ini dilakukan oleh bit “Start” dan bit “Stop”. Ketika saluran transmisi dalam keadaan idle, output UART adalah dalam keadaan logika “1”. Ketika Transmitter ingin mengirimkan data, output UART akan diset dulu ke logika “0” untuk waktu satu bit. Sinyal ini pada receiver akan dikenali sebagai sinyal “Start” yang digunakan untuk menyinkronkan fase clocknya sehingga sinkron dengan fase clock transmitter.

Selanjutnya data akan dikirimkan secara serial dari bit yang paling rendah (bit 0) sampai bit tertinggi. Selanjutnya akan dikirimkan sinyal “Stop” sebagai akhir dari pengiriman data serial.

Sebagai contoh misalnya akan dikirimkan data huruf “A” dalam format ASCII (atau sama dengan 41 heksa atau 0100 0001). Contoh tersebut ditunjukkan dalam Gambar 2.7.

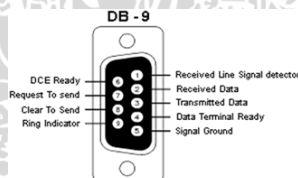


Gambar 2.7 Pengiriman Huruf “A” Tanpa bit Paritas
Sumber: www.ittelkom.ac.id

Kecepatan transmisi (baud rate) dapat dipilih bebas dalam rentang tertentu. Baud rate yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400, dan 9600 (bit/perdetik). Dalam komunikasi data serial, baud rate dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama. Selanjutnya harus ditentukan panjang data (6,7 atau 8 bit), paritas (genap, ganjil, atau tanpa paritas), dan jumlah bit “Stop” (1, 1 ½, atau 2 bit).

Konfigurasi Port Serial

Konektor DB-9 pada bagian belakang komputer adalah port serial RS232 yang biasa dinamai dengan COM1 dan COM2. Konfigurasi port serial ditunjukkan dalam Gambar 2.8.



Gambar 2.8 Konfigurasi Port Serial
Sumber: www.ittelkom.ac.id

Tabel 2.4 menunjukkan keterangan pin-pin port serial.

Tabel 2.4 Keterangan Pin-Pin Port Serial

No. Pin	Nama Sinyal	Direction	Keterangan
1	DCD	In	Data Carrier Detect/Receive Line Signal Detect
2	RxD	In	Receive Data
3	TxD	Out	Transmitt Data
4	DTR	Out	Data Terminal Ready
5	GND	-	Ground
6	DSR	In	Data Set Ready
7	RST	Out	Request to Send
8	CTS	In	Clear to Send
9	RI	In	Ring Indicator

Sumber: www.ittelkom.ac.id

BAB III METODOLOGI

Pada bab ini akan dijelaskan metode yang digunakan dalam merancang sebuah prototipe kartu berlangganan pertandingan sepak bola yang mengaplikasikan RFID dalam mengidentifikasi nomor serial pada tiap kartu. Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang dirancang adalah penentuan spesifikasi alat, studi literatur, perancangan dan pembuatan alat, pengujian alat, dan pengambilan kesimpulan.

3.1 Perancangan Alat

Perancangan alat prototipe kartu berlangganan pertandingan sepak bola meliputi tahapan-tahapan sebagai berikut:

- 1) Penentuan spesifikasi alat.
- 2) Pembuatan diagram blok sistem keseluruhan.
- 3) Perancangan perangkat keras masing-masing blok yang meliputi perencanaan dan pembuatan rangkaian dari masing-masing blok.
- 4) Menggabungkan beberapa blok menjadi keseluruhan sistem yang direncanakan.
- 5) Perancangan perangkat lunak komputer dan mikrokontroler untuk mengendalikan sistem secara keseluruhan.

3.2 Pembuatan Alat

Pembuatan alat prototipe kartu berlangganan pertandingan sepak bola Arema Indonesia meliputi tahapan-tahapan sebagai berikut:

- 1) Pembuatan sistem berdasarkan pada perancangan.
- 2) Pembuatan perangkat keras sistem dengan menggunakan komponen elektronika yang telah direncanakan.
- 3) Pembuatan perangkat lunak komputer dan mikrokontroler sesuai dengan diagram alir yang telah direncanakan.

3.3 Pengujian Alat

Untuk mengetahui kinerja piranti apakah sesuai dengan yang direncanakan maka perlu dilakukan pengujian rangkaian pada masing-masing blok berupa perangkat keras beserta perangkat lunaknya dan pengujian secara keseluruhan. Data pengujian dianalisis untuk dijadikan acuan dalam mengambil kesimpulan.

Pengujian alat dilakukan pada masing-masing bagian sesuai dengan blok diagram, untuk mengetahui apakah memenuhi spesifikasi yang telah ditentukan. Pengujian yang dilakukan dibagi menjadi beberapa tahap yaitu:

1. Pengujian program aplikasi sebagai penghubung dengan *database*
2. Pengujian rangkaian RFID dan komunikasi serial RS232
3. Pengujian LCD
4. Pengujian *counter* sebagai penentu nomor tempat duduk
5. Pengujian mencetak data menggunakan *printer*
6. Pengujian rangkaian driver motor DC

Sebagai tahapan yang terakhir adalah pengujian keseluruhan sistem yang bertujuan untuk mengetahui keberhasilan dari keseluruhan sistem yang dirancang. Pada pengujian ini semua sistem digabungkan menjadi satu kesatuan dan dilakukan simulasi seperti sistem pada kartu berlangganan pertandingan sepak bola yang sebenarnya.



BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Penyusunan laporan ini didasarkan pada masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasiian alat agar dapat bekerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Langkah-langkah yang perlu dilakukan untuk merealisasiikan alat yang dirancang adalah penentuan spesifikasi alat, studi literatur, perancangan dan pembuatan alat, pengujian alat, dan pengambilan kesimpulan.

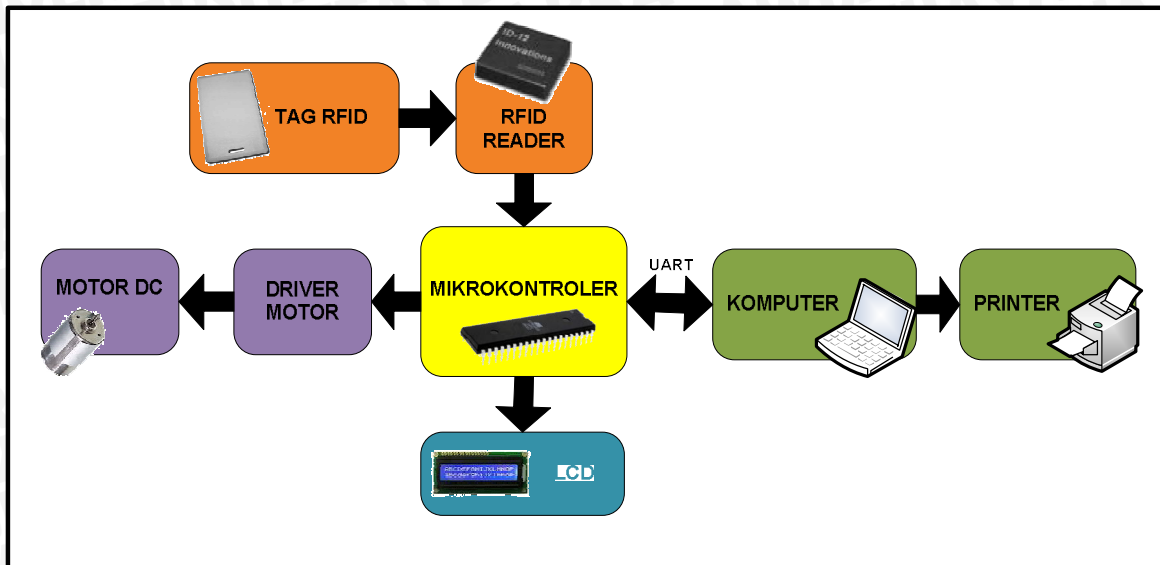
4.1 Penentuan Spesifikasi Alat

Spesifikasi alat secara global ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan yaitu:

1. Menggunakan RFID dengan tipe ID-12 dan tag RFID bertipe ISO Card GK4001.
2. Sebagai pengendali utama sistem digunakan mikrokontroler ATmega162.
3. Motor yang digunakan untuk membuka dan menutup palang pintu adalah motor DC 12V.
4. Menggunakan LCD dan komputer sebagai menu tampilan.
5. Data yang dapat disimpan dalam *database* adalah nomor serial, nama, alamat, nomor telepon, dan jadwal pertandingan yang telah dipesan.
6. Sistem menggunakan catu daya AC 220 V untuk komputer dan DC 12 V untuk motor DC.

4.2 Diagram Blok Sistem

Diagram blok direncanakan untuk memberikan gambaran yang jelas mengenai cara kerja keseluruhan sistem, fungsi blok-blok yang ada di dalamnya serta hubungan antara satu blok dengan blok yang lain. Secara garis besar, diagram blok perancangan *hardware* sistem secara keseluruhan ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram Blok Sistem

Fungsi masing-masing bagian dalam diagram blok ini adalah sebagai berikut:

1. Tag RFID

Tag RFID digunakan sebagai kartu anggota. Fungsinya adalah sebagai identitas ketika akan memesan tiket pertandingan dan untuk membuka palang pintu masuk stadion saat pertandingan berlangsung. Tag RFID didapatkan ketika seseorang mendaftarkan diri pada panitia yang bertanggung jawab atas pertandingan sepak bola tersebut.

2. RFID Reader

Sebagai pembaca *Tag* RFID yang digunakan sebagai data masukan pada Mikrokontroler.

3. Mikrokontroler

Mikrokontroler digunakan sebagai pengolah data masukan dari RFID Reader dan kemudian mengirimkan data hasil pengolahan secara serial ke komputer. Selain itu, mikrokontroler berfungsi sebagai pengendali ON/OFF Motor DC dengan cara memberikan tegangan pada *driver*.

4. Komputer

Komputer berfungsi sebagai bagian menerima data yang dimasukkan user saat proses pemesanan tiket, mengolah data masukan dan mengirimkan data keluaran ke piranti selanjutnya.

5. Printer

Printer berfungsi untuk mencetak nomor tempat duduk dalam format angka dan jadwal pertandingan yang telah dipesan sebelumnya.

6. *Driver*

Driver berfungsi sebagai saklar untuk mengaktifkan dan menonaktifkan motor DC.

7. Motor DC

Motor DC digunakan untuk membuka dan menutup palang pintu masuk stadion.

8. LCD

LCD digunakan untuk menampilkan nomor tempat duduk.

Prinsip kerja dari diagram blok di atas, diawali dengan seorang pelanggan yang mendekati tag RFID dengan Reader RFID. Mikrokontroler akan mengolah data masukan tersebut dan mencocokkan dengan *database* pada komputer. Jika pelanggan telah memesan tiket untuk jadwal pertandingan tersebut, maka *counter* akan aktif dan mencacah angka yang digunakan sebagai nomor tempat duduk. Kemudian nomor tempat duduk dengan format angka tersebut ditampilkan oleh LCD. Setelah itu *printer* akan mencetak data-data yang berupa nomor tempat duduk dan jadwal pertandingan yang telah dibeli. Kemudian motor DC ON sehingga palang pintu terbuka selama 5 detik dan akan tertutup secara otomatis.

4.3 Perancangan Perangkat Keras (*Hardware*)

Perancangan perangkat keras terdiri atas perancangan rangkaian RFID reader, perancangan rangkaian mikrokontroler, perancangan rangkaian *driver* motor, dan perancangan rangkaian LCD.

4.3.1 Perancangan Rangkaian RFID Reader

RFID *Reader* ini digunakan sebagai pembaca *tag* RFID yang terhubung secara serial dengan mikrokontroler ATmega162. RFID *Reader* ini dapat mengirimkan data dengan berbagai format data, yaitu ASCII, *Magnet Emulation*, atau Wiegand26. Namun dalam perancangan ini digunakan keluaran dengan format ASCII, karena keluaran ini mudah untuk dihubungkan pada mikrokontroler ATmega162. Hal ini dilakukan dengan menghubungkan Pin.6 (*Format Selector*) ke *ground*.

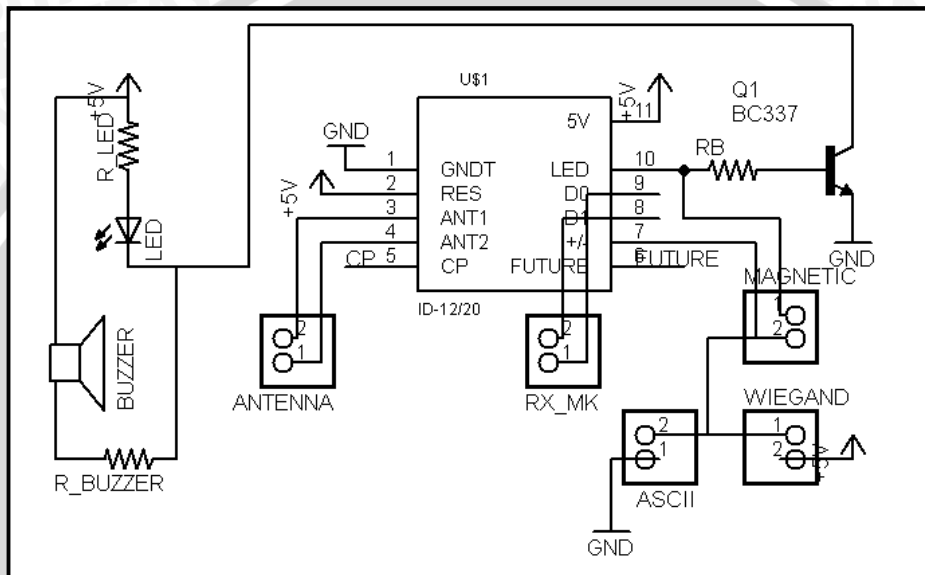
Fungsi utama RFID *Reader* adalah sebagai masukan pada mikrokontroler yang selanjutnya digunakan sebagai acuan perhitungan nomor tempat duduk. Gambar 4.3 menunjukkan gambar rangkaian RFID *Reader*.

Saat RFID *Reader* membaca sebuah tag RFID, maka *buzzer* akan berbunyi dan LED akan menyala. LED yang digunakan adalah LED 5 mm warna merah yang memiliki tegangan V_{LED} sebesar 2,1 V dan arus maksimal LED I_{LED} yang diperbolehkan mengalir

pada LED sebesar 20mA. Sehingga dengan menggunakan Persamaan (4.1) dapat diketahui besarnya R_{LED} .

$$\begin{aligned}
 &= \frac{V_{LED}}{I_{LED}} \quad (4.1) \\
 &= \frac{1.8}{0.02} \\
 &= 110\Omega
 \end{aligned}$$

Besarnya R_{LED} yang diperoleh sebesar 110Ω. R tersebut merupakan nilai minimum, sehingga besarnya resistor yang digunakan dalam perancangan adalah 1kΩ.



Gambar 4.2 Rangkaian RFID Reader

Buzzer yang digunakan adalah tipe HCM1206UX-G. Besarnya arus I_{buzzer} yang diperlukan untuk mengaktifkan buzzer adalah 30mA (maksimal). Sehingga dengan menggunakan Persamaan (4.2) akan diperoleh besarnya R_{buzzer} .

$$\begin{aligned}
 &= \frac{V_{BUZZER}}{I_{BUZZER}} \quad (4.2) \\
 &= \frac{1.2}{0.03} \\
 &= 43,33\Omega
 \end{aligned}$$

Besarnya R_{buzzer} yang diperoleh sebesar 43,33Ω. R tersebut merupakan nilai minimum, sehingga besarnya resistor yang digunakan adalah 220Ω. Dengan arus maksimal buzzer I_{buzzer} sebesar 30mA dan arus maksimal LED I_{LED} sebesar 20mA, maka akan diperoleh besar arus kolektor I_c dengan menggunakan Persamaan (4.3).

$$\begin{aligned}
 &= I_{LED} + I_{BUZZER} \quad (4.3) \\
 &= 20.10 + 30.10 \\
 &= 50
 \end{aligned}$$

Dengan arus kolektor I_c sebesar 50mA, maka transistor yang digunakan harus memiliki arus kolektor I_c lebih besar. Dalam perancangan ini digunakan transistor BC337 yang memiliki arus kolektor maksimal $I_{c(max)}$ sebesar 500mA, tegangan basis-emittor V_{BE} sebesar 1,2V, dan penguatan arus $\beta_{(min)}$ sebesar 160. Sehingga besarnya arus basis I_b dapat diperoleh dengan menggunakan Persamaan (4.4).

$$= \frac{I_c}{\beta_{(min)}} \quad (4.4)$$

$$= \frac{50 \text{ mA}}{160}$$

$$= 312,5$$

Dengan mengasumsikan V_{BB} saat aktif (logika high) sebesar 5V, maka dapat diperoleh besarnya R_b dengan menggunakan Persamaan (4.5).

$$= \frac{V_{BB} - V_{BE}}{I_b} \quad (4.5)$$

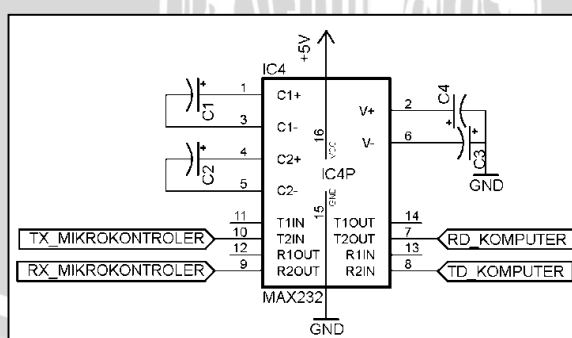
$$= \frac{5 \text{ V} - 1,2 \text{ V}}{312,5 \text{ mA}}$$

$$= 12,2 \text{ } \Omega$$

Sehingga besarnya R_b yang diperoleh dalam perancangan adalah 12,2k Ω .

4.3.2 Perancangan Rangkaian RS232

Perancangan rangkaian RS232 menggunakan rangkaian max232 yang terdiri atas IC max232 dan 4 buah kapasitor untuk mengkonversi level tegangan RS232 ke level tegangan TTL/CMOS. Rangkaian RS232 yang terdiri atas IC max232 dan 4 buah kapasitor ditunjukkan dalam Gambar 4.3.



Gambar 4.3 Rangkaian MAX232 yang terdiri atas IC max232 dan 4 buah kapasitor

Pada perancangan rangkaian max232, beberapa pin pada IC konverter RS232 ke TTL/CMOS dihubungkan dengan piranti-piranti dari luar. Pin C1+ dan C1- dihubungkan dengan menggunakan kapasitor. Pin C1+ terhubung dengan sisi positif kapasitor dan C1- terhubung dengan sisi negative kapasitor. Pin C2+ dan C2- dihubungkan dengan

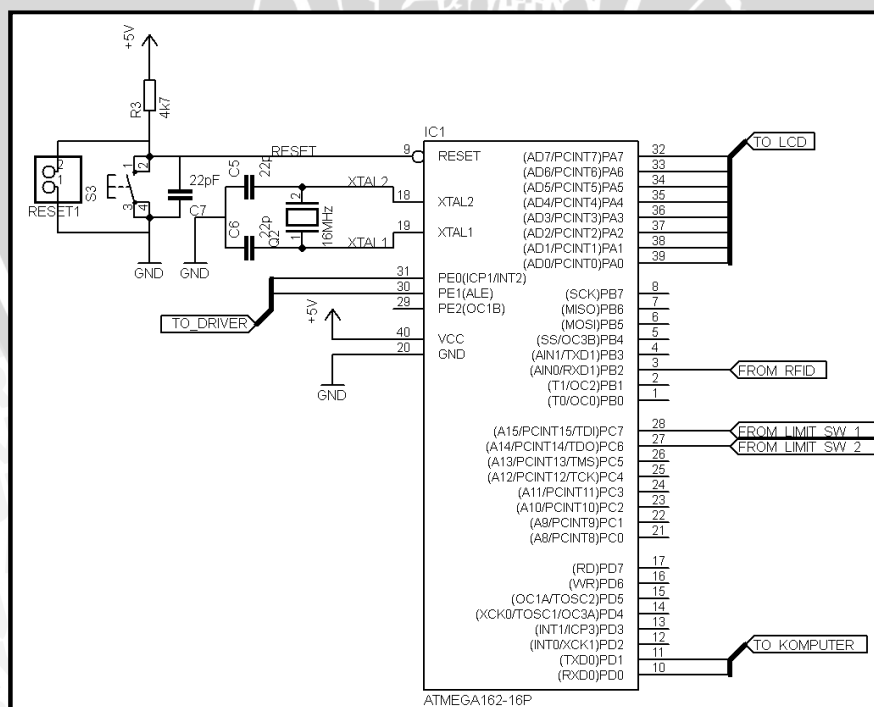
menggunakan kapasitor. Pin C2+ terhubung dengan sisi positif kapasitor dan C2- terhubung dengan sisi negative kapasitor. Pin T2_{IN} dan pin R2_{OUT} masing-masing dihubungkan dengan TX dan RX mikrokontroler, sedangkan pin T2_{OUT} dan pin R2_{IN} masing-masing dihubungkan dengan RD dan TD komputer.

Berdasarkan *datasheet*, nilai kapasitor yang digunakan sebanyak 4 buah dengan masing-masing kapasitor bernilai 1 μ F. Dalam perancangan ini nilai kapasitor yang digunakan sebesar 1 μ F.

4.3.3 Perancangan Rangkaian Mikrokontroler

Salah satu komponen yang digunakan dalam alat ini adalah mikrokontroler ATmega162. Jenis mikrokontroler ini dipilih karena terdapat *dual programmable serial USART* yang digunakan untuk menghubungkan antara mikrokontroler dengan RFID reader dan mikrokontroler dengan komputer yang terdapat pada sistem ini.

Mikrokontroler ini berfungsi sebagai pengolah data dari masukan RFID Reader, kemudian mencocokkan dengan data yang telah tersimpan pada *database* komputer, dan ditampilkan pada LCD serta mengendalikan ON/OFF motor DC. Pin-pin ATmega162 dihubungkan pada rangkaian pendukung membentuk suatu sistem minimum seperti dalam Gambar 4.4.



