

LEMBAR PERSETUJUAN



APLIKASI RFID PADA SISTEM PARKIR PRABAYAR TUGAS AKHIR

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Diploma III Teknik Elektronika*

Disusun Oleh :

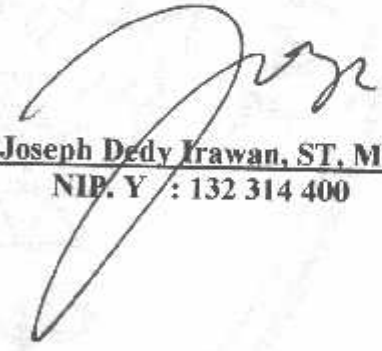
BAYU BAGUS TRIYO WIBOWO
NIM : 05.52.206

Diperiksa dan Disetujui

Ketua Jurusan Teknik Elektro DIII

Dosen Pembimbing


Ir. H. Taufik Hidayat, MT
NIP. Y 101 8700 151


Joseph Dedy Irawan, ST, MT
NIP. Y : 132 314 400

**KONSENTRASI TEKNIK ELEKTRONIKA DIII
JURUSAN TEKNIK ELEKTRO DIII
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2009**

Label (pada palet standart)	Name	Label8
	Caption	Plat Nomor
	FocusControl	DBEdit8
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit1
	Data field	No kartu
	Data Source	Data Source1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit2
	Data field	Nama
	Data Source	Data Source1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit3
	Data field	Alamat
	Data Source	Data Source1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit4
	Data field	Status
	Data Source	Data Source1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit5
	Data field	No telpn
	Data Source	Data Source1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit6
	Data field	Plat nomor
	Data Source	Data Source1
DBNavigator (pada palet data control)	Name	DBNavigator1
	DatSource	DataSource1
DBImage1 (pada palet Data Control)	Name	DBImage1
	Data Source	Data Source1
	Data Field	FOTO
DBGrid (pada palet Data Control)	Name	DBGrid1
	DataSource	Data Source1

BAB IV

PENGUJIAN RANGKAIN DAN ANALISA DATA

Untuk mendapatkan hasil yang maksimal setelah melaksanakan perancangan dan pembuatan perangkat, maka perlu dilakukan suatu pengujian terhadap perangkat yang telah dibuat. Pengujian ini bertujuan untuk mengetahui apakah perangkat yang telah dibuat dapat bekerja sesuai dengan perencanaan.

4.1 Pengujian Penerimaan Data dari database

4.1.1. Tujuan

Untuk mengetahui apakah data yang diterima oleh Delphi sesuai dengan apa yang ada pada database.

4.1.2. Peralatan yang digunakan

1. PC
2. Rangkaian RFID
3. Kabel Serial RS-232
4. Adaptor 9 – 12 Volt

4.1.3. Langkah Pengujian

1. Membuka program MS Access, Kemudian buat Database.
2. Membuka Program Delphi dan menjalankan program aplikasi.

3. Menghubungkan RFID ke PC menggunakan kabel serial RS-232.
4. Program dijalankan

4.1.4. Hasil Pengujian

Hasil pengujian Penerimaan Data dengan RFID dari suatu PC sebagai berikut :

- Setelah membuat Database kemudian pada program delphi dijalankan maka akan muncul suatu program absensi yang sudah dibuat,yaitu seperti gambar dibawah ini.

Utama
Menu

PARKIR PRABAYAR ITN		
NO KARTU	000-4617678	30/08/2009
NAMA	ADJIE	
STATUS	MAHASISWA	
TANGGAL	30/08/2009	
NO KENDARAAN	N 2121 JE	SALDO 1500
		KREDIT 500
SALDO ANDA : Rp 1500,00		<input checked="" type="checkbox"/>