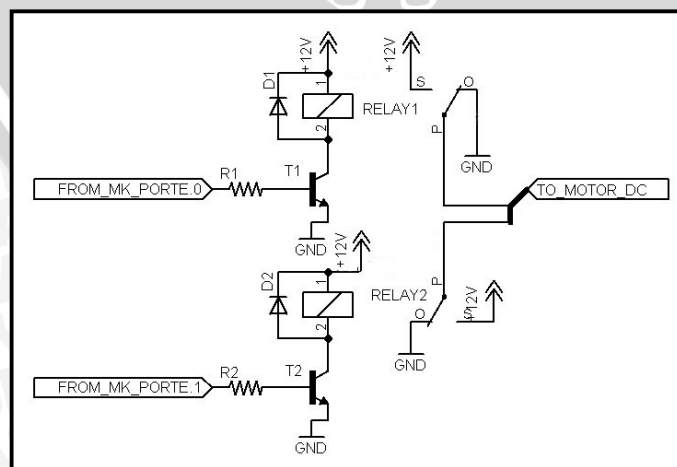
Gambar 4.4 Rangkaian Mikrokontroler ATmega162

Pin-pin yang digunakan pada perancangan ini adalah:

1. PORTA
 - PORTA.0 digunakan sebagai pengirim perintah RS pada LCD.
 - PORTA.1 digunakan sebagai pengirim perintah R/W pada LCD.
 - PORTA.2 digunakan sebagai pengirim perintah *enable* pada LCD.
 - PORTA.4- PORTA.7 digunakan sebagai jalur data dari LCD port D4-D7.
2. PORTB
 - PORTB.2 digunakan untuk komunikasi serial dari RFID *Reader* (pin RXD1).
3. PORTC
 - PORTC.6 digunakan untuk menerima data dari limit switch 2.
 - PORTC.7 digunakan untuk menerima data dari limit switch 1.
4. PORTD
 - PORTD.0 digunakan untuk komunikasi serial dari komputer (pin RXD0).
 - PORTD.1 digunakan untuk komunikasi serial ke komputer (pin TXD0).
5. PORTE
 - PORTE.0-PORTE.1 digunakan sebagai jalur data ke *driver*.

4.3.4 Perancangan Rangkaian *Driver Motor*

Perancangan rangkaian *driver motor* DC digunakan sebagai saklar yang menghubungkan antara rangkaian mikrokontroler dengan motor DC 12V. Rangkaian ini digunakan karena tegangan yang dikeluarkan dari *port* mikrokontroler tidak mencukupi untuk menggunakan peralatan yang dikontrol, yaitu motor DC. Perancangan ini terdiri atas relay, transistor dan resistor. Rangkaian *driver motor* DC ditunjukkan dalam Gambar 4.5.



Gambar 4.5 Rangkaian *Driver Motor* DC

Rangkaian *driver* menggunakan dua buah relay, dua buah transistor, dan dua buah resistor. Rangkaian pertama berfungsi untuk mengaktifkan motor, sedangkan rangkaian kedua berfungsi sebagai pembalik polaritas. Sehingga motor DC dapat berputar searah jarum jam juga dapat berputar berlawanan jarum jam.

Relay yang digunakan dalam rangkaian *driver* adalah relay tipe HRS4 dengan catu tegangan 12V. Besarnya resistansi koil relay adalah 320Ω , maka besar arus kolektor I_c dapat diperoleh dengan menggunakan Persamaan (4.6).

$$\begin{aligned}
 &= \frac{V}{R} \\
 &= \frac{12\text{V}}{320\Omega} \\
 &= 37,5 \text{ mA}
 \end{aligned} \tag{4.6}$$

Dengan arus kolektor I_c sebesar 37,5mA, maka transistor yang digunakan harus memiliki arus kolektor I_c lebih besar. Dalam perancangan ini digunakan transistor BC337 yang memiliki arus kolektor maksimal $I_{c(\max)}$ sebesar 500mA, tegangan basis-emitor VBE sebesar 1,2V, dan penguatan arus $\beta_{(\min)}$ sebesar 160. Sehingga besarnya arus basis I_b dapat diperoleh dengan menggunakan Persamaan (4.7).

$$\begin{aligned}
 &= \frac{I_c}{\beta} \\
 &= \frac{37,5 \text{ mA}}{160} \\
 &= 234,375 \mu\text{A}
 \end{aligned} \tag{4.7}$$

Dengan mengasumsikan V_{BB} , tegangan keluaran mikrokontroler, saat aktif (logika high) sebesar 5V, maka dapat diperoleh besarnya R_b dengan menggunakan Persamaan (4.8).

$$\begin{aligned}
 &= \frac{V_{BB}}{I_b} \\
 &= \frac{5\text{V}}{234,375 \mu\text{A}} \\
 &= 16,2 \text{ k}\Omega
 \end{aligned} \tag{4.8}$$

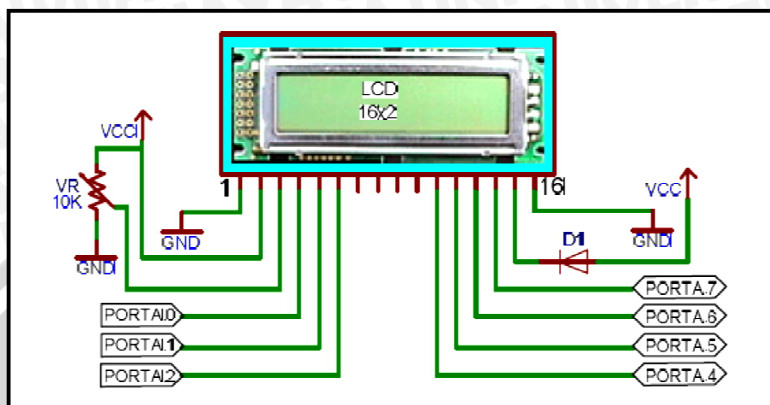
Karena rangkaian *driver* terdiri atas dua rangkaian relay dengan penggunaan tipe komponen yang sama, maka perhitungan cukup dilakukan satu kali. Sehingga untuk memperoleh besar R_1 dan R_2 dapat menggunakan Persamaan (4.9).

$$R_1 = R_2 = 16,2 \text{ k}\Omega \tag{4.9}$$

Sehingga besarnya R_b yang diperoleh dalam perancangan adalah 16,2k Ω .

4.3.5 Perancangan Rangkaian LCD

Perancangan antara mikrokontroler dengan LCD menggunakan tujuh buah pin, yaitu PORTA.0 untuk PIN RS (Register Select), PORTA.1 untuk R/W (Read/Write), PORTA.2 untuk EN (Enable), dan PORTA.4-PORTA.7 untuk DB4-DB7. Perancangan rangkaian LCD ditunjukkan dalam Gambar 4.6.



Gambar 4.6 Rangkaian LCD

Kaki nomor 3 digunakan untuk mengatur kontras LCD, pada kaki ini digunakan variabel resistor dengan nilai 10K Ω . Sedangkan kaki nomor 15 digunakan untuk mengatur lampu back light LCD dengan penambahan dioda IN4004 agar tegangan pada kaki ini sebesar 4.3 volt.

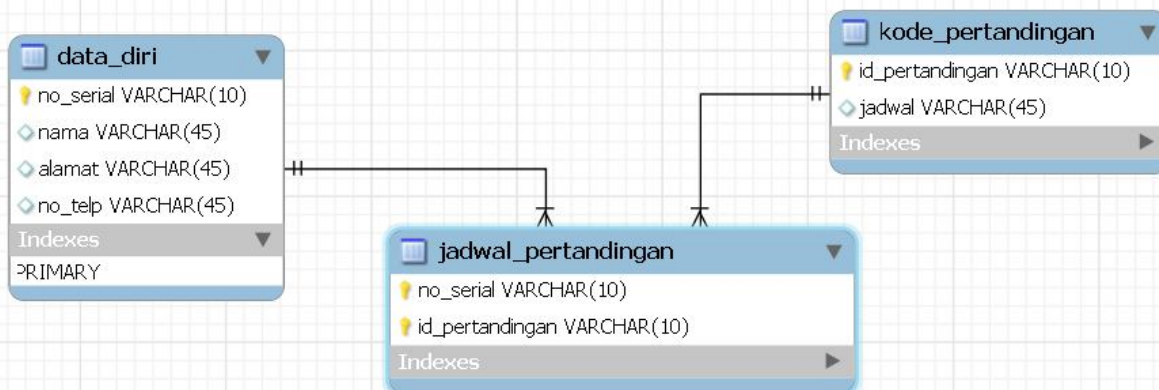
4.4 Perancangan Perangkat Lunak (*Software*)

Untuk mengendalikan perangkat keras dibutuhkan perangkat lunak. Perangkat lunak dalam sistem ini terdiri atas tiga bagian, yaitu perangkat lunak pada komputer dan perangkat lunak pada mikrokontroler. Perangkat lunak yang akan dirancang diharapkan dapat mengoperasikan sistem secara optimal.

4.4.1 Perancangan *Database*

Perancangan *database* dilakukan dengan membuat rancangan tabel serta isi dari tabel tersebut. Dalam perancangan, *database* terdiri atas tiga buah tabel yang saling berhubungan, yaitu tabel data diri yang berisi informasi nomor serial, nama, alamat, dan nomor telepon. Dengan nomor serial yang berfungsi sebagai *primary key* dari table data diri. Table yang kedua adalah kode pertandingan, yang berisi nomor *id* pertandingan, tim lawan, dan tanggal pertandingan berlangsung. Kunci primer dari table ini adalah *id* pertandingan. Table yang terakhir merupakan table relasi, karena table tersebut merupakan hubungan antara table data diri dengan table kode pertandingan. Untuk mempermudah

perancangan *database*, digunakan *software* MySQL Workbench. Hubungan antara table ditunjukkan dalam Gambar 4.7.



Gambar 4.7 Hubungan Antara Tabel pada Database

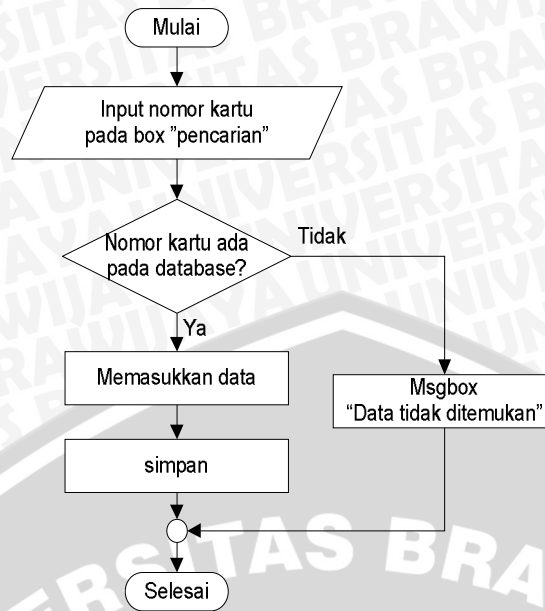
4.4.2 Perancangan Perangkat Lunak Komputer

Sistem ini terdiri atas mikrokontroler sebagai unit utama, RFID sebagai media untuk mengidentifikasi, *driver*, motor DC, penampil LCD dan komputer yang berfungsi untuk masukan data, menyimpan data, dan mengirimkan data ke piranti luar. Untuk dapat menjalankan fungsinya dengan baik maka komputer perlu diprogram terlebih dahulu. Perangkat lunak komputer dibuat menggunakan aplikasi program Delphi 7. Bahasa pemrograman yang digunakan adalah bahasa C.

Perancangan perangkat lunak komputer dibagi dua menurut waktu, yaitu saat proses pemesanan tiket dan saat hari pertandingan.

4.4.2.1 Perancangan Perangkat Lunak Komputer pada Proses Pemesanan Tiket

Perancangan yang dilakukan pada proses pemesanan tiket yang berfungsi untuk mengidentifikasi nomor kartu telah terdaftar sebagai anggota atau belum dan memasukkan data berupa jadwal pertandingan yang tiketnya telah dipesan oleh anggota pada *database*. Diagram alir sub rutin memasukkan data pada proses pemesanan tiket ditunjukkan dalam Gambar 4.8.



Gambar 4.8 Diagram Alir Proses Pemesanan Tiket

Perancangan dimulai saat seorang supporter (anggota) akan memesan tiket pertandingan pada saat yang bersamaan operator akan mengidentifikasi kartu anggota supporter tersebut dengan cara manual, yaitu mencari data berupa nomor kartu anggota (nomor serial tag RFID) pada *database* dengan memanfaatkan tombol “pencarian”. Setelah nomor kartu ditemukan, operator akan memasukkan data pada *database* berupa jadwal pertandingan yang telah dipesan oleh anggota tersebut.

4.4.2.2 Perancangan Perangkat Lunak Komputer di Hari Pertandingan

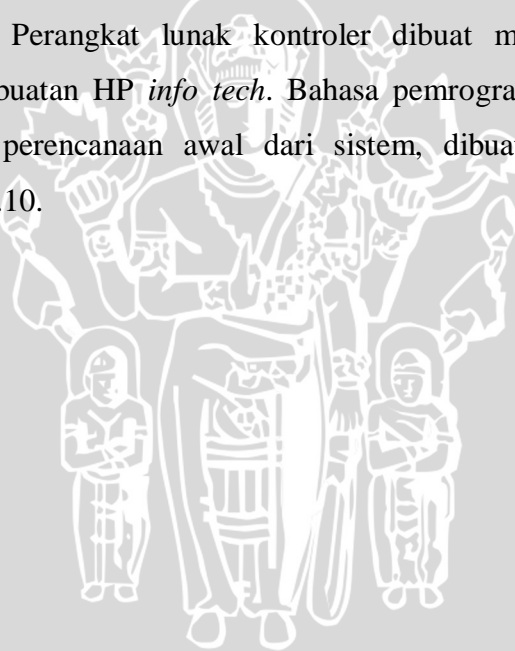
Perancangan perangkat lunak komputer di hari pertandingan berfungsi untuk mengidentifikasi nomor kartu anggota telah terdaftar pada jadwal pertandingan tersebut atau belum dan mencetak informasi bagi anggota berupa nomor tempat duduk dan jadwal pertandingan yang telah dipesan menggunakan *printer*. Diagram alir perancangan perangkat lunak komputer di hari pertandingan ditunjukkan dalam Gambar 4.9.

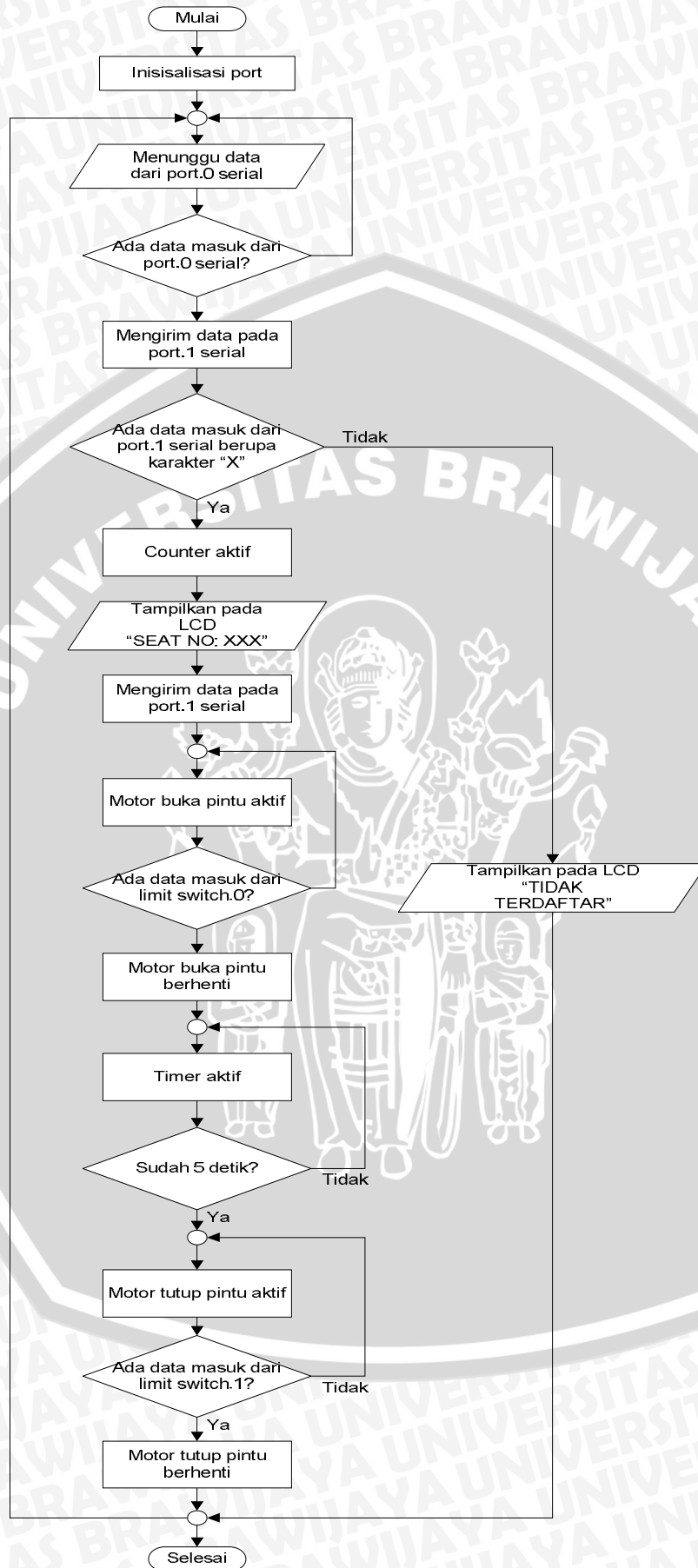
data yang berhasil dan tidak berhasil ditemukan pada database ke mikrokontroler melalui serial port. Kemudian, data nomor tempat duduk akan masuk melalui serial port. Setelah itu komputer mengambil data dari database berupa data jadwal pertandingan yang telah dipesan dan menggabungkannya dengan data nomor tempat duduk. Kemudian data-data tersebut dicetak dalam format tertentu menggunakan *printer*.

4.4.3 Perancangan Perangkat Lunak Mikrokontroler

Sistem ini terdiri atas mikrokontroler sebagai unit utama, RFID sebagai media untuk mengidentifikasi, *driver*, motor DC, penampil LCD dan komputer. Mikrokontroler sebagai unit utama pengolah data berfungsi untuk menerima data dari RFID *reader*, menentukan nomor tempat duduk dan menampilkan pada LCD, mengaktifkan motor, dan mengirimkan data ke piranti luar.

Untuk dapat menjalankan fungsinya dengan baik maka mikrokontroler perlu diprogram terlebih dahulu. Perangkat lunak kontroler dibuat menggunakan aplikasi program *Code Vision AVR* buatan HP *info tech*. Bahasa pemrograman yang digunakan adalah bahasa C. Sebagai perencanaan awal dari sistem, dibuat diagram alir yang ditunjukkan dalam Gambar 4.10.





Gambar 4.10 Diagram Alir Perancangan Perangkat Lunak Mikrokontroler

Diagram alir perancangan perangkat lunak mikrokontroler dapat diuraikan sebagai berikut. Setelah menginisialisasi port-port yang digunakan, mikrokontroler menunggu data dari port.0 serial yaitu port serial yang dihubungkan dengan RFID *reader*. Jika tag RFID didekatkan dengan RFID *reader*, maka RFID *reader* akan mengidentifikasi nomor serial pada tag tersebut. RFID *reader* akan mengirimkan data berupa bilangan ASCII pada mikrokontroler. Kemudian data tersebut oleh mikrokontroler dikirimkan ke port.1 serial, yaitu port serial untuk komputer. Kemudian mikrokontroler menunggu data yang dikirimkan oleh komputer. Jika data yang dikirim adalah karakter “X”, itu artinya nomor serial dari tag RFID tersebut telah terdaftar pada *database* sebagai penonton yang akan menyaksikan pertandingan pada jadwal tersebut. Jika data yang dikirim adalah karakter “Y”, berarti nomor serial tersebut belum terdaftar pada *database*, yang artinya adalah pelanggan belum memesan tiket pertandingan. Karena belum membeli tiket, maka pelanggan tersebut tidak dapat masuk stadion karena pintu tidak terbuka. Setelah mikrokontroler mengetahui nomor serial tersebut telah terdaftar pada *database*, *counter* menjadi aktif. Kegunaan *counter* adalah untuk menentukan nomor tempat duduk. Nomor tempat duduk akan ditampilkan pada LCD. Bersamaan dengan itu, data nomor tempat duduk dikirimkan ke komputer melalui port serial. Kemudian mikrokontroler memberikan logika *high* pada port keluaran yang dihubungkan dengan *driver* motor 0 sehingga motor DC aktif dan membuka palang pintu sampai mikrokontroler mendapatkan tegangan masukan dari limit *switch* 0. Ketika palang pintu menyentuh limit *switch* 0, *timer* aktif. Fungsi *timer* adalah untuk menghitung waktu sampai lima detik. Setelah lima detik, mikrokontroler akan memberikan logika *high* pada port keluaran yang dihubungkan dengan *driver* motor 1 yang akan menyebabkan motor tutup pintu aktif sampai palang pintu menyentuh limit *switch* 1 dan mikrokontroler mendapatkan data dari limit *switch* 1 untuk menghentikan putaran motor DC. Proses tersebut akan berlangsung terus menerus mulai dari proses pengambilan data dari RFID *reader* sampai dengan menutupnya palang pintu oleh motor DC dan akan berhenti ketika sistem tersebut dimatikan.

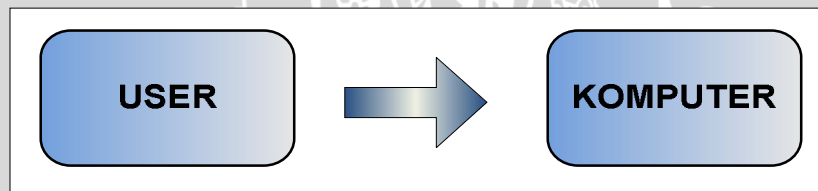
BAB V

PENGUJIAN DAN ANALISIS

Dalam bab ini membahas pengujian dan analisis alat yang telah dirancang dari peralatan yang telah dibuat. Pengujian dilakukan dengan pengukuran tiap-tiap blok dengan tujuan mengamati apakah blok-blok tersebut bekerja sesuai dengan yang diharapkan. Pengujian ini dilakukan berdasarkan pada masing-masing rangkaian pendukung secara keseluruhan. Pengujian yang dilakukan adalah sebagai berikut:

5.1 Pengujian Program Aplikasi sebagai Penghubung dengan *Database*

Pengujian program aplikasi sebagai penghubung dengan *database* terbagi dalam tiga bagian, yaitu untuk mencari data dari dalam *database*, mengubah data pada *database*, memasukkan data ke dalam *database* dan menghapus data dari dalam *database*. Diagram blok pengujian program aplikasi sebagai penghubung dengan *database* ditunjukkan dalam Gambar 5.1.



Gambar 5.1 Diagram Blok Pengujian Program Aplikasi sebagai Penghubung dengan *Database*

Pengujian program aplikasi sebagai penghubung dengan *database* ini dilakukan dengan memasukkan data oleh *user* ke dalam komputer yang telah terisi program aplikasi yang menghubungkan tampilan program Delphi dengan *database* MySQL.

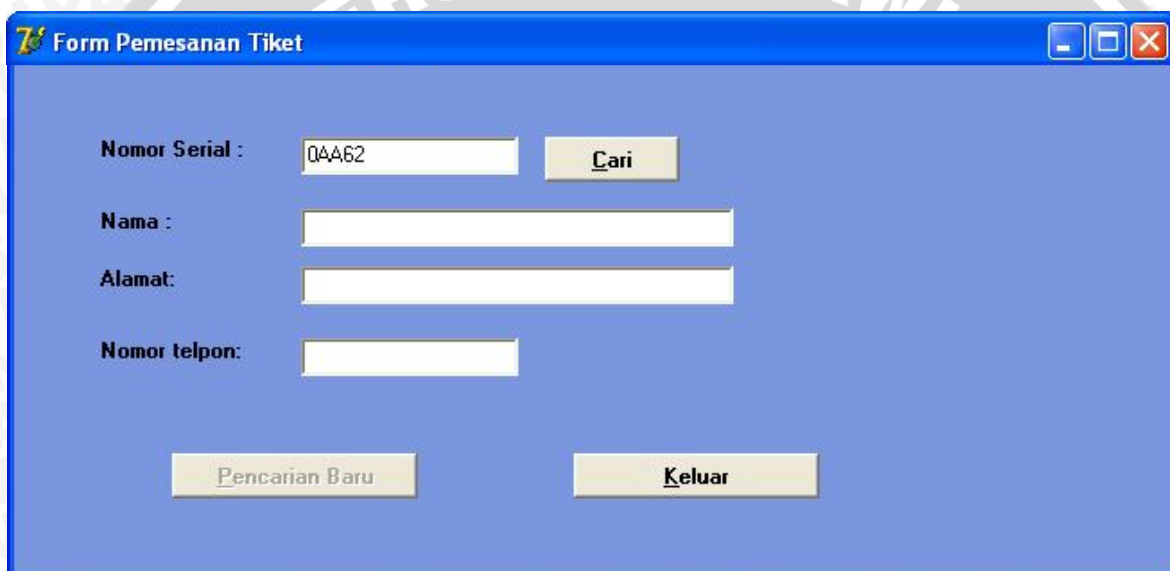
5.1.1 Pengujian Program Aplikasi untuk Mencari Data dari dalam *Database*

Pengujian program aplikasi untuk mencari data dari *database* bertujuan untuk mengetahui dan menganalisis keberhasilan data yang berupa kunci pencarian yang dimasukkan oleh *user* (operator) dapat ditampilkan di program aplikasi sesuai dengan program yang telah dibuat.

Dalam pengujian ini, program aplikasi Delphi akan menampilkan sebuah tampilan program yang akan diisi data oleh operator. Data tersebut berupa nomor serial yang ada pada kartu tiap anggota yang nantinya akan dicari dari dalam *database*. Nomor serial berfungsi sebagai kunci utama, karena nomor serial membawa data-data berupa nama, alamat, nomor telepon, dan tiket pertandingan yang dipesan. Data-data tersebut yang akan

ditampilkan pada program aplikasi Delphi yang telah disimpan sebelumnya dalam *database* tabel data diri. Pengujian dilakukan dengan melihat data yang ditampilkan pada program aplikasi Delphi setelah operator memasukkan nomor serial pada kotak pencarian apakah sesuai dengan data dalam *database*. Pengujian dilakukan sebanyak 3 kali dengan tampilan hasil pengujian yaitu tampilan program Delphi dan tampilan *database* tabel data diri.

Hasil pengujian yang pertama berupa tampilan Program Delphi saat operator memasukkan data ditunjukkan dalam Gambar 5.2, setelah operator melakukan pencarian data ditunjukkan dalam Gambar 5.3, serta tampilan *database* tabel data diri ditunjukkan dalam Tabel 5.1.



The image shows a screenshot of a Delphi application window titled "Form Pemesanan Tiket". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The main area of the window is light blue and contains the following elements:

- Nomor Serial :** A text input field containing the value "04A62". To its right is a button labeled "Cari".
- Nama :** An empty text input field.
- Alamat:** An empty text input field.
- Nomor telpon:** An empty text input field.
- At the bottom of the form, there are two buttons: "Pencarian Baru" on the left and "Keluar" on the right.

Gambar 5.2 Tampilan Program Aplikasi saat Operator Memasukkan Data dalam Kotak Pencarian

Gambar 5.3 Tampilan Program Aplikasi saat Data Telah Ditampilkan

Tabel 5.1 Tabel *Database* Data Diri

	no_serial	nama	alamat	no_telp
<input type="checkbox"/>	0AA62	Atika Iqlimah	Jl. Raya Dau no. 17F	08563689037
<input type="checkbox"/>	0F62E	Farishal Anas Ramadhan	Jl. D. Toba no. 67 Sawojajar	081703043365
<input type="checkbox"/>	24BB1	Hafrida Rahmah	jl. Sulfat Indah no. 80	081334535230
<input type="checkbox"/>	66A84	Fahmi Nurul Akbar	Jl. Stadion IX no. 4	085749247620
<input type="checkbox"/>	6E37D	Inas Widaningrum H.	Jl. Raya Tidar no. 110	087759985095
<input type="checkbox"/>	7462E	Suci Imani	jl. Batu Jahe 78	085755770008
<input type="checkbox"/>	7DB24	Rizal Lana	Jl. MT. Haryono no. 17	085920564515
<input type="checkbox"/>	8E737	Handiny Indah P.	Jl. Cempedak 1 no. 11 Perumnas Btr. Kemang	08568129589
<input type="checkbox"/>	BF467	Refita Edna K.	Perum Alam Hijau Jl. Raya Singosari 12A	085655585899
<input type="checkbox"/>	C71AB	Yuli Khoirul	Jl. Raya Bululawang Permai no. 19	085649335332

Hasil pengujian yang ada sesuai antara data yang ditampilkan dalam program aplikasi dengan data yang tersimpan dalam *database*. Pengujian yang kedua sama dengan pengujian pertama namun yang berbeda adalah data yang dimasukkan operator. Hasil pengujian kedua berupa tampilan program Delphi setelah operator melakukan pencarian data ditunjukkan dalam Gambar 5.4.

Form Pemesanan Tiket

Nomor Serial :

Nama :

Alamat:

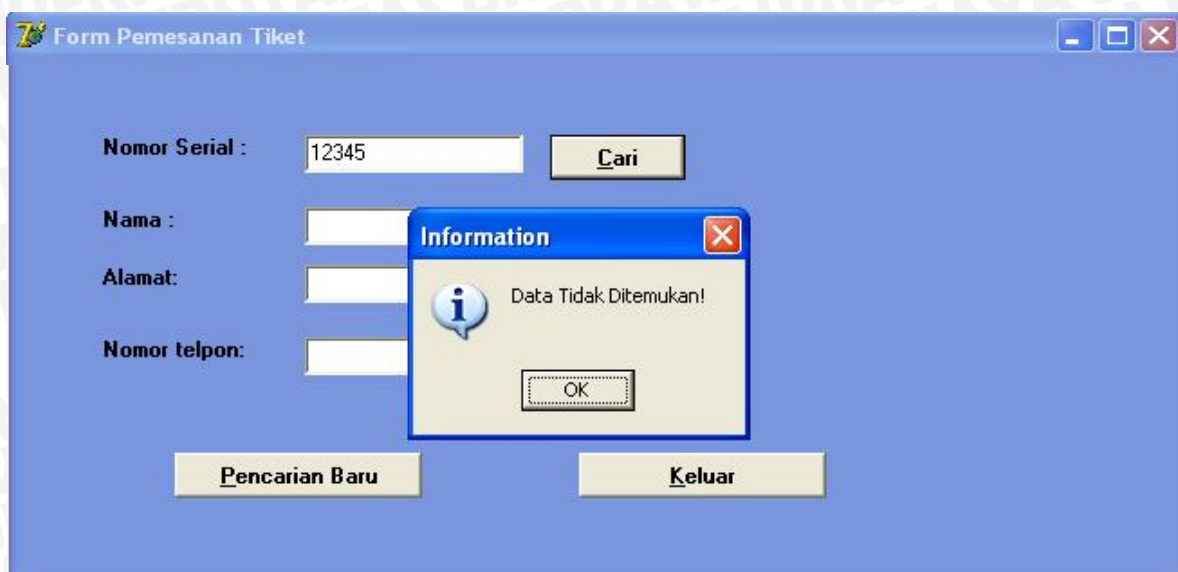
Nomor telpon:

Gambar 5.4 Tampilan Program Aplikasi saat Data Telah Ditampilkan

Tabel 5.2 Tabel *Database* Data Diri

			no_serial	nama	alamat	no_telp
<input type="checkbox"/>			0AA62	Atika Iqlimah	Jl. Raya Dau no. 17F	08563689037
<input type="checkbox"/>			0F62E	Farishal Anas Ramadhan	Jl. D. Toba no. 67 Sawojajar	081703043355
<input type="checkbox"/>			24BB1	Hafrida Rahmah	jl. Sulfat Indah no. 80	081334535230
<input type="checkbox"/>			66A84	Fahmi Nurul Akbar	Jl. Stadion IX no. 4	085749247620
<input type="checkbox"/>			6E37D	Inas Widaningrum H.	Jl. Raya Tidar no. 110	087759985095
<input type="checkbox"/>			7462E	Suci Imani	jl. Batu Jahe 78	085755770008
<input type="checkbox"/>			7DB24	Rizal Lana	Jl. MT. Haryono no. 17	085920564515
<input type="checkbox"/>			8E737	Handiny Indah P.	Jl. Cempedak 1 no. 11 Perumnas Btr. Kemang	08568129589
<input type="checkbox"/>			BF467	Refita Edna K.	Perum Alam Hijau Jl. Raya Singosari 12A	085655585699
<input type="checkbox"/>			C71AB	Yuli Khoirul	Jl. Raya Bululawang Permai no. 19	085649335332

Hasil pengujian yang ada sesuai antara data yang dimasukkan *user* dengan data yang tersimpan dalam *database*. Pengujian ketiga yaitu melihat data yang ditampilkan saat operator salah memasukkan data nomor serial ke dalam kotak pencarian. Hasil pengujian ketiga ditunjukkan dalam Gambar 5.5.



Gambar 5.5 Tampilan Program Aplikasi saat Operator Memasukkan Data yang Salah

Dikarenakan data yang dimasukkan oleh operator tidak ada dalam *database*, maka program akan menampilkan informasi yang memberitahukan bahwa data tersebut tidak ditemukan.

5.1.2 Pengujian Program Aplikasi untuk Mengubah Data pada *Database*

Pengujian program aplikasi untuk mengubah data pada *database* bertujuan untuk mengetahui dan menganalisis keberhasilan data yang diubah oleh operator melalui program aplikasi telah mengubah data pada *database* sesuai dengan program yang telah dibuat.

Dalam pengujian ini, data yang telah ditampilkan setelah proses pencarian akan diubah oleh operator pada program aplikasi Delphi. Data yang dapat diubah berupa nama, alamat, dan nomor telepon, sedangkan nomor serial tidak dapat diubah. Setelah operator mengubah data, kemudian oleh operator data tersebut disimpan. Data-data tersebut akan tersimpan di *database*. Pengujian dilakukan dengan membandingkan data yang ditampilkan pada program aplikasi Delphi setelah operator mengubah data dan menyimpannya apakah sesuai dengan perubahan data dalam *database*.

The screenshot shows a window titled "Form Pemesanan Tiket" with a blue border. It contains several input fields and buttons. The "Nomor Serial" field has the value "0F62E". The "Nama" field is highlighted with a red border and contains "Anas Ramadhan". The "Alamat" field contains "Jl. D. Toba no. 67 Sawojajar". The "Nomor telpon" field contains "081703043355". On the right side, there are buttons for "Pesan Tiket", "Lihat Tiket", "Ubah Data", and "Simpan". At the bottom, there are buttons for "Pencarian Baru" and "Keluar". A "Cari" button is located next to the "Nomor Serial" field.

Gambar 5.6 Tampilan Program Aplikasi
Sebelum Data Diubah

This screenshot is identical in layout to the previous one, but the "Nama" field now contains the updated name "Farishal Anas Ramadhan", which is also highlighted with a red border. All other fields and buttons remain the same.

Gambar 5.7 Tampilan Program Aplikasi
Setelah Data Diubah

Gambar 5.8 Tampilan Program Aplikasi
Setelah Data Disimpan

Tabel 5.3 Tabel Data Diri pada *Database*
Sebelum Data Diubah

	no_serial	nama	alamat	no_telp
<input type="checkbox"/>	0AA62	Atika Iqlimah	Jl. Raya Dau no. 17F	08563689037
<input type="checkbox"/>	0F62E	Anas Ramadhan	Jl. D. Toba no. 67 Sawojajar	081703043355
<input type="checkbox"/>	24BB1	Hafrida Rahmah	jl. Sulfat Indah no. 80	081334535230
<input type="checkbox"/>	66A84	Fahmi Nurul Akbar	Jl. Stadion IX no. 4	085749247620
<input type="checkbox"/>	6E37D	Inas Widaningrum H.	Jl. Raya Tidar no. 110	087759985095
<input type="checkbox"/>	7462E	Suci Imani	jl. Batu Jahe 78	085755770008
<input type="checkbox"/>	7DB24	Rizal Lana	Jl. MT. Haryono no. 17	085920564515
<input type="checkbox"/>	8E737	Handiny Indah P.	Jl. Cempedak 1 no. 11 Perumnas Btr. Kemang	08568129589
<input type="checkbox"/>	BF467	Refita Edna K.	Perum Alam Hijau Jl. Raya Singosari 12A	085655585699
<input type="checkbox"/>	C71AB	Yuli Khoirul	Jl. Raya Bululawang Permai no. 19	085649335332

Tabel 5.4 Tabel Data Diri pada *Database*
Setelah Data Diubah

	no_serial	nama	alamat	no_telp
<input type="checkbox"/>	0AA62	Atika Iqlimah	Jl. Raya Dau no. 17F	08563689037
<input type="checkbox"/>	0F62E	Farishal Anas Ramadhan	Jl. D. Toba no. 67 Sawojajar	081703043355
<input type="checkbox"/>	24BB1	Hafrida Rahmah	jl. Sulfat Indah no. 80	081334535230
<input type="checkbox"/>	66A84	Fahmi Nurul Akbar	Jl. Stadion IX no. 4	085749247620
<input type="checkbox"/>	6E37D	Inas Widaningrum H.	Jl. Raya Tidar no. 110	087759985095
<input type="checkbox"/>	7462E	Suci Imani	jl. Batu Jahe 78	085755770008
<input type="checkbox"/>	7DB24	Rizal Lana	Jl. MT. Haryono no. 17	085920564515
<input type="checkbox"/>	8E737	Handiny Indah P.	Jl. Cempedak 1 no. 11 Perumnas Btr. Kemang	08568129589
<input type="checkbox"/>	BF467	Refita Edna K.	Perum Alam Hijau Jl. Raya Singosari 12A	085655585699
<input type="checkbox"/>	C71AB	Yuli Khoirul	Jl. Raya Bululawang Permai no. 19	085649335332

Gambar 5.6 menunjukkan tampilan program aplikasi saat keadaan awal, yaitu saat data belum diubah. Kemudian operator mengubah data berupa nama ditunjukkan dalam Gambar 5.7. Setelah selesai mengubah data, operator menekan tombol 'Simpan' yang akan mengeksekusi perintah yang diberikan yaitu untuk mengubah data pada *database* MySQL dan pada tampilan program aplikasi akan tampil kotak pesan untuk memberitahu bahwa perintah telah dieksekusi.










Tabel 5.3 menunjukkan data di dalam *database* sebelum data diubah oleh operator dan kemudian data berupa nama pada *database* akan berubah sesuai dengan data yang diberikan oleh operator, hal tersebut ditunjukkan dalam Tabel 5.4.

Setelah membandingkan data pada program aplikasi Delphi dan *database* MYSQL, hasil pengujian yang ada sesuai antara data yang ditampilkan dalam program aplikasi dengan data yang tersimpan dalam *database*.

5.1.3 Pengujian Program Aplikasi untuk Memasukkan Data ke Dalam *Database*

Pengujian program aplikasi untuk memasukkan data ke dalam *database* bertujuan untuk mengetahui dan menganalisis keberhasilan data yang dimasukkan oleh operator melalui program aplikasi telah tertampung pada *database* sesuai dengan program yang telah dibuat. Pengujian program aplikasi untuk memasukkan data ke dalam *database* ini dilakukan dengan memasukkan data oleh operator ke dalam komputer yang telah terisi program aplikasi yang menghubungkan tampilan program Delphi dengan *database* MySQL. Dalam program aplikasi digunakan bentuk *checkbox*. Jika terdapat tanda *checkboxlist* (√), maka data tersebut akan dimasukkan dan disimpan dalam *database*. Pengujian memasukkan data ke dalam *database* melalui program aplikasi dilakukan sebanyak 3 kali.

Tabel 5.5 Tabel Jadwal Pertandingan pada *Database*
Belum Terisi Data

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	no_serial	varchar(5)	latin1_swedish_ci		No	None		      
<input type="checkbox"/>	id_pertandingan	varchar(10)	latin1_swedish_ci		No	None		      

	TIM LAWAN	TANGGAL
<input checked="" type="checkbox"/>	Tim A	4 - 11 - 2011
<input type="checkbox"/>	Tim B	12 - 11 - 2011
<input type="checkbox"/>	Tim C	19 - 11 - 2011
<input type="checkbox"/>	Tim D	25 - 11 - 2011
<input checked="" type="checkbox"/>	Tim E	3 - 12 - 2011
<input type="checkbox"/>	Tim F	9 - 12 - 2011
<input checked="" type="checkbox"/>	Tim G	16 - 12 - 2011
<input type="checkbox"/>	Tim H	23 - 12 - 2011
<input type="checkbox"/>	Tim I	30 - 12 - 2011
<input type="checkbox"/>	Tim J	6 - 1 - 2012
<input type="checkbox"/>	Tim K	14 - 1 - 2012
<input type="checkbox"/>	Tim L	21 - 1 - 2012
<input type="checkbox"/>	Tim M	27 - 1 - 2012
<input type="checkbox"/>	Tim N	3 - 2 - 2012
<input type="checkbox"/>	Tim O	11 - 2 - 2012
<input checked="" type="checkbox"/>	Tim P	18 - 2 - 2012

Gambar 5.9 Tampilan Program Aplikasi Memasukkan Data

Tabel 5.6 Tabel Jadwal Pertandingan pada *Database*

			no_serial	id_pertandingan
<input type="checkbox"/>			66A84	01
<input type="checkbox"/>			66A84	05
<input type="checkbox"/>			66A84	07
<input type="checkbox"/>			66A84	16

Tabel 5.5 menunjukkan tabel jadwal pertandingan pada *database* yang belum terisi data. Ketika operator memilih jadwal pertandingan yang dipesan oleh pelanggan dengan cara memberikan tanda *checkboxlist* () , ditunjukkan dalam Gambar 5.9, secara otomatis data masuk dan tersimpan dalam *database*, ditunjukkan dalam Tabel 5.6.












Hasil pengujian yang ada sesuai antara data yang dimasukkan operator dengan data yang tersimpan dalam *database*. Hasil pengujian ini merupakan hasil pengujian yang pertama sehingga pada *database* tabel jadwal pertandingan hanya tersimpan data pelanggan pertama.

Pengujian yang kedua sama dengan pengujian pertama namun yang berbeda adalah data yang dimasukkan operator. Hasil pengujian kedua berupa tampilan program Delphi yang dimasukkan oleh operator ditunjukkan dalam Gambar 5.10, tampilan *database* tabel jadwal pertandingan ditunjukkan dalam Tabel 5.8.

TIM LAWAN	TANGGAL
<input type="checkbox"/> Tim A	4 - 11 - 2011
<input type="checkbox"/> Tim B	12 - 11 - 2011
<input type="checkbox"/> Tim C	19 - 11 - 2011
<input checked="" type="checkbox"/> Tim D	25 - 11 - 2011
<input type="checkbox"/> Tim E	3 - 12 - 2011
<input type="checkbox"/> Tim F	9 - 12 - 2011
<input type="checkbox"/> Tim G	16 - 12 - 2011
<input checked="" type="checkbox"/> Tim H	23 - 12 - 2011
<input checked="" type="checkbox"/> Tim I	30 - 12 - 2011
<input type="checkbox"/> Tim J	6 - 1 - 2012
<input type="checkbox"/> Tim K	14 - 1 - 2012
<input type="checkbox"/> Tim L	21 - 1 - 2012
<input type="checkbox"/> Tim M	27 - 1 - 2012
<input type="checkbox"/> Tim N	3 - 2 - 2012
<input type="checkbox"/> Tim O	11 - 2 - 2012
<input type="checkbox"/> Tim P	18 - 2 - 2012

Gambar 5.10 Tampilan Program Aplikasi
Memasukkan Data

Tabel 5.7 Tabel Jadwal Pertandingan pada Database

			no_serial	id_pertandingan
<input type="checkbox"/>			66A84	01
<input type="checkbox"/>			66A84	07
<input type="checkbox"/>			66A84	16
<input type="checkbox"/>			BF467	04
<input type="checkbox"/>			BF467	08
<input type="checkbox"/>			BF467	09

Hasil pengujian yang ada sesuai antara data yang dimasukkan operator dengan data yang tersimpan dalam *database*. Hasil pengujian ini merupakan hasil pengujian kedua sehingga pada *database* tabel jadwal tersimpan data baru dan data sebelumnya.

Pengujian yang ketiga sama dengan pengujian pertama dan kedua namun yang berbeda adalah data yang dimasukkan operator. Hasil pengujian ketiga berupa tampilan program Delphi yang dimasukkan oleh operator ditunjukkan dalam Gambar 5.11, tampilan *database* tabel jadwal pertandingan ditunjukkan dalam Tabel 5.8.





Gambar 5.11 Tampilan Program Aplikasi Memasukkan Data

Tabel 5.8 Tabel Jadwal Pertandingan pada Database

	no_serial	id_pertandingan
<input type="checkbox"/> ✎ ✖	66A84	01
<input type="checkbox"/> ✎ ✖	66A84	07
<input type="checkbox"/> ✎ ✖	66A84	16
<input type="checkbox"/> ✎ ✖	BF467	04
<input type="checkbox"/> ✎ ✖	BF467	08
<input type="checkbox"/> ✎ ✖	BF467	09
<input type="checkbox"/> ✎ ✖	24BB1	11
<input type="checkbox"/> ✎ ✖	24BB1	10
<input type="checkbox"/> ✎ ✖	24BB1	13
<input type="checkbox"/> ✎ ✖	24BB1	06

Hasil pengujian yang ada sesuai antara data yang dimasukkan operator dengan data yang tersimpan dalam *database*. Hasil pengujian ini merupakan hasil pengujian ketiga sehingga pada *database* tabel jadwal tersimpan data baru dan data sebelumnya.

5.1.4 Pengujian Program Aplikasi untuk Menghapus Data

Pengujian program aplikasi untuk menghapus data bertujuan untuk mengetahui dan menganalisis keberhasilan data yang dihapus oleh operator sudah terhapus dari dalam *database* sesuai dengan program yang telah dibuat. Pengujian ini dilakukan dengan menghapus data yang telah dimasukkan oleh operator sebelumnya pada komputer yang telah terisi program aplikasi yang menghubungkan tampilan program Delphi dengan *database* MySQL.

Dalam pengujian ini, program aplikasi Delphi akan menampilkan sebuah tampilan program yang akan dihapus oleh operator berupa data-data yang sebelumnya telah dimasukkan oleh operator, yaitu jadwal pertandingan yang yang dipesan oleh pelanggan yang nantinya secara otomatis akan terhapus dari dalam *database*.

Gambar 5.12 (a) dan (b) menunjukkan tampilan program aplikasi memasukkan data saat operator menandai *checkbox* dengan *checkboxlist* (✓) dan saat operator membatalkan pilihannya, yaitu melakukan *unchecked*. Data yang terdapat dalam *database* ditunjukkan dalam Tabel 5.9 dan Tabel 5.10.