Gambar 4.1. Menu Utama Program

Prinsip kerja dari program ini adalah sebagai berikut, karena sistem parkir yang dirancang merupakan sistem parkir yang bersifat prabayar, maka setiap pemilik kendaraan harus memiliki kartu parkir. Pada saat kendaraan akan memasuki area parkir, maka pengemudi mendekatkan kartu parkir yang berisikan *tag* RFID pada RFID *reader* yang terdapat pada pintu masuk parkir. Setelah reader berhasil membaca *tag* RFID pada kartu parkir, maka hasil pembacaan tersebut akan dikirim ke *server* melalui RS232. Pada *server*, hasil pembacaan *reader* tersebut akan diidentifikasi dengan cara membandingkan *tag* yang berada dalam *database server* dengan *tag* RFID dari pengiriman RS232. Bila dalam *database server* terdapat *tag* RFID yang sesuai dengan *tag* dari pembacaan RFID *reader* (pada pintu masuk tersebut), maka server akan mengirimkan identitas pemilik kartu parkir tersebut ke pintu masuk tadi melalui RS232 yang akan di tampilkan pada PC.

➤ Tampilan Menu Pada Program Utama

Utama

Menu

PARKIR PRABAYAR ITN		
NO KARTU 0004617678	30/08/2009	
NAMA ADJIE		
STATUS MAHASISWA		
TANGGAL 30/08/2009		
NO KENDARAAN	SALDO 1500	<input checked="" type="checkbox"/>
N 2121 JE	KREDIT 500	
SALDO ANDA : Rp 1500,00		

Tampilan Design Menu Utama diperlukan komponen *ADO Connection*, *ADO Table*, *Data Source*, *Label 11* buah, *DBEdit 8* buah, satu buah *Dbimage1* satu buah *Edit1*, *DBQuery1*, *ADOCCommand1*, *Va Comm1* dan *DBnafigator1*.

Pada Design Menu Utama listing programnya sebagai berikut :

```
procedure TUtama.TampilParkir;
```

```
Begin
```

```
    ADOQuery2.Active := False;
```

```
    ADOQuery2.SQL.Clear;
```

```
    ADOQuery2.SQL.Add('SELECT * FROM Parkir ' + 'WHERE Tanggal = :tanggal  
And ID = :ID');
```

```
    ADOQuery2.Parameters.ParamByName('tanggal').Value := Date;
```

```
    ADOQuery2.Parameters.ParamByName('ID').Value := Edit1.Text;
```

```
    ADOQuery2.Active := True;
```

```
End;
```

```
procedure TUtama.SimpanParkir;
```

```
begin
```

```
    TampilParkir;
```

```
    IF ADOQuery2.RecordCount = 0
```

```
    Then
```

```
    Begin
```

```
        with ADOCommand1 Do
```

```
        Begin
```

**BERITA ACARA UJIAN TUGAS AKHIR
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : BAYU BAGUS TRIYO WIBOWO
NIM : 05.52.206
JURUSAN : TEKNIK ELEKTRO D-III
JUDUL SKRIPSI : APLIKASI RFID PADA SISTEM PARKIR

PRABAYAR

*Dipertahankan di hadapan Tim Penguji Tugas Akhir
Jenjang Program Diploma Tiga (DIII), pada:*

Hari / Tanggal : Jum'at / 02 Oktober 2009

Dengan Nilai : 80,70 (A) ¹⁰

Panitia Ujian Tugas Akhir

Ketua

Sekretaris



Ir. Sidik Noertjahjono, MT
NIP.Y 102 8700 163



Ir. H. Taufik Hidayat, MT
NIP.Y 101 8700 151

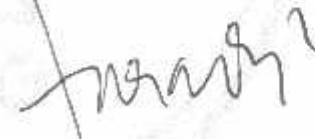
Anggota Penguji

Penguji I

Penguji II



Ir. H. Choirul Saleh, MT
NIP.Y 101 8800 190



Irmalia Suryani Faradisa ST, MT
NIP.Y 103 0100 365

ABSTRAKSI

APLIKASI RFID PADA SISTEM PARKIR PRABAYAR

Disusun Oleh : Bayu Bagus Triyo Wibowo , 0552206

Dosen Pembimbing : Joseph Dedy Irawan, ST, MT.

Kata Kunci : RFID

Salah satu tuntutan yang harus dihadapi oleh dunia industri jasa saat ini adalah menciptakan sistem pelayanan terhadap publik yang memberikan kemudahan dalam tukar menukar informasi, transaksi, dengan tingkat keamanan yang tinggi. Masalah ini juga dapat dijumpai pada lingkungan kerja apa saja yang membutuhkan faktor-faktor seperti : mobilitas tinggi, keamanan data, ketangguhan terhadap gangguan, kestabilan dan unjuk kerja yang tinggi. Perparkiran adalah salah satu dari jenis industri jasa yang menghadapi tantangan yang disebutkan di atas.

Salah satu teknologi yang menawarkan solusi adalah RFID (Radio Frequency Identification) adalah teknologi compact wireless yang diunggulkan untuk mentransformasi dunia komersial. Sebagai suksesor dari barcode, RFID dapat melakukan kontrol otomatis untuk banyak hal. Sistem-sistem RFID menawarkan peningkatan efisiensi, keamanan, dan juga kenyamanan si pengguna jasa.

Sistem parkir yang berlaku saat ini masih bersifat manual dengan menggunakan karcis parkir sebagai bukti parkir kendaraan dan pembayaran biaya

parkir kendaraan dilakukan secara tunai. Sistem parkir yang demikian memiliki kelemahan antara lain, kurangnya tingkat keamanan dan dapat menimbulkan praktek korupsi pada petugas parkir. Dari permasalahan tersebut, maka timbul ide untuk membuat sistem parkir prabayar yang dapat mengidentifikasi identitas pemilik kendaraan berupa no.ID kartu, nama, alamat, nomor pelat kendaraan dan jumlah uang.

KATA PENGANTAR



Alhamdulillahirobbil 'alamin, segala puji dan syukur hanyalah pantas tercurah kehadirat **الله** , yang senantiasa memberikan kekuatan serta cinta kasih kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir dengan judul **"APLIKASI RFID PADA SISTEM PARKIR PRABAYAR "** sebagai syarat menempuh tugas akhir pada program pendidikan Diploma Teknik Elektro Institut Teknologi Nasional Malang.

Salah satu teknologi RFID (Radio Frequency Identification) adalah teknologi compact wireless yang diunggulkan untuk mentransformasi dunia komersial. Sebagai suksesor dari barcode, RFID dapat melakukan kontrol otomatis untuk banyak hal, sistem-sistem RFID menawarkan peningkatan efisiensi, keamanan, dan juga kenyamanan si pengguna jasa. diharapkan dapat meningkatkan kreatifitas berfikir mahasiswa dalam mengembangkan ilmu pengetahuan khususnya dalam bidang Elektronika.

Penulis sangat menyadari tentunya banyak terdapat kekurangan dalam tugas akhir ini, karena itu berbagai masukan sangat diharapkan untuk perbaikan tugas akhir ini di masa yang akan datang. Semoga tugas akhir ini dapat memberikan manfaat bagi berbagai pihak.

Malang, September 2009

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iv
DAFTAR ISI	v
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix

BAB I PENDAHULUAN

1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Perumusan Masalah	2
1.4. Ruang Lingkup Pembahasan	4
1.5. Metodologi Penelitian.....	4
1.6. Sistematika Penulisan.....	5

BAB II TEORI PENUNJANG

2.1. Radio Frekuensi Identification (RFID)	7
2.1.1 RFID <i>tag</i>	7
2.1.2 RFID <i>Reader</i>	8
2.2. Borland Delphi 7.0	12
2.2.1. IDE Delphi	13
2.2.2. Menu Borland Delphi	16

2.3.3. Komponen Visual dan Non Visual	17
2.3.4. Informasi Sintak Umum	21
2.3.4.1. Komentar	21
2.3.4.2. Type Data	22
2.4. Dasar Basis Data	34
2.4.1. Konsep Basis Data	34
2.4.2. Definisi	34
2.4.3. Operasi Dasar Basis Data	36
2.5. Microsoft Access	37
2.5.1. Keunggulan Microsoft Access	38
2.5.2. Database Di Dalam MS Access	39
2.6. RS-232	41

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1. Pendahuluan	44
3.1.1. Blok Diagram Keseluruhan	44
3.2. Perancangan Alat	45
3.3. Komunikasi Data Antara RFID Tag Dengan Komputer	47
3.4. Flowchart	49
3.5. Perancangan Perangkat Lunak (<i>Software</i>)	51
3.5.1. Program Aplikasi Komputer	51
3.5.2. Perancangan Pembuatan Software Menggunakan Bahasa Pemrograman Borland Delphi7 dan MS Access	52