Jadwal Pertandingan Tim Z (a)

Nomor Serial : 66A84

TIM LAWAN	TANGGAL
<input checked="" type="checkbox"/> Tim A	4 - 11 - 2011
<input type="checkbox"/> Tim B	12 - 11 - 2011
<input type="checkbox"/> Tim C	19 - 11 - 2011
<input type="checkbox"/> Tim D	25 - 11 - 2011
<input checked="" type="checkbox"/> Tim E	3 - 12 - 2011
<input type="checkbox"/> Tim F	9 - 12 - 2011
<input checked="" type="checkbox"/> Tim G	16 - 12 - 2011
<input type="checkbox"/> Tim H	23 - 12 - 2011
<input type="checkbox"/> Tim I	30 - 12 - 2011
<input type="checkbox"/> Tim J	6 - 1 - 2012
<input type="checkbox"/> Tim K	14 - 1 - 2012
<input type="checkbox"/> Tim L	21 - 1 - 2012
<input type="checkbox"/> Tim M	27 - 1 - 2012
<input type="checkbox"/> Tim N	3 - 2 - 2012
<input type="checkbox"/> Tim O	11 - 2 - 2012
<input checked="" type="checkbox"/> Tim P	18 - 2 - 2012

Simpan Keluar

Jadwal Pertandingan Tim Z (b)

Nomor Serial : 66A84

TIM LAWAN	TANGGAL
<input checked="" type="checkbox"/> Tim A	4 - 11 - 2011
<input type="checkbox"/> Tim B	12 - 11 - 2011
<input type="checkbox"/> Tim C	19 - 11 - 2011
<input type="checkbox"/> Tim D	25 - 11 - 2011
<input type="checkbox"/> Tim E	3 - 12 - 2011
<input type="checkbox"/> Tim F	9 - 12 - 2011
<input checked="" type="checkbox"/> Tim G	16 - 12 - 2011
<input type="checkbox"/> Tim H	23 - 12 - 2011
<input type="checkbox"/> Tim I	30 - 12 - 2011
<input type="checkbox"/> Tim J	6 - 1 - 2012
<input type="checkbox"/> Tim K	14 - 1 - 2012
<input type="checkbox"/> Tim L	21 - 1 - 2012
<input type="checkbox"/> Tim M	27 - 1 - 2012
<input type="checkbox"/> Tim N	3 - 2 - 2012
<input type="checkbox"/> Tim O	11 - 2 - 2012
<input checked="" type="checkbox"/> Tim P	18 - 2 - 2012

Simpan Keluar

(a)

(b)

Gambar 5.12 (a) Tampilan Program Aplikasi Memasukkan Data dengan Memberi Tanda Checklist






(b) Tampilan Program Aplikasi Memasukkan Data dan Menghapus Tanda Checklist

Tabel 5.9 Tabel Jadwal Pertandingan pada Database

Sebelum Data Dihapus

			no_serial	id_pertandingan
<input type="checkbox"/>			66A84	01
<input type="checkbox"/>			66A84	05
<input type="checkbox"/>			66A84	07
<input type="checkbox"/>			66A84	16

Tabel 5.10 Tabel Jadwal Pertandingan pada Database
Setelah Data Dihapus

			no_serial	id_pertandingan
<input type="checkbox"/>			66A84	01
<input type="checkbox"/>			66A84	07
<input type="checkbox"/>			66A84	16

Hasil pengujian yang ada sesuai antara data yang dihapus operator melalui program aplikasi dengan data yang terhapus dari dalam database.

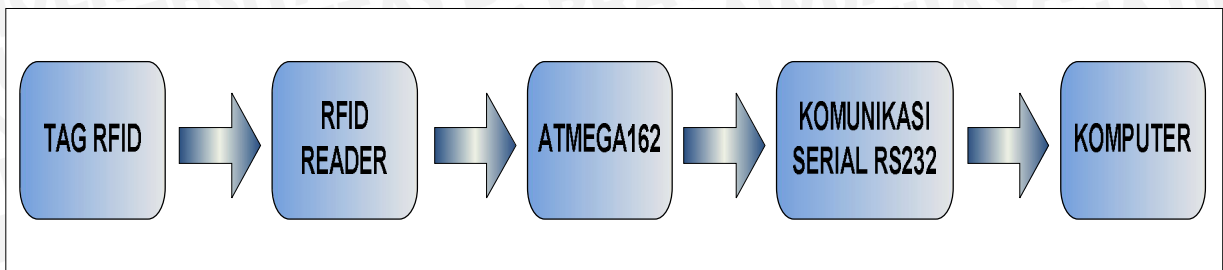
5.2 Pengujian Rangkaian RFID Reader ID-12 dan RS232

Pengujian ini bertujuan untuk mengetahui keberhasilan dari RFID reader untuk membaca data yang terdapat pada tag RFID, serta mengetahui apakah mikrokontroler dapat mengirim data secara serial ke komputer menggunakan komunikasi RS232. Pengujian ini dilakukan dengan bantuan software hyper terminal. Dalam pengujian ini peralatan yang dibutuhkan adalah:

- komputer,
- software hyper terminal,
- tag RFID,
- RFID reader,
- Rangkaian RS232, dan
- Minimum sistem mikrokontroler ATmega162.

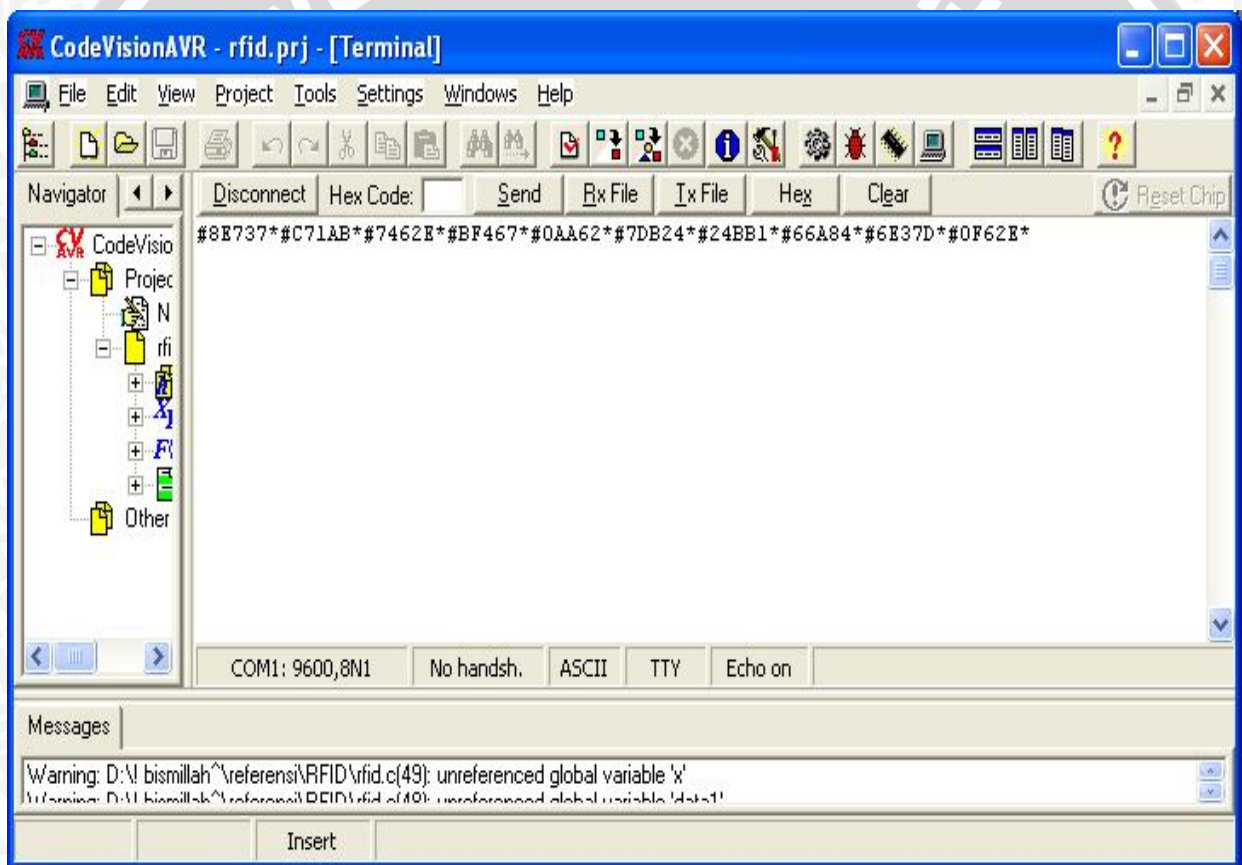
Prosedur pengujian rangkaian RFID ID-12 dan RS232 adalah sebagai berikut:

- Mengatur rangkaian seperti yang ditunjukkan dalam Gambar 5.13.
- Menghubungkan minimum sistem mikrokontroler ATmega162 dengan catu daya sebesar 5 volt.
- Mendekatkan tag RFID ke RFID reader sampai LED dan buzzer aktif.
- Mengamati data yang ditampilkan pada Hyper terminal.



Gambar 5.13 Diagram Blok Pengujian Rangkaian
RFID Reader ID-12 dan RS232

Dari hasil pengujian yang ditunjukkan dalam Gambar 5.14, dapat diamati bahwa masukan yang diberikan oleh RFID reader dapat ditampilkan pada Hyper terminal di komputer. Data yang ditampilkan berupa nomor serial tag RFID dengan format data ASCII. Pengujian ini dilakukan sebanyak 10 kali pendeteksian tag RFID yang berbeda dengan jarak pembacaan yang tetap (1 cm).



Gambar 5.14 Hasil Pengujian Rangkaian
RFID Reader ID-12 dan RS232

Dari hasil pengujian dapat disimpulkan bahwa RFID reader dapat membaca tag RFID dengan tingkat keberhasilan 100%. Rangkaian RFID reader tersebut juga diuji seberapa jauh jangkauan pembacaannya. Hasil pengujian jangkauan RFID reader tanpa penghalang ditunjukkan dalam Tabel 5.11 dan hasil pengujian jangkauan RFID reader

dengan menggunakan penghalang dengan ketebalan 3 mm dan jarak 2 cm dari RFID reader ditunjukkan dalam Tabel 5.12.

Tabel 5.11 Hasil Pengujian Jangkauan RFID Reader

No.	Jarak Pembacaan sudut 90°	Hasil Pembacaan	Jarak Pembacaan sudut 45°	Hasil Pembacaan
1.	1 cm	Terbaca	1 cm	Terbaca
2.	2 cm	Terbaca	2 cm	Terbaca
3.	3 cm	Terbaca	3 cm	Terbaca
4.	4 cm	Terbaca	4 cm	Terbaca
5.	5 cm	Terbaca	5 cm	Terbaca
6.	6 cm	Terbaca	6 cm	Tidak terbaca
7.	7 cm	Terbaca	7 cm	Tidak terbaca
8.	8 cm	Terbaca	8 cm	Tidak terbaca
9.	9 cm	Terbaca	9 cm	Tidak terbaca
10.	10 cm	Tidak terbaca	10 cm	Tidak terbaca

Tabel 5.12 Hasil Pengujian Jangkauan RFID Reader Dengan Menggunakan Penghalang

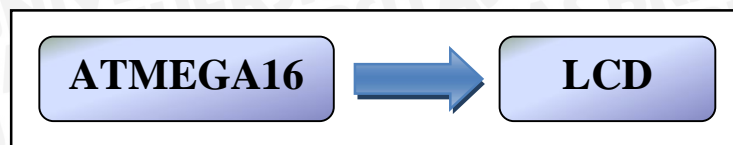
No.	Jarak Pembacaan sudut 90°	Hasil Pembacaan	Jarak Pembacaan sudut 45°	Hasil Pembacaan
1.	1 cm	Terbaca	1 cm	Terbaca
2.	2 cm	Terbaca	2 cm	Terbaca
3.	3 cm	Terbaca	3 cm	Terbaca
4.	4 cm	Terbaca	4 cm	Tidak Terbaca
5.	5 cm	Terbaca	5 cm	Tidak Terbaca
6.	6 cm	Terbaca	6 cm	Tidak terbaca
7.	7 cm	Tidak Terbaca	7 cm	Tidak terbaca

Melalui pengujian jarak pembacaan terlihat bahwa pada saat jarak pembacaan dengan sudut 90° tag RFID dapat terbaca pada jarak maksimum 9 cm dan saat jarak pembacaan dengan sudut pembacaan 45° tag RFID dapat terbaca pada jarak maksimum 5 cm. Sedangkan dengan menggunakan penghalang, tag RFID dapat terbaca pada jarak maksimum 6 cm pada jarak pembacaan dengan sudut 90° dan jarak maksimum 3 cm pada jarak pembacaan dengan sudut 45°.

5.3 Pengujian LCD

Tujuan pengujian modul LCD adalah untuk mengetahui keberhasilan LCD menampilkan tulisan sesuai dengan karakter yang dikirimkan oleh mikrokontroler. Pengujian modul LCD dilakukan dengan menghubungkan LCD dengan mikrokontroler

yang sudah berisi perangkat lunak untuk menampilkan tulisan tertentu. Diagram blok pengujian modul LCD ini ditunjukkan dalam Gambar 5.15.



Gambar 5.15 Diagram Blok Pengujian Tampilan LCD

Pada pengujian ini, mikrokontroler diberikan program agar LCD dapat menampilkan tulisan “Pengujian LCD” pada baris pertama dan tulisan “& Mikrokontroler” pada baris kedua. Kemudian *board* mikrokontroler diaktifkan dengan memberi catu daya sebesar 5 volt. Tampilan hasil pengujian modul LCD ditunjukkan dalam Gambar 5.16 Dari hasil pengujian dapat disimpulkan bahwa rangkaian LCD dapat menampilkan tulisan sesuai dengan karakter yang dikirimkan oleh mikrokontroler.



Gambar 5.16 Hasil Pengujian LCD

5.4 Pengujian Counter sebagai Penentu Nomor Tempat Duduk

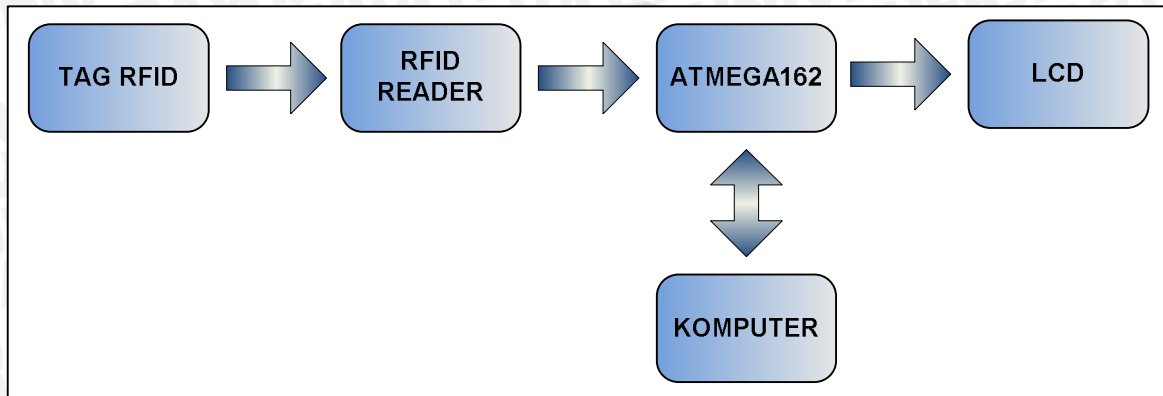
Pengujian ini bertujuan untuk mengetahui keberhasilan *counter* yang telah diprogram melalui mikrokontroler ATmega162 untuk menentukan nomor tempat duduk secara urut. Setelah nomor serial sebuah *tag* dideteksi oleh RFID *reader*, data nomor serial dikirimkan ke mikrokontroler. Kemudian mikrokontroler mengirimkan data tersebut secara serial ke komputer untuk mengeceknya pada *database*. Jika data tersedia pada *database*, komputer akan mengirimkan data ke mikrokontroler sehingga *counter* pada mikrokontroler akan menghitung dan ditampilkan pada LCD.

Peralatan yang dibutuhkan untuk pengujian ini adalah:

- Tag RFID,
- RFID *reader*,
- Minimum sistem mikrokontroler Atmega162,
- Komputer, dan
- LCD.

Prosedur pengujian *counter* adalah sebagai berikut:

- Mengatur rangkaian seperti yang ditunjukkan dalam Gambar 5.17.
- Menghubungkan minimum sistem mikrokontroler Atmega162 dengan catu daya sebesar 5 volt.
- Mendekatkan *tag* RFID ke RFID *reader* sampai LED dan *buzzer* aktif.
- Mengamati tampilan LCD.



Gambar 5.17 Diagram Blok Pengujian *Counter* sebagai Penentu Nomor Tempat Duduk

Hasil pengujian saat kondisi awal, yaitu sebelum diberikan masukan *tag* RFID ditunjukkan dalam Gambar 5.18(a), tampilan LCD setelah diberikan masukan *tag* RFID yang telah didaftarkan pada *database* ditunjukkan dalam Gambar 5.18(b), dan Gambar 5.18(c) yang menunjukkan tampilan LCD setelah diberikan masukan *tag* RFID yang tidak terdaftar pada *database* dikarenakan belum memesan tiket sebelumnya. Dari hasil pengujian dapat diamati jika nomor serial telah terdaftar pada *database*, *counter* akan menghitung dan menampilkan nomor tempat duduk. Sedangkan jika data tidak terdaftar, LCD akan menampilkan tulisan “Tidak Terdaftar”.

Pengujian ini menggunakan 10 buah *tag*, 7 buah *tag* telah didaftarkan nomor serialnya dan 3 buah *tag* belum didaftarkan. Data pengujian *counter* ditunjukkan dalam Tabel 5.13.



(a)

(b)



(c)

- Gambar 5.18 (a) Kondisi Awal Sebelum Ada Masukan *Tag* RFID
 (b) Tampilan LCD Saat Nomor Serial *Tag* RFID Telah Terdaftar di *Database*
 (c) Tampilan LCD Saat Nomor Serial *Tag* RFID Tidak Terdaftar di *Database*

Tabel 5.13 Data Pengujian *Counter* sebagai Penentu Nomor Tempat Duduk

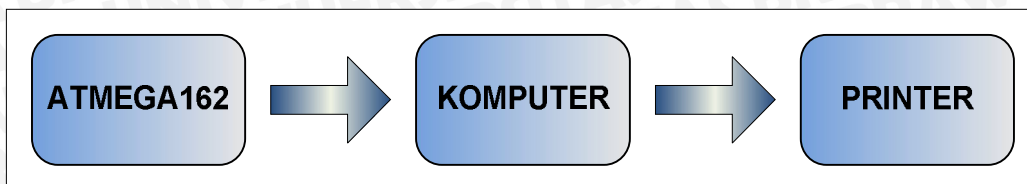
No.	Nomor Serial	Tampilan LCD (No. Tempat Duduk)	Keterangan
1.	7DB24	001	Terdaftar
2.	C71AB	002	
3.	0AA62	TIDAK TERDAFTAR	Tidak Terdaftar
4.	BF467	003	Terdaftar
5.	8E737	004	
6.	66A84	005	
7.	0F62E	TIDAK TERDAFTAR	Tidak Terdaftar
8.	7462E	TIDAK TERDAFTAR	
9.	6E37D	006	Terdaftar
10.	24BB1	007	

Dari hasil pengujian dapat disimpulkan bahwa *counter* dapat berfungsi ketika mikrokontroler mendapatkan data dari komputer. Selain itu, tingkat keberhasilan komunikasi serial antara mikrokontroler dan komputer (*database*) adalah 100%.

5.5 Pengujian Mencetak Data Menggunakan Printer

Pengujian mencetak data menggunakan *printer* bertujuan untuk mengetahui keberhasilan hasil *printout* sesuai dengan perencanaan dan data masukan. Pengujian mencetak data menggunakan *printer* ini dilakukan dengan menghubungkan mikrokontroler

dengan computer dan komputer dengan mesin pencetak (*printer*). Diagram blok pengujian mencetak data menggunakan *printer* ditunjukkan dalam Gambar 5.19.



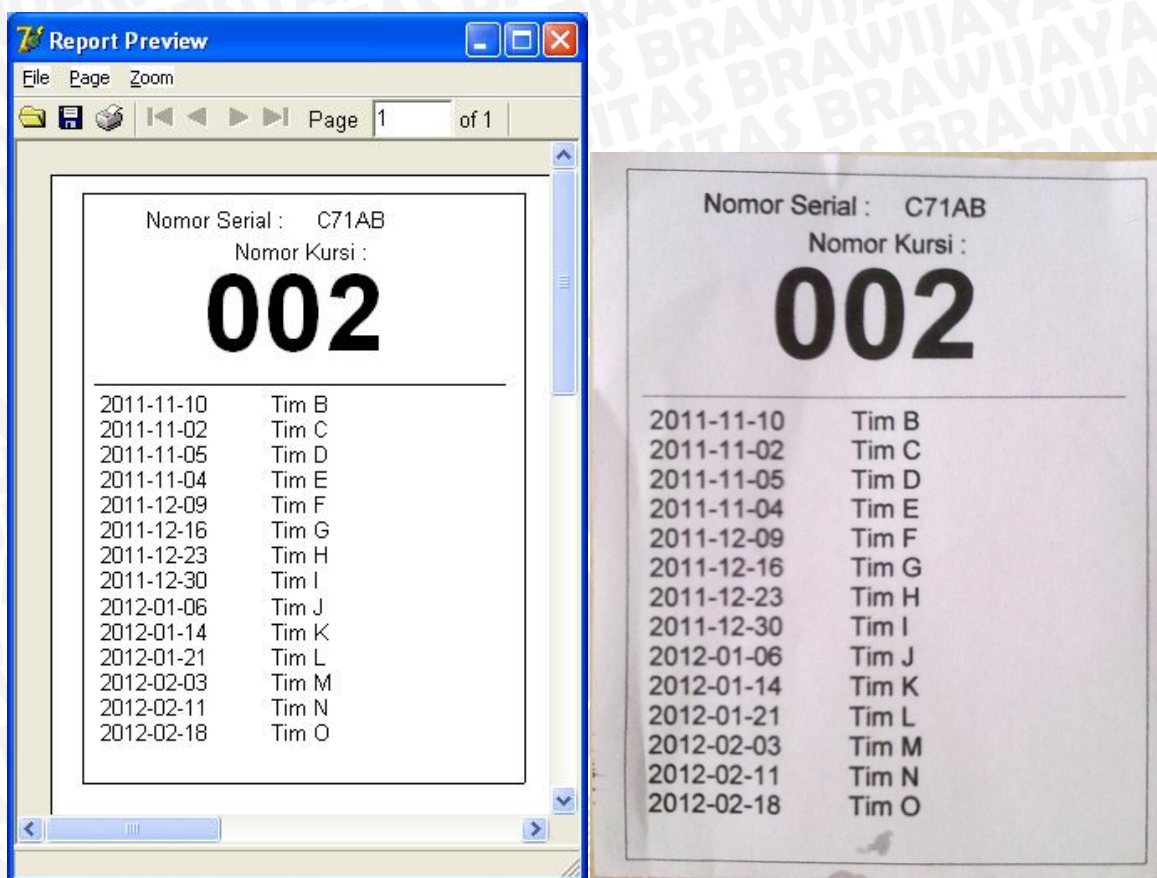
Gambar 5.19 Diagram Blok Pengujian Mencetak Data Menggunakan *Printer*

Data hasil perhitungan *counter*, yaitu berupa nomor tempat duduk, oleh mikrokontroler dikirimkan ke komputer secara serial. Kemudian data tersebut digabungkan dengan data yang terdapat dalam *database*, yaitu data jadwal pertandingan yang telah dipesan. Setelah itu data tersebut digabungkan dengan menggunakan Delphi. Tampilan Delphi yang terdapat pada komputer akan dicetak. Hasil cetak berisi nama, nomor tempat duduk, dan jadwal pertandingan yang telah dipesan.

Tampilan isi program yang akan dicetak ditunjukkan dalam Gambar 5.20. *Preview* data yang akan dicetak ditunjukkan dalam Gambar 5.21(a) dan hasil *printout* data ditunjukkan dalam Gambar 5.21 (b).



Gambar 5.20 Tampilan Isi Program yang Akan Dicetak



(a)

(b)

Gambar 5.21 (a) *Preview* Data yang akan Dicetak
(b) Hasil *Printout* Data

Hasil pengujian cetakan menggunakan *printer* terdapat ketidaksempurnaan dalam proses *printout* data. Hal ini mungkin terjadi karena ketidaktepatan pada ukuran kertas dan dalam meletakkan kertas yang digunakan untuk mencetak.

5.6 Pengujian Rangkaian *Driver* Motor DC

Pengujian ini bertujuan untuk mengetahui keberhasilan kerja dari rangkaian *driver* motor DC yang terdiri dari 2 buah transistor npn tipe 2n2222 dan 2 buah relay tipe HKE. Jika keluaran dari mikrokontroler pada PORTE.0 adalah logika 1 maka motor DC aktif dan pintu akan terbuka, sedangkan jika keluarannya adalah logika 0 maka motor DC tidak aktif. Hal tersebut juga berlaku untuk keluaran mikrokontroler pada PORTE.1. Jika keluarannya adalah logika 1, maka motor DC aktif dan pintu tertutup, sedangkan jika keluarannya adalah logika 0, motor DC tidak aktif.

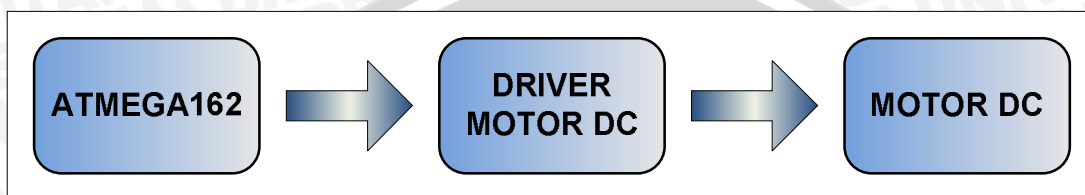
Peralatan yang dibutuhkan untuk melakukan pengujian rangkaian *driver* motor DC adalah:

- Minimum sistem mikrokontroler Atmega162,
- Rangkaian *driver* motor DC, dan

- Motor DC

Prosedur pengujian rangkaian *driver* motor DC adalah sebagai berikut:

1. Mengatur rangkaian seperti yang ditunjukkan dalam Gambar 5.22.
2. Menghubungkan minimum sistem mikrokontroler Atmega162 dengan catu daya sebesar 5 volt dan relay dengan catu daya sebesar 12 volt.
3. Memberikan masukan pada rangkaian *driver*.
4. Mengamati perubahan pada pintu.



Gambar 5.22 Diagram Blok Pengujian Rangkaian *Driver* Motor DC

Tabel 5.14 Hasil Pengujian Rangkaian *Driver* Motor DC

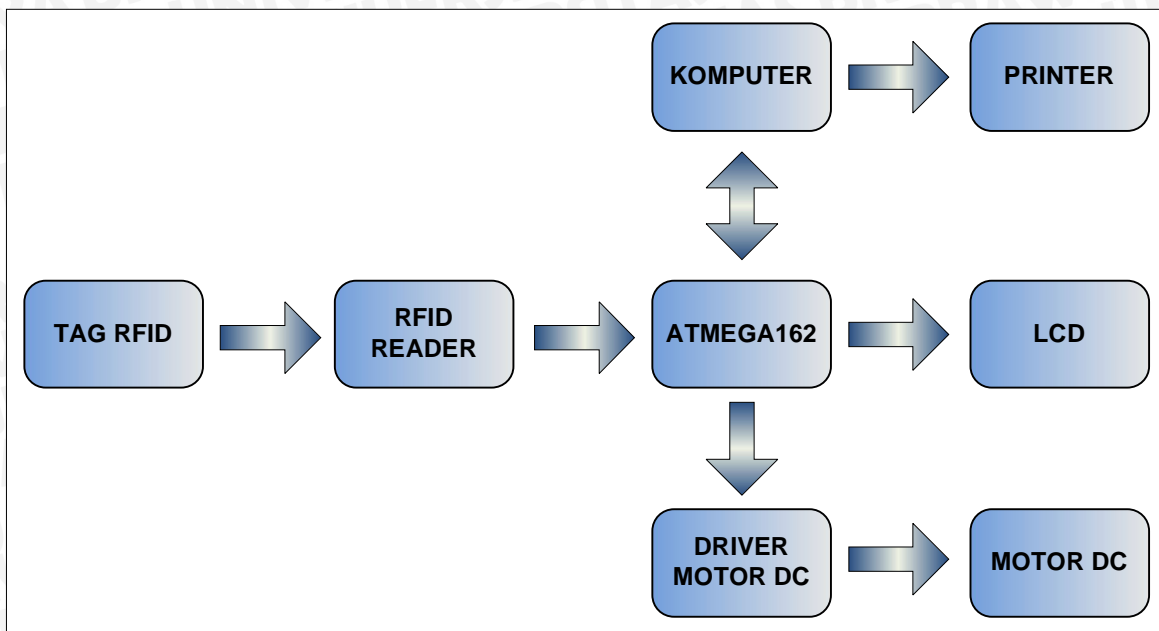
Mikrokontroler	Masukan	Motor DC	Pintu
PORTE.0	0 volt	Tidak aktif	Diam
	5 volt	Aktif	Terbuka
PORTE.1	0 volt	Tidak aktif	Diam
	5 volt	Aktif	Tertutup

Dari hasil pengujian yang ditunjukkan dalam Tabel 5.14, pada saat masukan berlogika 0, maka arus akan masuk ke mikrokontroler dan pada saat itu pula basis transistor tidak mendapat arus, sehingga transistor akan mengalami *cut-off*. Pada keadaan seperti itu, relay tidak akan aktif dikarenakan transistor tersebut seperti sebuah saklar yang terbuka. Karena relay tidak aktif, maka motor DC juga tidak aktif. Sedangkan saat masukan berlogika 1, basis transistor akan mendapat arus yang berasal dari resistor pull-up dan mikrokontroler, maka mikrokontroler akan mengalami keadaan saturasi. Relay akan aktif dikarenakan transistor tersebut seperti sebuah saklar tertutup. Karena relay aktif, maka motor DC juga aktif dan keadaan pintu dapat terbuka atau tertutup.

5.7 Pengujian Rangkaian Keseluruhan Sistem

Pengujian sistem secara keseluruhan dilakukan dengan cara menggabungkan semua bagian alat yang dibuat dan melihat kinerja alat. Pengujian ini bertujuan untuk mengetahui kinerja alat yang dibuat apakah telah bekerja sesuai dengan yang diharapkan pada perancangan. Peralatan yang digunakan adalah alat yang telah selesai dirancang dilengkapi

dengan catu daya DC 12 volt. Diagram blok pengujian rangkaian keseluruhan sistem ditunjukkan dalam Gambar 5.23.



Gambar 5.23 Diagram Blok Pengujian Rangkaian Keseluruhan Sistem

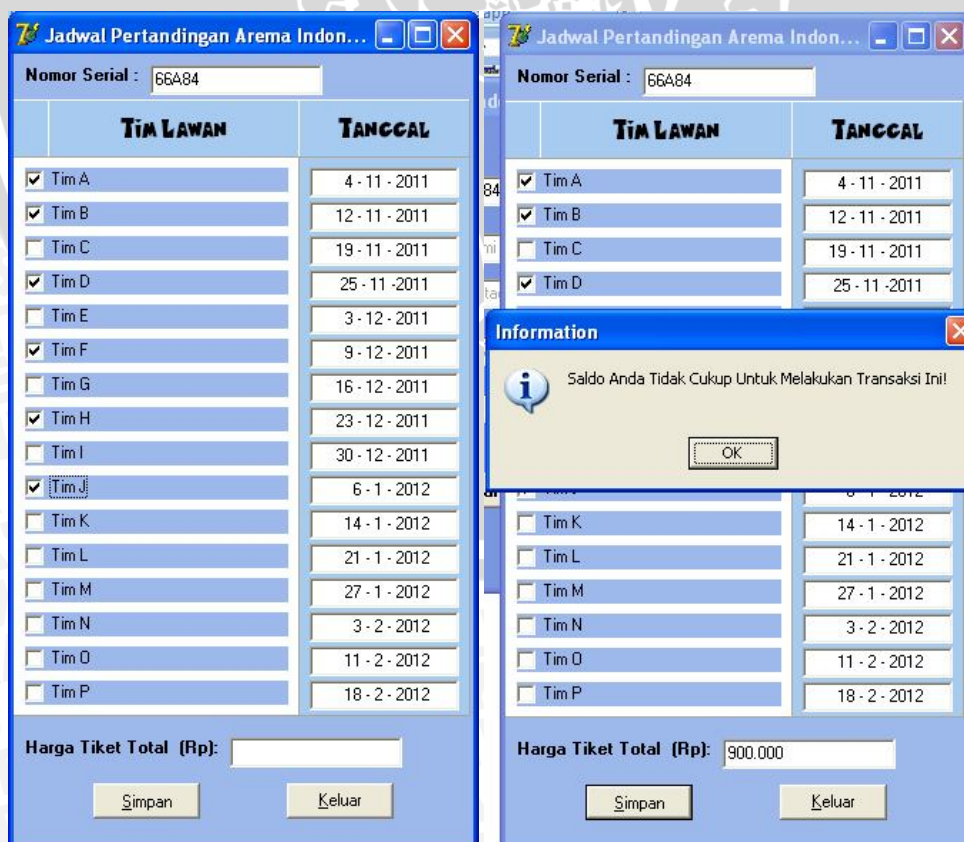
Pengujian sistem secara keseluruhan dilakukan sebanyak 5 kali pengujian dengan data yang berbeda. Pengujian secara keseluruhan diawali dengan pemesanan tiket di hari sebelum hari pertandingan, yaitu pelanggan akan mendatangi *customer service* untuk memesan tiket dan pembayaran dilakukan dengan memotong saldo yang terdapat pada kartu secara otomatis. Pengujian dilakukan sebanyak tiga kali, yaitu saat operator salah memasukkan nomor kartu, saat saldo pada kartu tersebut tidak cukup untuk melakukan transaksi, dan saat saldo kartu mencukupi. Hasil pengujian ditunjukkan dalam Gambar

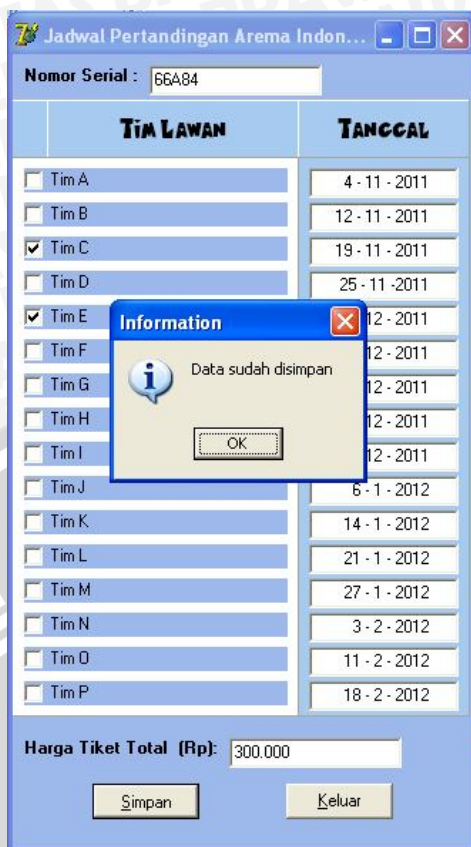
Selanjutnya dilakukan pengujian dengan memasukkan tanggal pada hari tersebut untuk mengetahui tim mana yang akan melawan tim tuan rumah. Setelah itu menghubungkan program aplikasi pada komputer dengan rangkaian mikrokontroler. Setelah terhubung, dilakukan pengujian dengan cara mendekatkan *tag* RFID dengan RFID *reader*. Jika nomor serial *tag* RFID terdapat dalam *database*, LCD akan menampilkan nomor tempat duduk hasil penghitungan *counter*. Jika tidak terdaftar, LCD akan menampilkan tulisan 'KARTU ANDA TIDAK TERDAFTAR'. Data nomor tempat duduk tersebut dikirimkan ke komputer. Data tersebut digabungkan dengan data jadwal pertandingan yang telah dipesan dalam format tertentu dan dicetak menggunakan *printer*. Setelah itu palang pintu terbuka selama 5 detik dan secara otomatis palang pintu akan tertutup.

Dalam pengujian ini dilakukan pengisian tanggal sebanyak dua kali, yaitu ketika tanggal yang dimasukkan tidak sesuai dengan tanggal hari tersebut dan tanggal yang dimasukkan sesuai. Selain itu dalam pengujian ini digunakan 5 buah *tag* RFID yang berbeda dan diasumsikan 3 buah *tag* telah terdaftar dan 2 buah *tag* belum terdaftar. Hasil pengujian sistem secara keseluruhan berupa tampilan program aplikasi saat pengisian tanggal yang tidak sesuai ditunjukkan dalam Gambar 5.24 (a) dan Gambar 5.24 (b) menunjukkan tampilan program aplikasi saat pengisian tanggal yang sesuai, tampilan LCD ditunjukkan dalam Gambar 5.25 (a), (b), (c), (d), (e), tampilan hasil cetakan *printer* ditunjukkan dalam Gambar 5.26 (a), (b), (c), dan data hasil pengujian terhadap palang pintu ditunjukkan dalam Tabel 5.15.



Gambar 5.24 (a) Pengisian Nomor Serial yang Tidak Sesuai
(b) Pengisian Nomor Serial yang Sesuai





Gambar 5.25 (a) Pemesanan Enam Tiket Pertandingan
 (b) Saldo pada Kartu Tidak Cukup untuk Melakukan Transaksi
 (c) Saldo pada Kartu Mencukupi untuk Pemesanan Dua Tiket



Gambar 5.26 Saldo Secara Otomatis Berkurang



(a)



(b)

Gambar 5.27 (a) Pengisian Tanggal yang Tidak Sesuai pada Program Aplikasi
 (b) Pengisian Tanggal yang Sesuai pada Program Aplikasi



(a)



(b)



(c)



(d)



(e)

Gambar 5.28 (a) Tampilan LCD Pengujian Sistem Secara Keseluruhan dengan Nomor Serial 7DB24

(b) Tampilan LCD Pengujian Sistem Secara Keseluruhan dengan Nomor Serial C71AB

(c) Tampilan LCD Pengujian Sistem Secara Keseluruhan dengan Nomor Serial 7462E

(d) Tampilan LCD Pengujian Sistem Secara Keseluruhan dengan Nomor Serial BF467

(e) Tampilan LCD Pengujian Sistem Secara Keseluruhan dengan Nomor Serial 24BB1

Nomor Serial : 7DB24
Nomor Kursi :

001

2011-11-10	Tim B
2011-11-02	Tim C
2011-11-05	Tim D
2011-11-04	Tim E
2011-12-09	Tim F
2011-12-16	Tim G
2011-12-23	Tim H
2011-12-30	Tim I
2012-01-06	Tim J
2012-01-14	Tim K
2012-01-21	Tim L
2012-02-03	Tim M
2012-02-11	Tim N
2012-02-18	Tim O

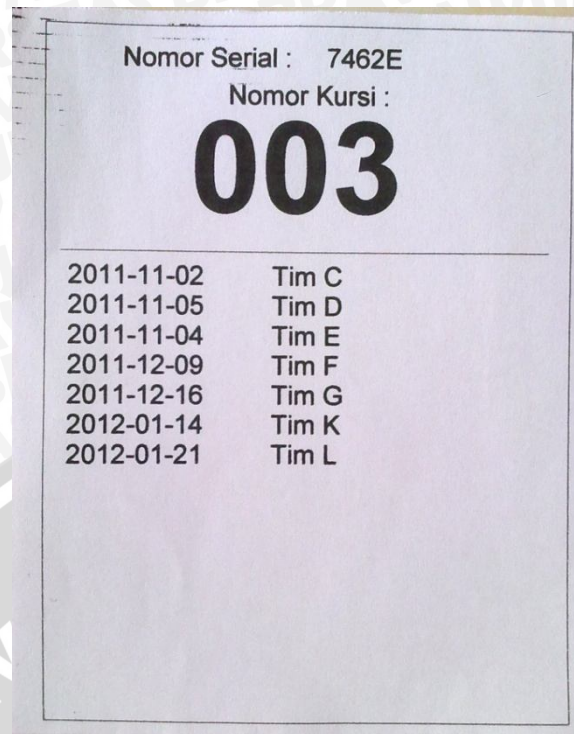
(a)

Nomor Serial : C71AB
Nomor Kursi :

002

2011-11-10	Tim B
2011-11-02	Tim C
2011-11-05	Tim D
2011-11-04	Tim E
2011-12-09	Tim F
2011-12-16	Tim G
2011-12-23	Tim H
2011-12-30	Tim I
2012-01-06	Tim J
2012-01-14	Tim K
2012-01-21	Tim L
2012-02-03	Tim M
2012-02-11	Tim N
2012-02-18	Tim O

(b)



(c)

Gambar 5.29 (a) Hasil Cetak Data dengan Nomor Serial 7DB24

(b) Hasil Cetak Data dengan Nomor Serial C71AB

(c) Hasil Cetak Data dengan Nomor Serial BF467

Tabel 5.15 Hasil Pengujian Terhadap Palang Pintu

Nomor Serial	Palang Pintu
7DB24	Terbuka
C71AB	Terbuka
7462E	Tidak Terbuka
BF467	Terbuka
24BB1	Tidak Terbuka

Dari hasil pengujian *sistem* secara keseluruhan didapatkan hasil sesuai dengan perancangan yang dibuat. Dimulai dari pencarian data, pengubahan data, data masukan, data tersimpan dalam *database*, membandingkan data masukan tanggal, identifikasi nomor serial pada tag RFID dengan RFID *reader*, komunikasi serial antara mikrokontroler dengan komputer, data dapat dicetak hingga motor DC dapat membuka dan menutup palang pintu.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

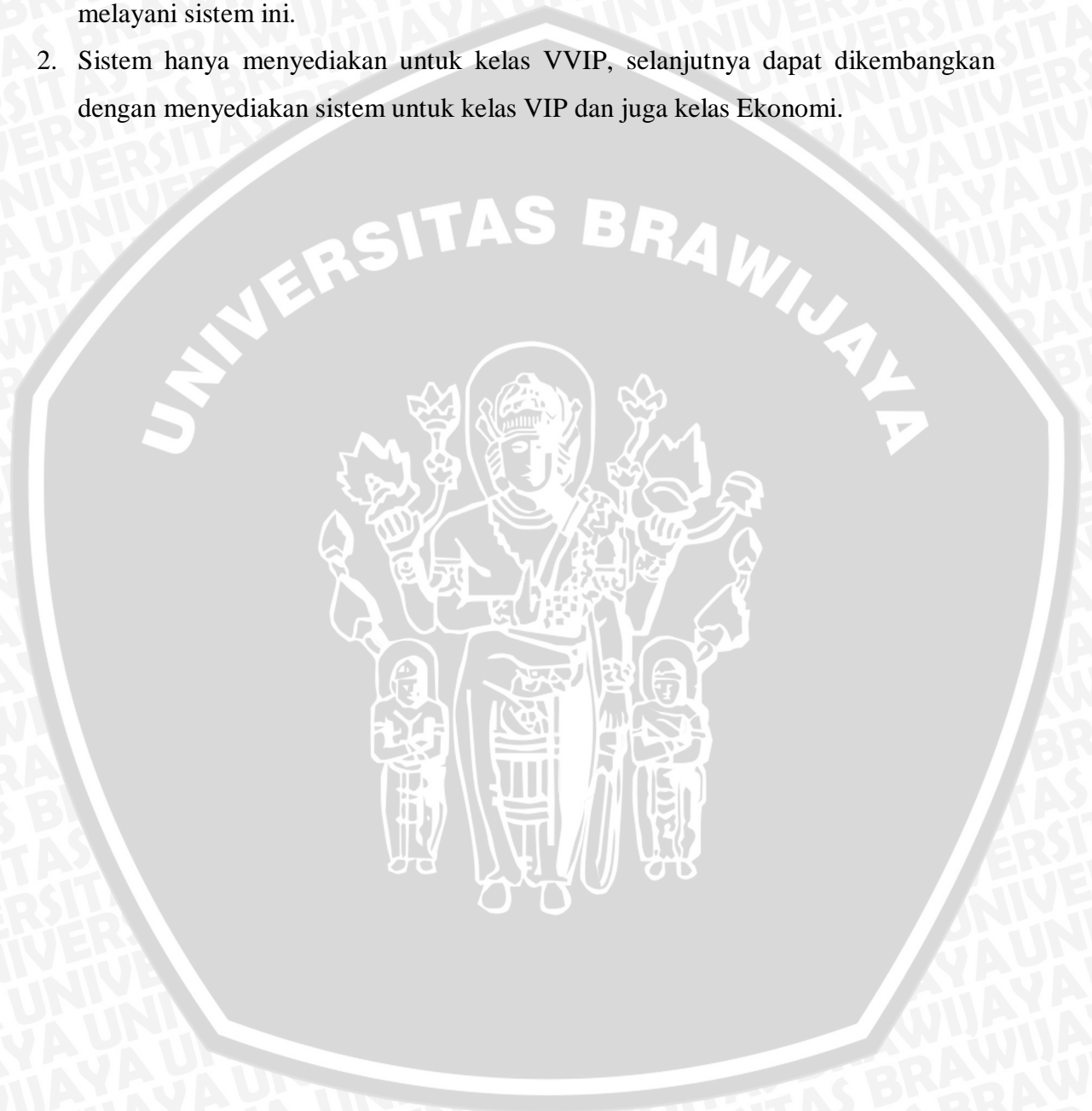
Berdasarkan hasil pengujian tiap bagian dan keseluruhan sistem yang telah dilakukan, didapat kesimpulan sebagai berikut:

1. RFID ID-12 dapat digunakan sebagai sensor identifikasi karena kombinasinya yang berbeda satu sama lain. Sensor ini memberi masukan kepada Mikrokontroler berupa kode ASCII dan kemudian diolah sebagai data oleh Mikrokontroler. RFID ID-12 ini dapat bekerja dengan baik pada jarak pembacaan ± 9 cm saat Tag RFID didekatkan tegak lurus dari RFID Reader, dan ± 5 cm saat posisi Tag RFID bersudut $\pm 45^\circ$ dari posisi RFID Reader tanpa penghalang. Jika menggunakan penghalang berbahan plastik dengan ketebalan ± 3 mm dan jarak ± 2 cm, RFID reader dapat bekerja dengan baik pada jarak pembacaan ± 6 cm saat Tag RFID didekatkan tegak lurus dari RFID Reader, dan ± 3 cm saat posisi Tag RFID bersudut $\pm 45^\circ$. RFID reader tipe ID-12 telah dapat membaca tag RFID dengan tingkat keberhasilan 90%.
2. Pengolah data yang utama pada sistem ini adalah Mikrokontroler Atmega162. Antarmuka mikrokontroler dengan komputer menggunakan TX, RX dan GND yang ada dalam mikrokontroler dengan menggunakan *boudrate* 9600 bps sudah berjalan dengan baik. Mikrokontroler dapat menerima data dari komputer sehingga data tersebut menjadi pengendali driver untuk mengaktifkan motor DC. Mikrokontroler juga dapat mengirimkan data ke komputer sehingga data tersebut dapat digabungkan dengan beberapa data dari *database* yang selanjutnya akan mengaktifkan *printer* dan mencetak data tersebut.
3. Perangkat lunak mikrokontroler Atmega162 menggunakan bahasa C telah dapat menangani RFID reader, LCD, komunikasi serial, dan driver motor DC dengan baik. Mikrokontroler telah dapat mengolah data masukan dari RFID reader mengirim dan menerima data dari komputer menggunakan komunikasi serial, menampilkan data pada LCD, dan dapat mengaktifkan motor DC.