3.5.2.1. Penyusunan Desain Software	52
---	----

BAB IV PENGUJIAN RANGKAIAN DAN ANALISA DATA

4.1. Pengujian Penerimaan Data dari Database	61
4.1.1. Tujuan	61
4.1.2. Peralatan yang digunakan	61
4.1.3. Langkah Pengujian	61
4.1.4. Hasil Pengujian	62

BAB V PENUTUP

5.1. Kesimpulan	71
5.2. Saran	72

DAFTAR PUSTAKA

DAFTAR TABEL

Tabel 2.1 Karakteristik Fisik	II-10
Tabel 2.2 Pin Description & Output Data Format	II-11
Tabel 2.3 Macam-macam Komponen Standar Delphi	II-18
Tabel 2.4 Tabel Tipe Data Integer	II-23
Tabel 2.5 Tabel Tipe Data Real	II-23
Tabel 2.6 Tabel Tipe Data Boolean	II-25
Tabel 2.7 Tabel Tipe Data Karakter.....	II-25
Tabel 2.8 Tabel Tipe Data String.....	II-26
Tabel 2.9 Tabel Operator Aritmatika Pada Delphi	II-29
Tabel 2.10 Tabel Operator Relasi Pada Delphi	II-30
Tabel 2.11 Tabel operator logika pada Delphi	II-31
Table 2.12 Fungsi Pin dalam DB 9	II-42
Tabel 2.13 Spesifikasi RS 232.....	II-43
Tabel 3.1 Merubah isi Properti pada Layar Utama.....	III-55
Tabel 3.2 Merubah isi Properti pada Layar Biodata.....	III-59
Table 4.1 Jarak Sinyal Yang Di Terima Oleh RFID.....	IV-70

DAFTAR GAMBAR

Gambar 2.1	Transponders Atau Tag.....	II-8
Gambar 2.2	Konfigurasi Pin ID – 12.....	II-9
Gambar 2.3	Lembar Kerja Delphi	II-13
Gambar 2.4	Database dalam MS Access	II-39
Gambar 2.5	Object Tabel.....	II-40
Gambar 2.6	Konfigurasi Pin Konektor RS-232.....	II-41
Gambar 3.1	Blok Diagram	III-44
Gambar 3.2	Rangkaian RFID	III-46
Gambar 3.3	Conection Hyper-Terminal	III-47
Gambar 3.4	Conect Using	III-47
Gambar 3.5	Serial Port Parameter	III-48
Gambar 3.6	Tampilan Kode Tag RFID pada Hyper-Terminal.....	III-48
Gambar 3.7	Flow Chart Program RFID.....	III-49
Gambar 3.8	Tampilan Utama MS Access	III-58
Gambar 3.9	Blank Database	III-53
Gambar 3.10	Tampilan Tabel Database	III-54
Gambar 3.11	Tampilan Design Menu Utama.....	III-54
Gambar 3.12	Tampilan Design Menu Biodata.....	III-58
Gambar 4.1	Menu Utama Program.....	IV-62



BAB I

PENDAHULUAN

1.1. Latar Belakang

Sistem parkir yang digunakan saat ini, khususnya di kota Malang, sebagian besar masih menggunakan sistem parkir bersifat manual. Dalam pengertian bahwa jika suatu kendaraan akan memasuki area parkir, maka petugas pada pintu masuk akan mencatat nomor pelat kendaraan tersebut pada karcis parkir. Kemudian karcis parkir tersebut akan di berikan kepada pemilik kendaraan apabila telah membayar biaya parkir secara tunai, lalu kendaraan tersebut sudah dapat memasuki lahan parkir. Pada saat kendaraan akan keluar dari area parkir, maka pengemudi kendaraan harus memberikan karcis tersebut kepada petugas pada pintu keluar sebagai bukti bahwa nomor pelat kendaraan yang tertulis pada karcis parkir sesuai dengan nomor pelat pada kendaraan tersebut. Bila sesuai, maka kendaraan tersebut baru boleh keluar dari area parkir. Sistem parkir manual tersebut kurang aman dan efisien karena jika pada saat suatu kendaraan akan keluar dari area parkir, kendaraan di kendari oleh orang lain yang memiliki karcis parkir dengan nomor pelat kendaraan yang sesuai dengan nomor pelat pada kendaraan, maka kendaraan tersebut tetap dapat keluar dari area parkir.