6.2. Saran

Saran-saran dalam pengimplementasian maupun peningkatan kerja sistem ini dapat diuraikan sebagai berikut:

1. Sistem yang dirancang hanya menyediakan satu pintu masuk, sehingga dalam pengembangannya diharapkan dapat menyediakan dua atau lebih pintu masuk yang melayani sistem ini.
2. Sistem hanya menyediakan untuk kelas VVIP, selanjutnya dapat dikembangkan dengan menyediakan sistem untuk kelas VIP dan juga kelas Ekonomi.



DAFTAR PUSTAKA

- AB, MySQL. 2006. *Administrator's Guide and Language Reference*. Seattle:MySQL Press
- Ali, Muhammad. 2009. *Membuat Laporan Menggunakan Rave*. Yogyakarta: Teknik Elektro Universitas Negeri Yogyakarta
- Anonim. 2008. *Komunikasi Serial RS232*. http://www.ittelkom.ac.id/library/index.php?view=article&catid=Amikroprocessorkontroller&id=288%3Akomunikasi-serial-rs232&option=com_content&Itemid=15. Diakses tanggal 30 Maret 2011.
- Atmel. 2009. *ATMega162 Datasheet*. San Jose: Atmel Corporation.
- Delphi Indonesia. 2009. *Koneksi Delphi dengan MySQL*. www.delphi.co.id, diakses tanggal 5 Oktober 2011.
- Dunia Script. 2011. *Pengertian Tentang Xampp*. <http://www.blogduniascript.com>, diakses tanggal 20 September 2011.
- Innovations, ID. 2005. *ID Series Datasheet*. <http://www.alldatasheet.com>. Diakses pada tanggal 31 Maret 2011.
- Innovative Electronics. 2008. *RFID Starter Kit*. http://www.innovativeelectronics.com/-innovative_electronics, diakses tanggal 17 Juli 2011.
- Kadir, Abdul. 2004. *Dasar Aplikasi Database MySQL Delphi*. Yogyakarta: Andi.
- Maxim. 2003. *±15kV ESD-Protected, +5V RS-232 Transceivers*. www.alldatasheet.com/datasheet-pdf/pdf/73074/MAXIM/MAX232.html. Diakses pada tanggal 28 September 2011.
- Multicomp. 2009. *HRS4 Series PCB Power Relay*. Multicomp Corporation.
- Pawestri, Yulita Ika. 2010. *Aplikasi RFID dan RTC dalam Perhitungan Biaya pada Sistem Parkir Progresif*. Malang: Teknik Elektro Universitas Brawijaya
- Philips Semiconductor. 1997. *2N2222 NPN Switching Transistors*. Philips Corporation.
- Pranata, Antony. 2002. *Pemrograman Borland Delphi, Edisi Ketiga*. Yogyakarta: Andi.

Sucy. 2010. *Membuat Database di Delphi dengan Paradoks*. <http://suci-yapat.blogspot.com>.

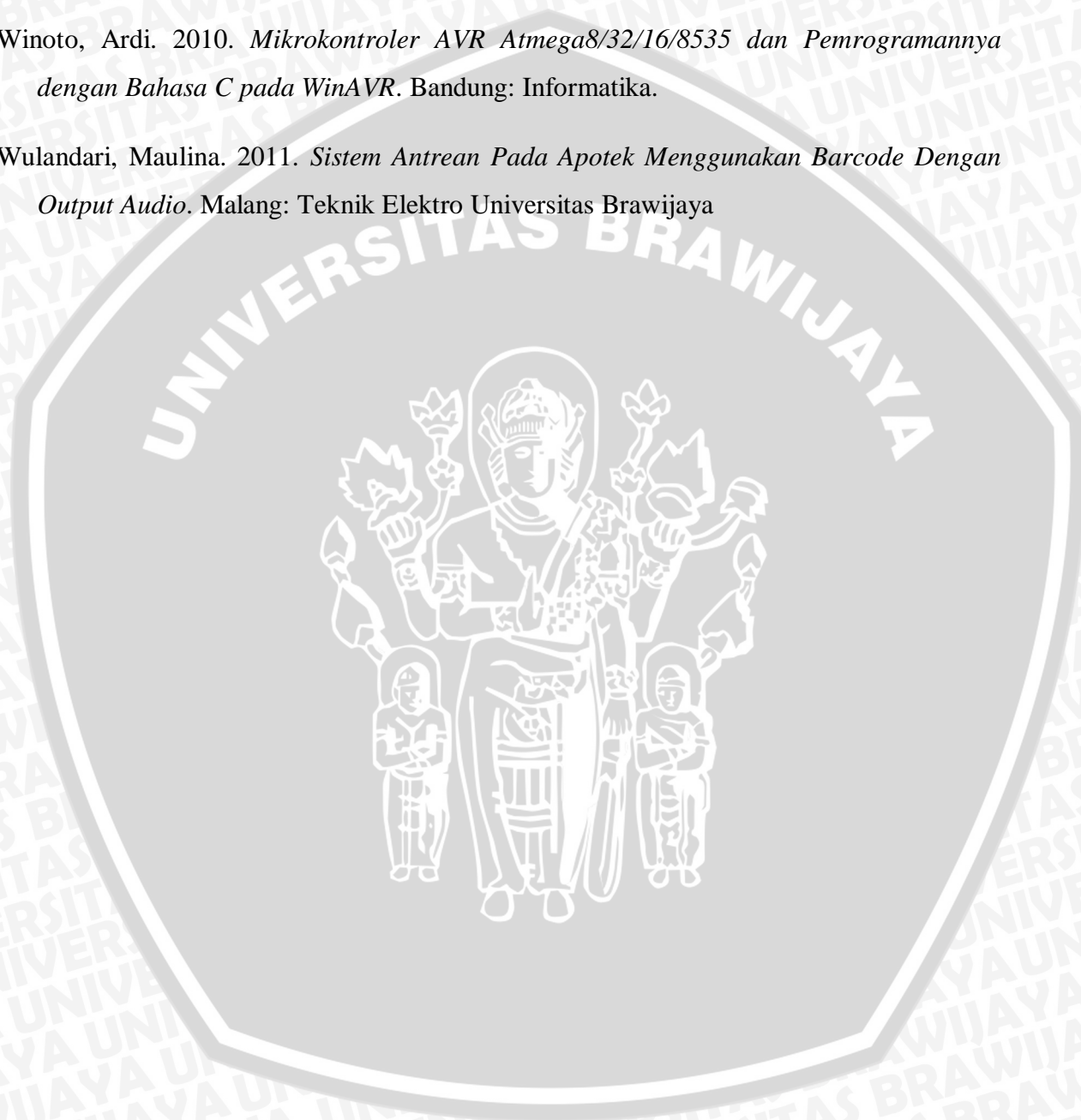
Diakses tanggal 6 April 2011.

Wicaksana, Pandu. 2009. *How To Use Innovations ID-12 RFID Tag Reader*.

<http://panduwicaksana.blogspot.com/2009/09/how-to-use-innovations-id-12-rfid-tag.html>. Diakses tanggal 3 April 2011.

Winoto, Ardi. 2010. *Mikrokontroler AVR Atmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*. Bandung: Informatika.

Wulandari, Maulina. 2011. *Sistem Antrean Pada Apotek Menggunakan Barcode Dengan Output Audio*. Malang: Teknik Elektro Universitas Brawijaya



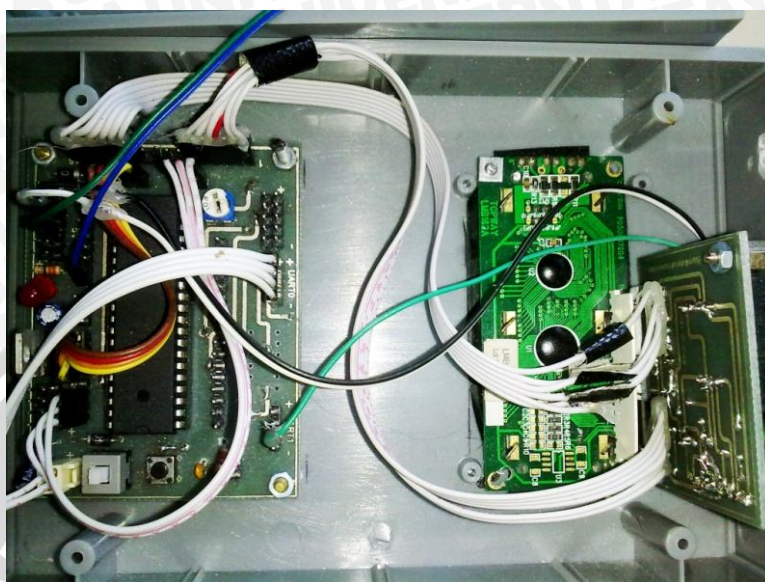
LAMPIRAN



LAMPIRAN I

FOTO ALAT





Gambar 1 Rangkaian di dalam sistem



Gambar 2 Sistem secara keseluruhan (tampak depan)



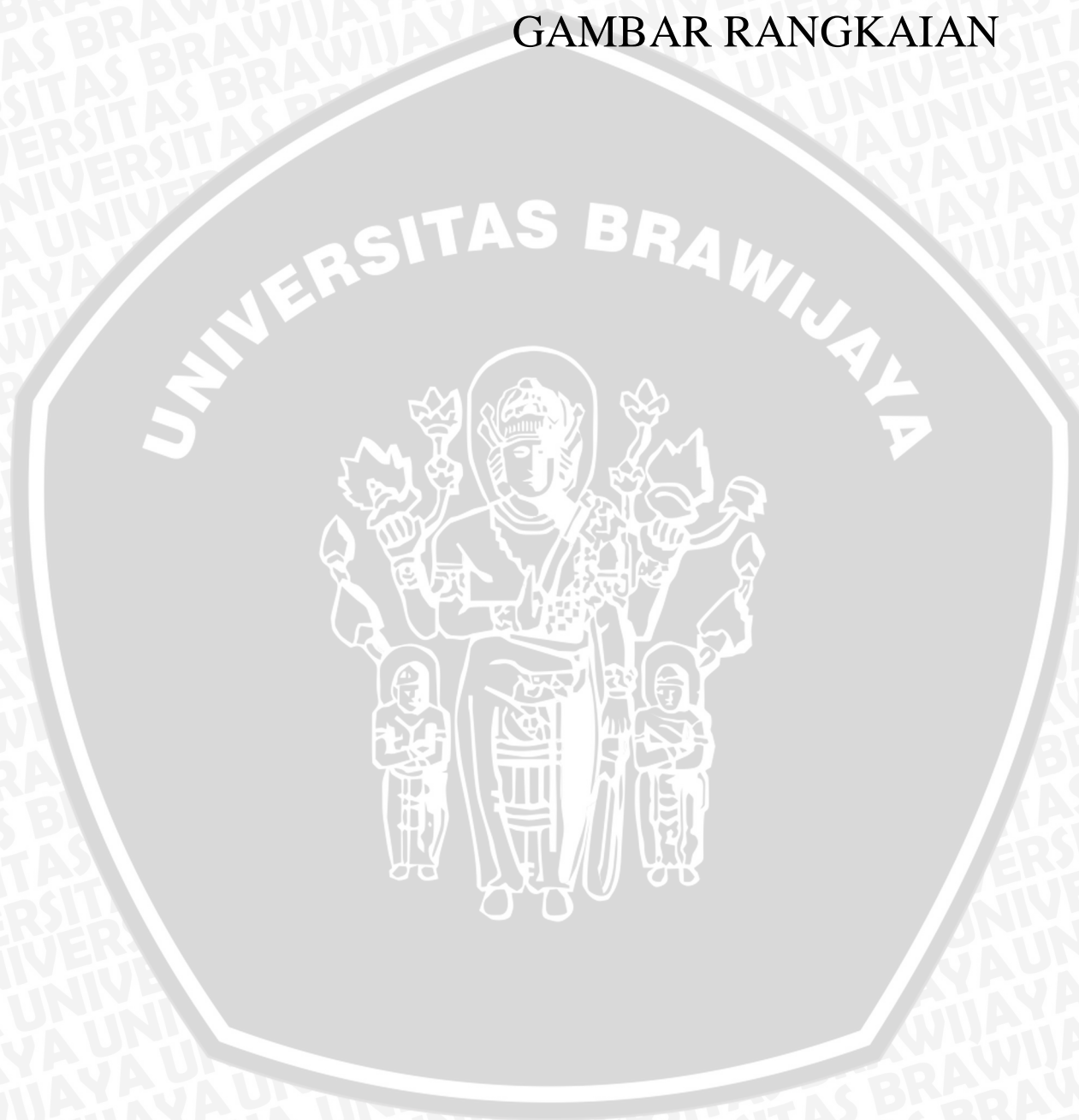
Gambar 3 Sistem secara keseluruhan (tampak belakang)

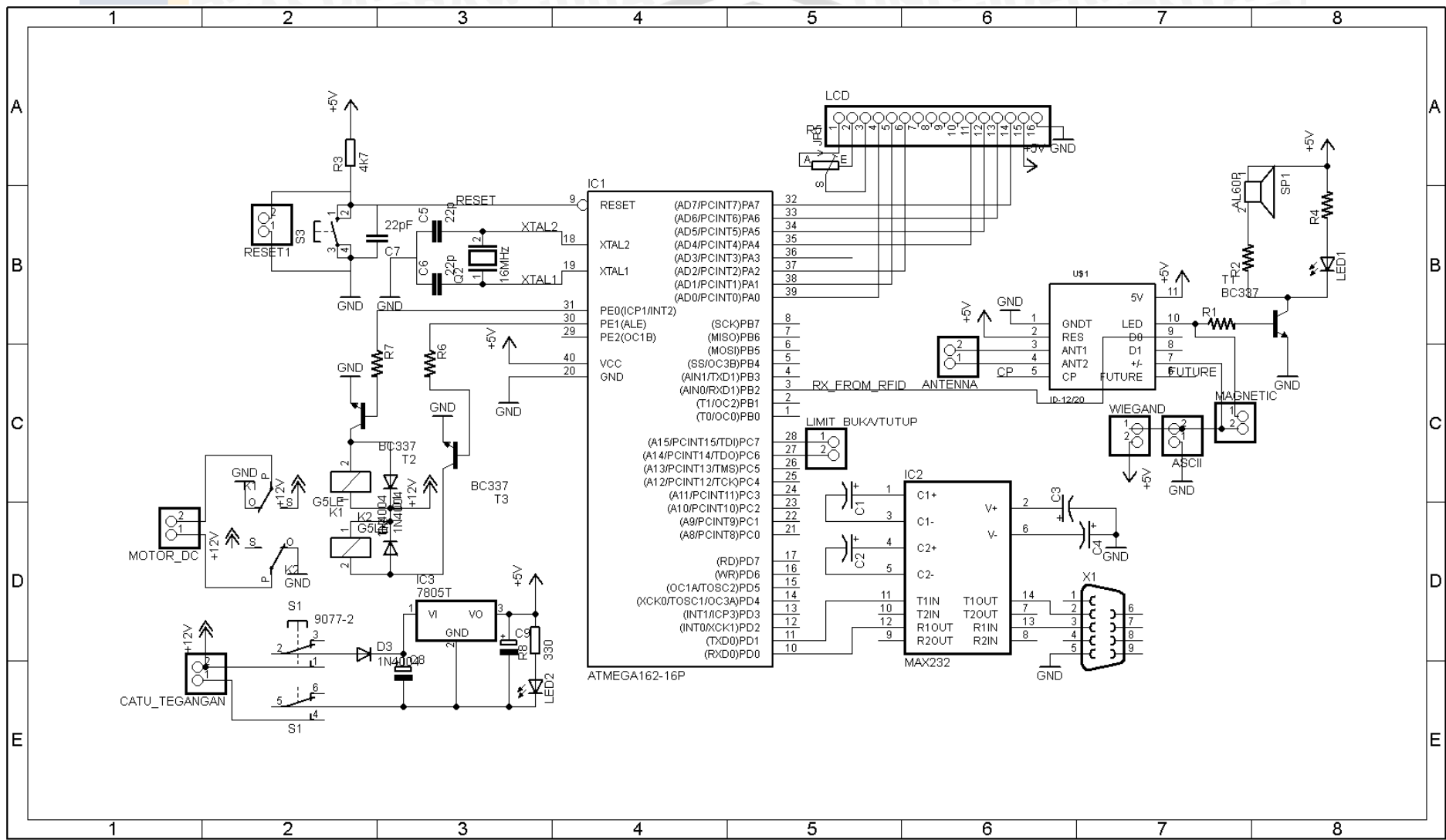


Gambar 4 Sistem saat palang pintu terbuka

LAMPIRAN II

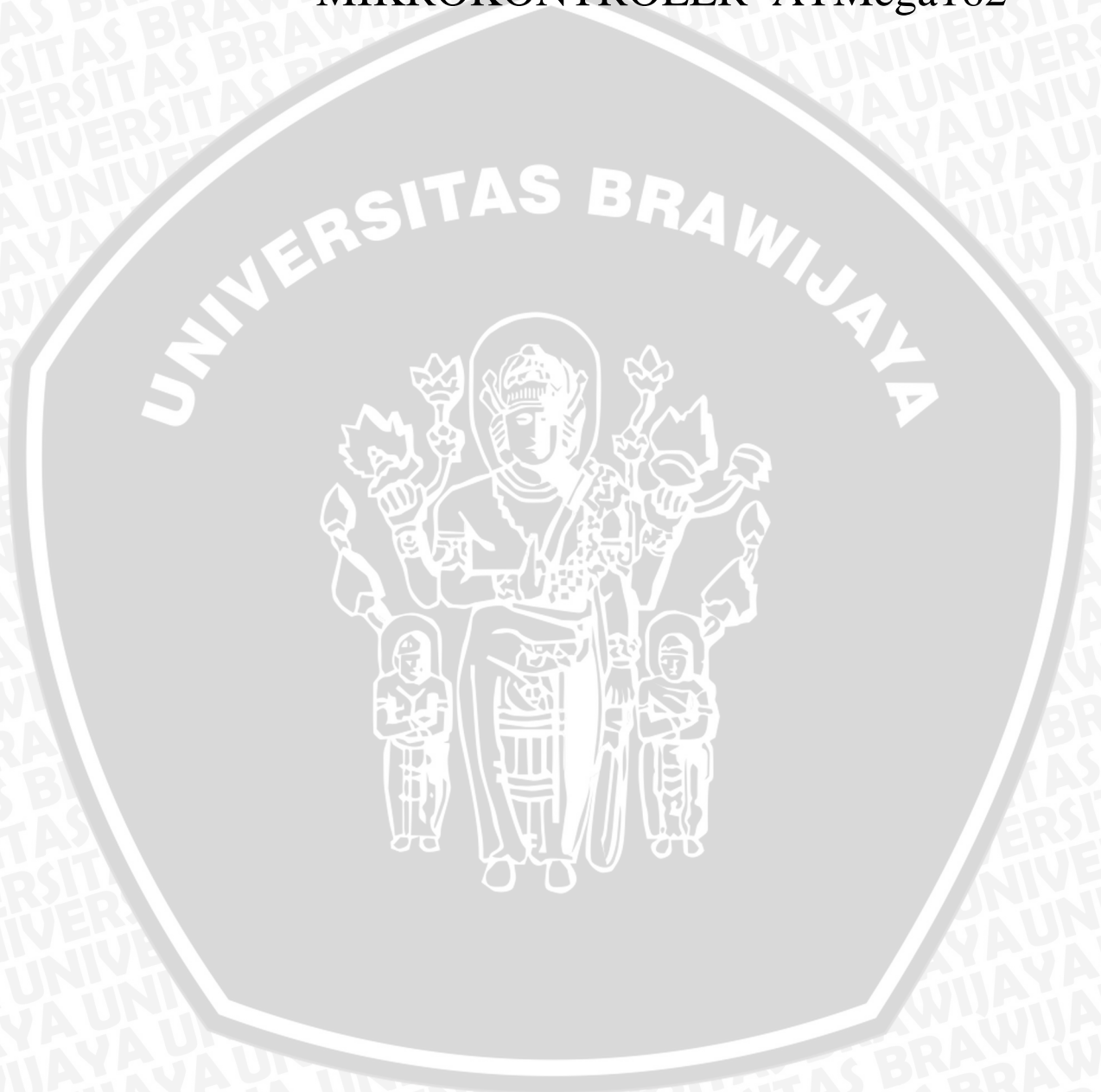
GAMBAR RANGKAIAN





LAMPIRAN III

LISTING PROGRAM MIKROKONTROLER ATmega162



```
/*  
This program was produced by the  
CodeWizardAVR V1.24.8d Professional  
Automatic Program Generator  
© Copyright 1998-2006 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project : My Skripsyong  
Version :  
Date : 12/11/2011  
Author : @doyotcore  
Company : e'07  
Comments:
```

```
Chip type : ATmega162  
Program type : Application  
Clock frequency : 16.000000 MHz  
Memory model : Small  
External SRAM size : 0  
Data Stack size : 256  
*/
```

```
#include <mega162.h>  
#include <stdio.h>  
#include <delay.h>
```

```
// Alphanumeric LCD Module functions  
#asm  
.equ __lcd_port=0x1B ;PORTA  
#endasm  
#include <lcd.h>
```

```
#define RXB8 1  
#define TXB8 0  
#define UPE 2  
#define OVR 3  
#define FE 4  
#define UDRE 5  
#define RXC 7
```

```
#define FRAMING_ERROR (1<<FE)  
#define PARITY_ERROR (1<<UPE)  
#define DATA_OVERRUN (1<<OVR)  
#define DATA_REGISTER_EMPTY (1<<UDRE)  
#define RX_COMPLETE (1<<RXC)
```

```
char data[15];  
unsigned char i,data2[16],data3,data4,data5,data6,data7,data8;  
int counter,a,b,c,d;  
int v,u,w,y;
```

```
// USART0 Receiver buffer  
#define RX_BUFFER_SIZE0 8  
char rx_buffer0[RX_BUFFER_SIZE0];
```

```
#if RX_BUFFER_SIZE0<256  
unsigned char rx_wr_index0,rx_rd_index0,rx_counter0;  
#else  
unsigned int rx_wr_index0,rx_rd_index0,rx_counter0;  
#endif
```

```
// This flag is set on USART0 Receiver buffer overflow  
bit rx_buffer_overflow0;
```

```
// USART0 Receiver interrupt service routine  
interrupt [USART0_RXC] void usart0_rx_isr(void)
```

```

{
char status,data;
status=UCSR0A;
data=UDR0;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
{
rx_buffer0[rx_wr_index0]=data;
if (++rx_wr_index0 == RX_BUFFER_SIZE0) rx_wr_index0=0;
if (++rx_counter0 == RX_BUFFER_SIZE0)
{
rx_counter0=0;
rx_buffer_overflow0=1;
};
if (data==0xFF || ++rx_wr_index0 == RX_BUFFER_SIZE0)
{
rx_wr_index0=0;
if(rx_buffer0[0]!='#' && rx_buffer0[7]!='*')
{
rx_buffer_overflow0=1;
data3=(int)rx_buffer0[1];
data4=(int)rx_buffer0[2];
data5=(int)rx_buffer0[3];
data6=(int)rx_buffer0[4];
data7=(int)rx_buffer0[5];
data8=(int)rx_buffer0[6];
}
};
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART0 Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter0==0);
data=rx_buffer0[rx_rd_index0];
if (++rx_rd_index0 == RX_BUFFER_SIZE0) rx_rd_index0=0;
#asm("cli")
--rx_counter0;
#asm("sei")
return data;
}
#pragma used-
#endif

// Get a character from the USART1 Receiver
#pragma used+
char getchar1(void)
{
char status,data;
while (1)
{
while (((status=UCSR1A) & RX_COMPLETE)==0);
data=UDR1;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
return data;
};
}
#pragma used-

// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here

```

```
void main(void)
{
// Declare your local variables here

// Crystal Oscillator division factor: 1
#pragma optsize-
CLKPR=0x80;
CLKPR=0x00;
#ifdef _OPTIMIZE_SIZE_
#pragma optsize+
#endif

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=P State6=P State5=P State4=T State3=T State2=T State1=T State0=T
PORTC=0xE0;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Port E initialization
// Func2=Out Func1=Out Func0=Out
// State2=0 State1=0 State0=0
PORTE=0x00;
DDRE=0x07;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
```

```
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

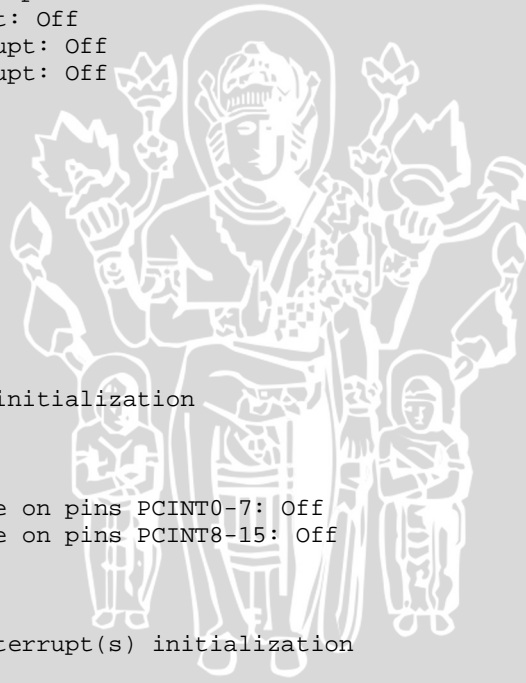
// Timer/Counter 3 initialization
// Clock value: Timer 3 Stopped
// Mode: Normal top=FFFFh
// Noise Canceler: Off
// Input Capture on Falling Edge
// OC3A output: Discon.
// OC3B output: Discon.
// Timer 3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
// Interrupt on any change on pins PCINT0-7: Off
// Interrupt on any change on pins PCINT8-15: Off
MCUCR=0x00;
EMUCR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
ETIMSK=0x00;

// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: On
// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud rate: 9600
UCSR0A=0x00;
UCSR0B=0x98;
UCSR0C=0x86;
UBRR0H=0x00;
UBRR0L=0x67;

// USART1 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART1 Receiver: On
// USART1 Transmitter: Off
```



```
// USART1 Mode: Asynchronous
// USART1 Baud rate: 9600
UCSR1A=0x00;
UCSR1B=0x10;
UCSR1C=0x86;
UBRR1H=0x00;
UBRR1L=0x67;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;

// LCD module initialization
lcd_init(16);

// Global enable interrupts
#asm("sei")

counter=1;

while (1)
{
    // Tampilan LCD pada kondisi awal
    lcd_gotoxy(0,0);
    lcd_putsf("Silakan dekatkan");
    lcd_gotoxy(0,1);
    lcd_putsf("    kartu Anda    ");

    w=0;
    u=0;
    y=0;

    // Mengambil data berupa nomor serial dari tag RFID
    i=0;
    data[i]=getchar1();
    if (data[i]==02)
    {
        for (i=0;i<15;i++){data[i]=getchar1();}
    };

    // Mengirim nomor serial ke komputer
    printf("#%c%c%c%c*c",data[7],data[8],data[9],data[10],data[11]);

    lcd_clear();
    lcd_gotoxy(0,0); lcd_putsf("Mencari Data....");
    w=1;

    while(w==1)
    {
        delay_ms(5000); // Menunggu data yang dikirim dari komputer
        // Jika data dari komputer = 1 (nomor serial terdaftar)
        if(PINC.5==0)
        {
            // Counter aktif (nomor tempat duduk)
            a=counter/100;
            b=counter%100;
            c=b/10;
            d=b%10;

            // Mengirim nomor tempat duduk ke komputer
            printf("%c%c%c^",a+48,c+48,d+48);
            lcd_clear();

            // Tampilan pada LCD
            sprintf(data2,"%d%d%d",a,c,d);
            lcd_gotoxy(0,0);lcd_putsf("NO.TEMPAT DUDUK:");
        }
    }
}
```

```
lcd_gotoxy(7,1);lcd_puts(data2);
delay_ms(3000);
counter++;
y=1;

while(y==1)
{
    // Jika limit_tutup aktif & limit_buka tidak aktif
    if(PINC.6==1 && PINC.7==0 && u==0)
    {
        PORTE.0=1; // motor_buka aktif
        PORTE.1=0; // motor_tutup tidak aktif
    }

    //Jika limit_buka tidak aktif & limit_tutup tidak aktif
    else if(PINC.6==1 && PINC.7==1 && u==0)
    {
        PORTE.0=1; // motor_buka aktif
        PORTE.1=0; // motor_tutup tidak aktif
    }

    //Jika limit_buka aktif & limit_tutup tidak aktif
    else if(PINC.6==0 && PINC.7==1 && u==0)
    {
        PORTE.0=0; // motor_buka tidak aktif
        PORTE.1=0; // motor_tutup tidak aktif
        delay_ms(5000); // setelah 5 detik
        v=1;

        while(v==1)
        {
            PORTE.0=0; // motor_buka tidak aktif
            PORTE.1=1; // motor_tutup aktif

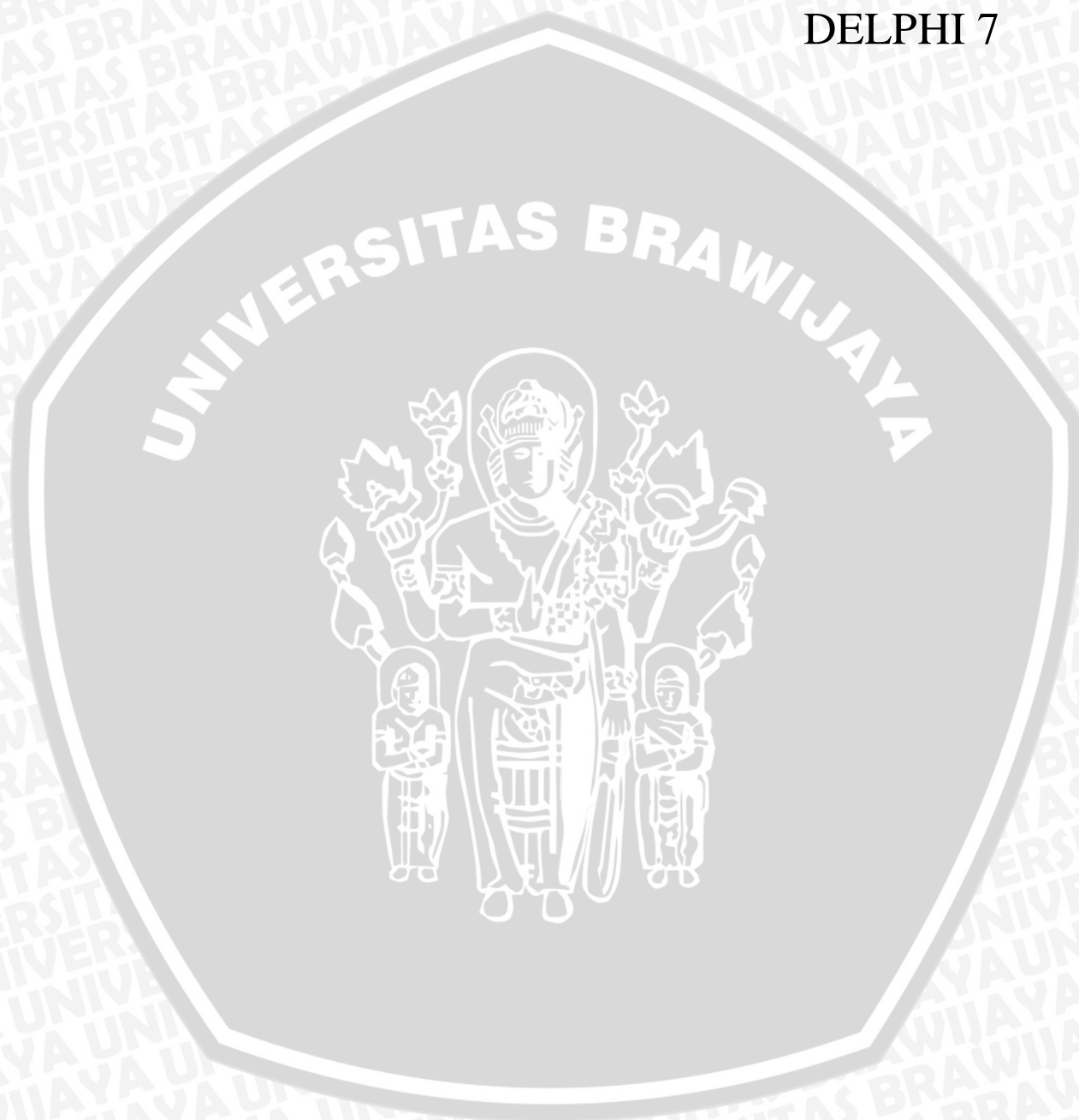
            //Jika limit_buka tidak aktif & limit_tutup aktif
            if(PINC.6==1 && PINC.7==0)
            {
                PORTE.0=0; // motor_buka tidak aktif
                PORTE.1=0; // motor_tutup tidak aktif
                v=0;
            }
        }
        u=1;
    }

    //Jika limit_buka tidak aktif & limit_tutup aktif
    else if(PINC.6==1 && PINC.7==0 && u==1)
    {
        PORTE.0=0; // motor_buka tidak aktif
        PORTE.1=0; // motor_tutup tidak aktif
        y=0;
    }
}
w=0;
}
// Jika data dari komputer = 0 (nomor serial tidak terdaftar)
else
{
    lcd_gotoxy(0,0);lcd_putsf(" KARTU ANDA ");
    lcd_gotoxy(0,1);lcd_putsf("TIDAK TERDAFTAR");
    delay_ms(3000);
    lcd_clear();
    w=0;
}
};
};
```


LAMPIRAN IV

LISTING PROGRAM

DELPHI 7



1. PROSES PEMESANAN TIKET

unit pemesanan_tiket;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Mask, Buttons, DB, ADOdb, jpeg, ExtCtrls;

type

TForm1 = class(TForm)

StaticText1: TStaticText;

StaticText2: TStaticText;

StaticText3: TStaticText;

StaticText4: TStaticText;

EditNomor: TMaskEdit;

EditNama: TMaskEdit;

EditAlamat: TMaskEdit;

EditTelpon: TMaskEdit;

BitBtnCari: TBitBtn;

BitBtnUbah: TBitBtn;

BitBtnPesan: TBitBtn;

BitBtnSimpan: TBitBtn;

SQLConnectionMySQL: TADOConnection;

SQLDataSet1: TADODataSet;

BitBtnLihat: TBitBtn;

BitBtnBaru: TBitBtn;

BitBtnOut: TBitBtn;

procedure BitBtnCariClick(Sender: TObject);

procedure AturField(Keadaan : Boolean);

procedure KeadaanAwal;

procedure BitBtnUbahClick(Sender: TObject);

procedure BitBtnSimpanClick(Sender: TObject);

procedure BitBtnBaruClick(Sender: TObject);

procedure BitBtnOutClick(Sender: TObject);

procedure BitBtnPesanClick(Sender: TObject);

procedure BitBtnLihatClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

```
implementation
```

```
uses psan_tiket, lihattiket;
```

```
{ $\$R$  *.dfm}
```

```
procedure TForm1.AturField(Keadaan : Boolean);
```

```
begin
```

```
EditNomor.Enabled := not Keadaan;
```

```
EditNama.Enabled := Keadaan;
```

```
EditAlamat.Enabled := Keadaan;
```

```
EditTelpon.Enabled := Keadaan;
```

```
end;
```

```
procedure TForm1.KeadaanAwal;
```

```
begin
```

```
BitBtnCari.Visible := True;
```

```
BitBtnUbah.Visible := False;
```

```
BitBtnPesan.Visible := False;
```

```
BitBtnSimpan.Visible := False;
```

```
BitBtnLihat.Visible := False;
```

```
BitBtnBaru.Enabled := False;
```

```
AturField(False);
```

```
EditNomor.Text := '';
```

```
EditNama.Text := '';
```

```
EditAlamat.Text := '';
```

```
EditTelpon.Text := '';
```

```
end;
```

```
procedure TForm1.BitBtnCariClick(Sender: TObject);
```

```
begin
```

```
BitBtnBaru.Enabled := True;
```

```
// kalau kosong hentikan
```

```
if EditNomor.Text = '' then
```

```
begin
```

```
EditNomor.SetFocus;
```

```
Exit;
```

```
end;
```

```
// Memproses pernyataan SQL
```

```
SQLDataSet1.Active := False;
```

```
SQLDataSet1.CommandText :=
```



```
' SELECT * FROM data_diri WHERE no_serial = '+' + EditNomor.Text + ''';

SQLDataSet1.Active := True;

if SQLDataSet1.RecordCount < 1 then
begin
MessageDlg('Data Tidak Ditemukan!',mtInformation,[mbOK],0);
EditNama.Text      := ' ';
EditAlamat.Text    := ' ';
EditTelpon.Text    := ' ';
EditNomor.SetFocus;
Exit;
end;

// berarti ketemu
with SQLDataSet1 do
begin
BitBtnUbah.Visible := True;
BitBtnPesan.Visible := True;
BitBtnSimpan.Visible := True;
BitBtnLihat.Visible := True;

EditNama.Text := FieldByName('nama').Value;
EditAlamat.Text := FieldByName ('alamat').Value;
EditTelpon.Text := FieldByName ('no_telp').Value;
end; {akhir with}

EditNomor.SetFocus;

end;

procedure TForm1.BitBtnUbahClick(Sender: TObject);
begin
BitBtnUbah.Enabled := False;
BitBtnSimpan.Visible := True;
AturField(True);
EditNama.Enabled := True;
EditAlamat.Enabled := True;
EditTelpon.Enabled := True;
EditNama.SetFocus();
end;

procedure TForm1.BitBtnSimpanClick(Sender: TObject);
const
PETIK : Char = ''';
var
```

```
PernyataanSQL: String;
begin
BitBtnUbah.Enabled := True;
PernyataanSQL := 'UPDATE data_diri SET ' +
' nama = ' + PETIK + EditNama.Text + PETIK + ',' +
' alamat = ' + PETIK + EditAlamat.Text + PETIK + ',' +
' no_telp = ' + PETIK + EditTelpon.Text + PETIK +
'WHERE no_serial = ' + PETIK + EditNomor.Text + PETIK;
SQLConnectionMySQL.Execute(PernyataanSQL);

MessageDlg('Data sudah disimpan',mtInformation,[mbOK],0);
BitBtnSimpan.Enabled := False;
BitBtnUbah.Enabled := True;
EditNama.Enabled := False;
EditAlamat.Enabled := False;
EditTelpon.Enabled := False;
end;

procedure TForm1.BitBtnBaruClick(Sender: TObject);
begin
KeadaanAwal;
EditNomor.SetFocus;
end;

procedure TForm1.BitBtnOutClick(Sender: TObject);
begin
Close;
end;

procedure TForm1.BitBtnPesanClick(Sender: TObject);
begin
Form2.Showmodal;
end;

procedure TForm1.BitBtnLihatClick(Sender: TObject);
begin
Form3.Show;
end;
end.

unit psan_tiket;

interface

uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
```

```
Dialogs, StdCtrls, Mask, DBCtrls, ExtCtrls, DB, ADODB, Buttons;
```

```
type
```

```
TForm2 = class(TForm)
```

```
  Panel1: TPanel;
```

```
  Panel2: TPanel;
```

```
  Panel3: TPanel;
```

```
  CheckBox1: TCheckBox;
```

```
  CheckBox2: TCheckBox;
```

```
  CheckBox3: TCheckBox;
```

```
  CheckBox4: TCheckBox;
```

```
  CheckBox5: TCheckBox;
```

```
  CheckBox6: TCheckBox;
```

```
  CheckBox7: TCheckBox;
```

```
  CheckBox8: TCheckBox;
```

```
  CheckBox9: TCheckBox;
```

```
  CheckBox10: TCheckBox;
```

```
  CheckBox11: TCheckBox;
```

```
  CheckBox12: TCheckBox;
```

```
  CheckBox13: TCheckBox;
```

```
  CheckBox14: TCheckBox;
```

```
  CheckBox15: TCheckBox;
```

```
  CheckBox16: TCheckBox;
```

```
  Panel4: TPanel;
```

```
  Edit1: TEdit;
```

```
  Edit2: TEdit;
```

```
  Edit3: TEdit;
```

```
  Edit4: TEdit;
```

```
  Edit5: TEdit;
```

```
  Edit6: TEdit;
```

```
  Edit7: TEdit;
```

```
  Edit8: TEdit;
```

```
  Edit9: TEdit;
```

```
  Edit10: TEdit;
```

```
  Edit11: TEdit;
```

```
  Edit12: TEdit;
```

```
  Edit13: TEdit;
```

```
  Edit14: TEdit;
```

```
  Edit15: TEdit;
```

```
  Edit16: TEdit;
```

```
  Panel5: TPanel;
```

```
  DataSource1: TDataSource;
```

```
  Label1: TLabel;
```

```
  ADOTable1: TADOTable;
```

```
  ADOConnection1: TADOConnection;
```

```
  MaskEdit1: TMaskEdit;
```



```
Button1: TButton;
ADOCCommand1: TADOCCommand;
BitBtn1: TBitBtn;

procedure FormShow(Sender: TObject);

procedure Button1Click(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure CheckBox2Click(Sender: TObject);
procedure CheckBox3Click(Sender: TObject);
procedure CheckBox4Click(Sender: TObject);
procedure CheckBox5Click(Sender: TObject);
procedure CheckBox6Click(Sender: TObject);
procedure CheckBox7Click(Sender: TObject);
procedure CheckBox8Click(Sender: TObject);
procedure CheckBox9Click(Sender: TObject);
procedure CheckBox10Click(Sender: TObject);
procedure CheckBox11Click(Sender: TObject);
procedure CheckBox12Click(Sender: TObject);
procedure CheckBox13Click(Sender: TObject);
procedure CheckBox14Click(Sender: TObject);
procedure CheckBox15Click(Sender: TObject);
procedure CheckBox16Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);

private
protected
// procedure ActionChange(Sender: TObject; CheckDefaults: Boolean);
// override;
{ Private declarations }
public
{ Public declarations }
end;

var
Form2: TForm2;

implementation

uses pemesanan_tiket;

{$R *.dfm}

procedure TForm2.FormShow(Sender: TObject);
begin
MaskEdit1.Text := Form1.EditNomor.Text;
MaskEdit1.Show;
```

```
end;

procedure TForm2.Button1Click(Sender: TObject);
begin
Close;
end;

procedure TForm2.CheckBox1Click(Sender: TObject);
const
petik : Char = '';
begin
if CheckBox1.Checked then
begin
ADOCommand1.CommandType := cmdText;
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan
(no_serial,id_pertandingan)#13+ 'VALUES (' + petik + Form1.EditNomor.Text + petik
+', ' + petik + '01' + petik + ')';
ADOCommand1.Execute;
end
else
begin
ADOCommand1.CommandType := cmdText;
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '01'+
petik;
ADOCommand1.Execute;
end;
end;

procedure TForm2.CheckBox2Click(Sender: TObject);
const
petik : Char = '';
begin
if CheckBox2.Checked then
begin
ADOCommand1.CommandType := cmdText;
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan
(no_serial,id_pertandingan)#13+ 'VALUES (' + petik + Form1.EditNomor.Text + petik
+', ' + petik + '02' + petik + ')';
ADOCommand1.Execute;
end
else
begin
ADOCommand1.CommandType := cmdText;
```



```
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '02'+  
petik;  
ADOCommand1.Execute;  
end;  
  
end;
```

```
procedure TForm2.CheckBox3Click(Sender: TObject);  
const  
petik : Char = ''';  
begin  
if CheckBox3.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', '+ petik + '03'+ petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '03'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox4Click(Sender: TObject);  
const  
petik : Char = ''';  
begin  
if CheckBox4.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', '+ petik + '04'+ petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;
```

```
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '04'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox5Click(Sender: TObject);
```

```
const  
petik : Char = ''';  
begin  
if CheckBox5.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '05' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '05'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox6Click(Sender: TObject);
```

```
const  
petik : Char = ''';  
begin  
if CheckBox6.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '06' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;
```

```
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '06'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox7Click(Sender: TObject);
```

```
const  
petik : Char = ''';  
begin  
if CheckBox7.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '07' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '07'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox8Click(Sender: TObject);
```

```
const  
petik : Char = ''';  
begin  
if CheckBox8.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '08' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;
```

```
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '08'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox9Click(Sender: TObject);  
const  
petik : Char = ''';  
begin  
if CheckBox9.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '09' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '09'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox10Click(Sender: TObject);  
const  
petik : Char = ''';  
begin  
if CheckBox10.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '10' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;
```

```
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '10'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox11Click(Sender: TObject);  
const  
petik : Char = ''';  
begin  
if CheckBox11.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '11' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '11'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox12Click(Sender: TObject);  
const  
petik : Char = ''';  
begin  
if CheckBox12.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '12' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;
```

```
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '12'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox13Click(Sender: TObject);  
const  
petik : Char = ''';  
begin  
if CheckBox13.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '13' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '13'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox14Click(Sender: TObject);  
const  
petik : Char = ''';  
begin  
if CheckBox14.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '14' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;
```

```
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '14'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox15Click(Sender: TObject);  
const  
petik : Char = ''';  
begin  
if CheckBox15.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '15' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '15'+  
petik;  
ADOCommand1.Execute;  
end;  
end;
```

```
procedure TForm2.CheckBox16Click(Sender: TObject);  
const  
petik : Char = ''';  
begin  
if CheckBox16.Checked then  
begin  
ADOCommand1.CommandType := cmdText;  
ADOCommand1.CommandText := 'INSERT INTO jadwal_pertandingan  
(no_serial,id_pertandingan)'#13+ 'VALUES ('+ petik + Form1.EditNomor.Text + petik  
+', ' + petik + '16' + petik + ')';  
ADOCommand1.Execute;  
end  
else  
begin  
ADOCommand1.CommandType := cmdText;
```

```
ADOCommand1.CommandText := 'DELETE FROM jadwal_pertandingan WHERE no_serial = ' +  
petik + Form1.EditNomor.Text + petik + 'AND id_pertandingan = ' + petik + '16'+  
petik;  
ADOCommand1.Execute;  
end;  
end;  
  
procedure TForm2.BitBtn1Click(Sender: TObject);  
begin  
MessageDlg('Data sudah disimpan',mtInformation,[mbOK],0);  
end;  
  
end.  
  
unit lihattiket;  
  
interface  
  
uses  
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Mask, Grids, DBGrids, DB, ADODB, Buttons;  
  
type  
TForm3 = class(TForm)  
ADOConnection1: TADOConnection;  
ADOQuery1: TADOQuery;  
DataSource1: TDataSource;  
DBGrid1: TDBGrid;  
StaticText1: TStaticText;  
MaskEdit1: TMaskEdit;  
BitBtn1: TBitBtn;  
procedure FormShow(Sender: TObject);  
procedure FormCreate(Sender: TObject);  
procedure FormActivate(Sender: TObject);  
procedure BitBtn1Click(Sender: TObject);  
private  
{ Private declarations }  
public  
{ Public declarations }  
end;  
  
var  
Form3: TForm3;  
  
implementation
```



```
uses pemesanan_tiket;
```

```
{ $R *.dfm }
```

```
procedure TForm3.FormShow(Sender: TObject);
```

```
begin
```

```
Form3.MaskEdit1.Text := Form1.EditNomor.Text;
```

```
end;
```

```
procedure TForm3.FormActivate(Sender: TObject);
```

```
const
```

```
petik : Char = '';
```

```
begin
```

```
// untuk menampilkan data berupa jadwal pertandingan yang telah dipesan
```

```
ADOQuery1.Active := False;
```

```
ADOQuery1.SQL.Text := 'SELECT kode_pertandingan.tim_lawan,  
kode_pertandingan.tanggal FROM jadwal_pertandingan, kode_pertandingan WHERE  
jadwal_pertandingan.no_serial = '+' + petik + MaskEdit1.Text + petik + 'AND  
jadwal_pertandingan.id_pertandingan = kode_pertandingan.id_pertandingan ORDER BY  
tanggal';
```

```
ADOQuery1.Active := True;
```

```
DBGrid1.SelectedIndex;
```

```
end;
```

```
procedure TForm3.BitBtn1Click(Sender: TObject);
```

```
begin
```

```
Close;
```

```
end;
```

```
end.
```

2. HARI PERTANDINGAN

```
unit pertandinganhariini;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ExtCtrls, Grids, DBGrids, StdCtrls, Mask, DB, ADODB, Buttons,  
DBClient, Provider, RpCon, RpConDS, RpDefine, RpRave, ComCtrls, DBCtrls,  
XComDrv;
```

```
type
```

```
TForm1 = class(TForm)
```



```
ADOConnection1: TADOConnection;
ADOQuery1: TADOQuery;
DataSource1: TDataSource;
DBGrid1: TDBGrid;
Timer1: TTimer;
MaskEdit3: TMaskEdit;
BitBtn1: TBitBtn;
Panel1: TPanel;
Panel2: TPanel;
Panel3: TPanel;
Panel4: TPanel;
Label1: TLabel;
Label2: TLabel;
BitBtn3: TBitBtn;
ADOQuery2: TADOQuery;
DataSource2: TDataSource;
MaskEdit1: TMaskEdit;
MaskEdit2: TMaskEdit;
DBGrid2: TDBGrid;
BitBtn4: TBitBtn;
RvProject1: TRvProject;
RvDataSetConnection1: TRvDataSetConnection;
BitBtn5: TBitBtn;
ADOCommand1: TADOCommand;
MaskEdit4: TMaskEdit;
MaskEdit5: TMaskEdit;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
Edit8: TEdit;
Edit9: TEdit;
Edit10: TEdit;
Edit11: TEdit;
Edit12: TEdit;
Edit13: TEdit;
Edit14: TEdit;
Edit15: TEdit;
Edit16: TEdit;
Edit17: TEdit;
Edit18: TEdit;
Edit19: TEdit;
Edit20: TEdit;
```



```
Edit21: TEdit;
Edit22: TEdit;
Edit23: TEdit;
Edit24: TEdit;
Edit25: TEdit;
Edit26: TEdit;
Edit27: TEdit;
Edit28: TEdit;
Edit29: TEdit;
Memol: TMemo;
Button1: TButton;
XComm1: TXComm;
Edit30: TEdit;
RvSystem1: TRvSystem;
procedure Timer1Timer(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure TampilanAwal();
procedure BitBtn4Click(Sender: TObject);
procedure XComm1Data(Sender: TObject; const Received: Cardinal);
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  Buffer: widestring;
  data,ID,no_seat: string;
  i:integer;
  connect: boolean;

implementation

{$R *.dfm}

procedure TForm1.TampilanAwal;
begin
  ADOQuery1.Active := False;
  ADOQuery2.Active := False;
  MaskEdit3.Text := '';
  MaskEdit1.Text := '';
  MaskEdit2.Text := '';
```

```
MaskEdit3.SetFocus;
end;

//Membandingkan tanggal yang dimasukkan dengan tanggal sekarang
procedure TForm1.Timer1Timer(Sender: TObject);
var
date : String;
Ndate : TDateTime;
begin
Ndate:=Now;
DateTimeToString(date,'yyyy-mm-dd', Ndate);
Labell.Caption:= date;
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
MaskEdit3.SetFocus;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
const
petik : Char = '';
var
sA : String;
begin
if MaskEdit3.Text = Labell.Caption then
begin
// untuk mengambil data dari database dan menampilkan data berupa nomor serial
yang telah terdaftar pd pertandingan tsb
ADOQuery1.Active := False;
ADOQuery1.SQL.Text := 'SELECT jadwal_pertandingan.no_serial,
kode_pertandingan.tim_lawan FROM jadwal_pertandingan, kode_pertandingan WHERE
kode_pertandingan.tanggal = '+ petik + MaskEdit3.Text + petik + 'AND
jadwal_pertandingan.id_pertandingan = kode_pertandingan.id_pertandingan';
ADOQuery1.Active := True;

DBGrid1.SelectedIndex;

//menampikan tim lawan yang bertanding pada papan versus
with ADOQuery1 do
begin
sA := ADOQuery1.fieldbyname('tim_lawan').AsString;
end;

Panel4.Caption:= sA;
end
```

```
else
begin
//jika tanggal bukan tanggal pada hari ini
MessageDlg('Cek kembali masukan tanggal!',mtInformation,[mbOK],0);
end;
end;

procedure TForm1.BitBtn3Click(Sender: TObject);
begin
Close;
end;

procedure delay(lama:longint);
var
ref:longint;
begin
ref:=gettickcount;
repeat
application.ProcessMessages;
until((gettickcount-ref)>=lama)
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
if(connect=false)then
begin
XComm1.OpenDevice;
connect:=true;
Button1.Caption:='Disconnect';
i:=0;
repeat
delay(1000);
inc(i);
until((i=100)or(connect=false));
end
else
begin
XComm1.CloseDevice;
connect:=false;
Button1.Caption:='Connect';
end;
end;

procedure TForm1.XComm1Data(Sender: TObject; const Received: Cardinal);
const
```

```
petik : Char = ''';
var
sB, sC, sD, sE, sF, sG, sH, sI, sJ, sK, sL, sM, sN, sO, sP, sQ, sR, sS, sT, sU,
sV, sW, sX, sY, sZ, s0, s1, s2, s3, s4, s5, s6 : String;
begin
// Mengambil data dari serial
XComm1.ReadString(data);
Buffer:= Buffer + data;
//Memo1.Lines.Append(Buffer);
if (data='') then begin
ID:= copy(Buffer,pos('#',Buffer)+1,5);
MaskEdit1.Text:=string(ID);

// Mencari nomor serial pada database sudah terdaftar atau belum
if ADOQuery1.Locate('no_serial',MaskEdit1.Text,[]) = True then
// Mengirim data = 1 ke mikrokontroler
XComm1.SendString('#100000*') and
//menampilkan data berupa jadwal pertandingan yang telah dipesan
ADOQuery2.SQL.Text := 'SELECT kode_pertandingan.tim_lawan,
kode_pertandingan.tanggal' + #13 +
'FROM kode_pertandingan, jadwal_pertandingan' + #13 +
'WHERE jadwal_pertandingan.no_serial =' + petik +
MaskEdit1.Text + petik + #13 +
'AND jadwal_pertandingan.id_pertandingan =
kode_pertandingan.id_pertandingan';
ADOQuery2.Active:= True;

if (data='^') then begin
no_seat:= Copy(Buffer,pos('*',Buffer)+1,3);
MaskEdit2.Text:= string(no_seat);

ADOQuery2.First;
ADOQuery2.Next;

//mengambil data dari database berupa pertandingan lainnya yang telah dipesan
with ADOQuery2 do
begin
sB := ADOQuery2.fieldbyname('tim_lawan').AsString;
sC := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
sD := ADOQuery2.fieldbyname('tim_lawan').AsString;
sE := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
sF := ADOQuery2.fieldbyname('tim_lawan').AsString;
sG := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
```

```
sH := ADOQuery2.fieldbyname('tim_lawan').AsString;
sI := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
sJ := ADOQuery2.fieldbyname('tim_lawan').AsString;
sK := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
sL := ADOQuery2.fieldbyname('tim_lawan').AsString;
sM := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
sN := ADOQuery2.fieldbyname('tim_lawan').AsString;
sO := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
sP := ADOQuery2.fieldbyname('tim_lawan').AsString;
sQ := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
sR := ADOQuery2.fieldbyname('tim_lawan').AsString;
sS := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
sT := ADOQuery2.fieldbyname('tim_lawan').AsString;
sU := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
sV := ADOQuery2.fieldbyname('tim_lawan').AsString;
sW := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
sX := ADOQuery2.fieldbyname('tim_lawan').AsString;
sY := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
sZ := ADOQuery2.fieldbyname('tim_lawan').AsString;
s0 := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
s1 := ADOQuery2.fieldbyname('tim_lawan').AsString;
s2 := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
s3 := ADOQuery2.fieldbyname('tim_lawan').AsString;
s4 := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
end;
```

```
ADOQuery2.First;
while not ADOQuery2.Eof do
begin
s5 := ADOQuery2.fieldbyname('tim_lawan').AsString;
s6 := ADOQuery2.fieldbyname('tanggal').AsString;
ADOQuery2.Next;
end;
```

```
//Menampilkan data pertandingan yang telah dipesan
MaskEdit4.Text := sB;MaskEdit5.Text := sC;Edit1.Text := sD;Edit2.Text := sE;
Edit3.Text := sF;Edit4.Text := sG;Edit5.Text := sH;Edit6.Text := sI;
Edit7.Text := sJ;Edit8.Text := sK;Edit9.Text := sL;Edit10.Text := sM;
Edit11.Text := sN;Edit12.Text := sO;Edit13.Text := sP;Edit14.Text := sQ;
Edit15.Text := sR;Edit16.Text := sS;Edit17.Text := sT;Edit18.Text := sU;
Edit19.Text := sV;Edit20.Text := sW;Edit21.Text := sX;Edit22.Text := sY;
Edit23.Text := sZ;Edit24.Text := s0;Edit25.Text := s1;Edit26.Text := s2;
Edit27.Text := s3;Edit28.Text := s4;Edit29.Text := s5;
```

```
//menghapus data jika ada data yang sama
```

```
if Edit29.Text = MaskEdit4.Text then
```

```
begin
```

```
Edit1.Text := '';Edit2.Text := '';Edit3.Text := '';Edit4.Text := '';
```

```
Edit5.Text := '';Edit6.Text := '';Edit7.Text := '';Edit8.Text := '';
```

```
Edit9.Text := '';Edit10.Text := '';Edit11.Text := '';Edit12.Text := '';
```

```
Edit13.Text := '';Edit14.Text := '';Edit15.Text := '';Edit16.Text := '';
```

```
Edit17.Text := '';Edit18.Text := '';Edit19.Text := '';Edit20.Text := '';
```

```
Edit21.Text := '';Edit22.Text := '';Edit23.Text := '';Edit24.Text := '';
```

```
Edit25.Text := '';Edit26.Text := '';Edit27.Text := '';Edit28.Text := '';
```

```
Edit29.Text := '';
```

```
end
```

```
else
```

```
begin
```

```
if Edit29.Text = Edit1.Text then
```

```
begin
```

```
Edit3.Text := '';Edit4.Text := '';Edit5.Text := '';Edit6.Text := '';
```

```
Edit7.Text := '';Edit8.Text := '';Edit9.Text := '';Edit10.Text := '';
```

```
Edit11.Text := '';Edit12.Text := '';Edit13.Text := '';Edit14.Text := '';
```

```
Edit15.Text := '';Edit16.Text := '';Edit17.Text := '';Edit18.Text := '';
```

```
Edit19.Text := '';Edit20.Text := '';Edit21.Text := '';Edit22.Text := '';
```

```
Edit23.Text := '';Edit24.Text := '';Edit25.Text := '';Edit26.Text := '';
```

```
Edit27.Text := '';Edit28.Text := '';Edit29.Text := '';
```

```
end
```

```
else
```

```
begin
```

```
if Edit29.Text = Edit3.Text then
```

```
begin
```

```
Edit5.Text := '';Edit6.Text := '';Edit7.Text := '';Edit8.Text := '';
```

```
Edit9.Text := '';Edit10.Text := '';Edit11.Text := '';Edit12.Text := '';
```

```
Edit13.Text := '';Edit14.Text := '';Edit15.Text := '';Edit16.Text := '';
```

```
Edit17.Text := '';Edit18.Text := '';Edit19.Text := '';Edit20.Text := '';
```

```
Edit21.Text := '';Edit22.Text := '';Edit23.Text := '';Edit24.Text := '';
```

```
Edit25.Text := '';Edit26.Text := '';Edit27.Text := '';Edit28.Text := '';
```

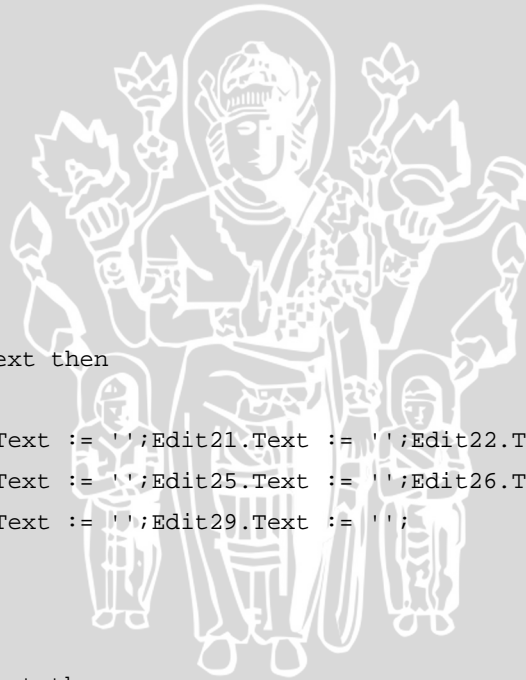
```
Edit29.Text := '';
```

```
end
```



```
else
begin
if Edit29.Text = Edit5.Text then
begin
Edit7.Text := '';Edit8.Text := '';Edit9.Text := '';Edit10.Text := '';
Edit11.Text := '';Edit12.Text := '';Edit13.Text := '';Edit14.Text := '';
Edit15.Text := '';Edit16.Text := '';Edit17.Text := '';Edit18.Text := '';
Edit19.Text := '';Edit20.Text := '';Edit21.Text := '';Edit22.Text := '';
Edit23.Text := '';Edit24.Text := '';Edit25.Text := '';Edit26.Text := '';
Edit27.Text := '';Edit28.Text := '';Edit29.Text := '';
end
else
begin
if Edit29.Text = Edit7.Text then
begin
Edit9.Text := '';Edit10.Text := '';Edit11.Text := '';Edit12.Text := '';
Edit13.Text := '';Edit14.Text := '';Edit15.Text := '';Edit16.Text := '';
Edit17.Text := '';Edit18.Text := '';Edit19.Text := '';Edit20.Text := '';
Edit21.Text := '';Edit22.Text := '';Edit23.Text := '';Edit24.Text := '';
Edit25.Text := '';Edit26.Text := '';Edit27.Text := '';Edit28.Text := '';
Edit29.Text := '';
end
else
begin
if Edit29.Text = Edit9.Text then
begin
Edit11.Text := '';Edit12.Text := '';Edit13.Text := '';Edit14.Text := '';
Edit15.Text := '';Edit16.Text := '';Edit17.Text := '';Edit18.Text := '';
Edit19.Text := '';Edit20.Text := '';Edit21.Text := '';Edit22.Text := '';
Edit23.Text := '';Edit24.Text := '';Edit25.Text := '';Edit26.Text := '';
Edit27.Text := '';Edit28.Text := '';Edit29.Text := '';
end
else
begin
if Edit29.Text = Edit11.Text then
begin
Edit13.Text := '';Edit14.Text := '';Edit15.Text := '';Edit16.Text := '';
Edit17.Text := '';Edit18.Text := '';Edit19.Text := '';Edit20.Text := '';
Edit21.Text := '';Edit22.Text := '';Edit23.Text := '';Edit24.Text := '';
Edit25.Text := '';Edit26.Text := '';Edit27.Text := '';Edit28.Text := '';
Edit29.Text := '';
end
else
begin
if Edit29.Text = Edit13.Text then
begin
```

```
Edit15.Text := '';Edit16.Text := '';Edit17.Text := '';Edit18.Text := '';
Edit19.Text := '';Edit20.Text := '';Edit21.Text := '';Edit22.Text := '';
Edit23.Text := '';Edit24.Text := '';Edit25.Text := '';Edit26.Text := '';
Edit27.Text := '';Edit28.Text := '';Edit29.Text := '';
end
else
begin
if Edit29.Text = Edit15.Text then
begin
Edit17.Text := '';
Edit18.Text := '';
Edit19.Text := '';
Edit20.Text := '';
Edit21.Text := '';
Edit22.Text := '';
Edit23.Text := '';
Edit24.Text := '';
Edit25.Text := '';
Edit26.Text := '';
Edit27.Text := '';
Edit28.Text := '';
Edit29.Text := '';
end
else
begin
if Edit29.Text = Edit17.Text then
begin
Edit19.Text := '';Edit20.Text := '';Edit21.Text := '';Edit22.Text := '';
Edit23.Text := '';Edit24.Text := '';Edit25.Text := '';Edit26.Text := '';
Edit27.Text := '';Edit28.Text := '';Edit29.Text := '';
end
else
begin
if Edit29.Text = Edit19.Text then
begin
Edit21.Text := '';Edit22.Text := '';Edit23.Text := '';Edit24.Text := '';
Edit25.Text := '';Edit26.Text := '';Edit27.Text := '';Edit28.Text := '';
Edit29.Text := '';
end
else
begin
if Edit29.Text = Edit21.Text then
begin
Edit23.Text := '';Edit24.Text := '';Edit25.Text := '';Edit26.Text := '';
Edit27.Text := '';Edit28.Text := '';Edit29.Text := '';
end
end
end
end
end
```



```
else
begin
if Edit29.Text = Edit23.Text then
begin
Edit25.Text := '';Edit26.Text := '';Edit27.Text := '';Edit28.Text := '';
Edit29.Text := '';
end
else
begin
if Edit29.Text = Edit25.Text then
begin
Edit27.Text := '';Edit28.Text := '';Edit29.Text := '';
end
else
begin
if Edit29.Text = Edit27.Text then
begin
Edit29.Text := '';
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;
end;

//perintah untuk mencetak data
RvProject1.Open;
RvProject1.SelectReport('ReportPrinted', False);
RvProject1.SetParam('no_serial',MaskEdit1.Text);
RvProject1.SetParam('no_seat',MaskEdit2.Text);
RvProject1.SetParam('1',MaskEdit4.Text);
RvProject1.SetParam('2',MaskEdit5.Text);
RvProject1.SetParam('3',Edit1.Text);
RvProject1.SetParam('4',Edit2.Text);
RvProject1.SetParam('5',Edit3.Text);
RvProject1.SetParam('6',Edit4.Text);
RvProject1.SetParam('7',Edit5.Text);
```

UNIVERSITAS BRAWIJAYA



```

RvProject1.SetParam('8',Edit6.Text);
RvProject1.SetParam('9',Edit7.Text);
RvProject1.SetParam('10',Edit8.Text);
RvProject1.SetParam('11',Edit9.Text);
RvProject1.SetParam('12',Edit10.Text);
RvProject1.SetParam('13',Edit11.Text);
RvProject1.SetParam('14',Edit12.Text);
RvProject1.SetParam('15',Edit13.Text);
RvProject1.SetParam('16',Edit14.Text);
RvProject1.SetParam('17',Edit15.Text);
RvProject1.SetParam('18',Edit16.Text);
RvProject1.SetParam('19',Edit17.Text);
RvProject1.SetParam('20',Edit18.Text);
RvProject1.SetParam('21',Edit19.Text);
RvProject1.SetParam('22',Edit20.Text);
RvProject1.SetParam('23',Edit21.Text);
RvProject1.SetParam('24',Edit22.Text);
RvProject1.SetParam('25',Edit23.Text);
RvProject1.SetParam('26',Edit24.Text);
RvProject1.SetParam('27',Edit25.Text);
RvProject1.SetParam('28',Edit26.Text);
RvProject1.SetParam('29',Edit27.Text);
RvProject1.SetParam('30',Edit28.Text);
RvProject1.Execute;
RvProject1.Close
end;

Buffer:='';

end;
end;
end.

```



LAMPIRAN V

DATASHEET