Bertolak dari latar belakang tersebut, maka timbul ide untuk membuat suatu sistem parkir prabayar yang dapat mengidentifikasi data pemilik kendaraan berupa no. ID kartu, nama, alamat dan nomor pelat kendaraan yang

akan di tampilkan pada suatu PC baik saat kendaraan akan memasuki area parkir maupun keluar dari area parkir. Sehingga petugas parkir akan mencocokkan data pemilik kendaraan pada STNK dengan data pemilik kendaraan yang ditampilkan pada PC, kendaraan baru boleh memasuki atau keluar dari area parkir jika data tersebut sesuai.

1.2. Tujuan

Tujuan yang hendak dicapai dalam pembuatan tugas akhir ini sebagai berikut:

- Membuat suatu sistem parkir prabayar yang dapat mengidentifikasi data pemilik kendaraan berupa no.ID kartu, nama, alamat, no. pelat kendaraan dan jumlah uang.
- Manfaat teknologi RFID dalam mengidentifikasi data pemilik kendaraan.
- Membuat suatu sistem parkir prabayar dan aplikasinya yang berguna pada kondisi di Indonesia.

1.3. Perumusan Masalah

Yang menjadi titik fokus pada tugas akhir ini adalah bagaimana membuat sistem parkir yang dapat melakukan identifikasi terhadap data pemilik kendaraan, menghitung biaya parkir kendaraan, serta dapat mengidentifikasi data pemilik kendaraan berupa no.ID, nama, alamat dan nomer pelat kendaraan, dan juga dalam merancang dan membuat software sehingga dapat mengoperasikan *RFID* untuk sistem parkir prabayar.

1.5.1. Tinjauan Pustaka

Tinjauan pustaka ini meliputi studi mengenai RFID, teori, dari IC RFID Reader(ID 12), komponen tambahan pada Borland Delphi 7 untuk meng-akses port serial. Dan intruksi – intruksi beserta fitur-fitur yang dimiliki oleh masing- masing komponen agar dapat diterapkan pada sistem parkir prabayar ini.

1.5.2. Pengumpulan Data

Pengumpulan data yang dilakukan meliputi pengumpulan data RFID reader(ID 12), RS 232 dan library properties.

1.5.3. Perencanaan Perangkat Keras

Perencanaan perangkat keras pada sistem parkir prabayar ini meliputi Perencanaan rangkaian RFID.

1.5.4. Perencanaan Perangkat Lunak

Perencanaan perangkat lunak pada sistem parkir prabayar ini menggunakan bahasa pemrograman Borland Delphi 7. Dan untuk database digunakan Microsoft Access 2007.

1.5.5. Pembuatan Perangkat Keras

Pembuatan perangkat keras pada sistem parkir prabayar ini meliputi pembuatan rangkaian RFID Reader yang dapat dihubungkan langsung dengan PC.

1.5.6. Pembuatan Perangkat Lunak

Pembuatan perangkat lunak pada sistem parkir prabayar ini meliputi pembuatan perangkat lunak untuk sistem parkir dan pembuatan database untuk menampung data sistem parkir ini.

1.4. Ruang Lingkup Pembahasan

Pada tugas akhir ini, akan dilakukan pembahasan mengenai hal sebagai berikut:

- Sistem komunikasi RFID (*Radio Frequency Identification*) dan proses pembacaan data dari *tag* RFID oleh *RFID Reader* ketika kartu dideteksi.
- Menghubungkan *RFID Reader* dengan PC melalui port serial (COM1).
- Pembacaan data yang masuk di serial port melalui bahasa pemrograman Borland Delphi 7.
- Proses penyimpanan data kartu parkir (*tag* RFID) ke dalam database sistem.
- Proses penyimpanan data pelanggan ke dalam database sistem.
- Proses penghitungan biaya parkir dengan tarif sekali masuk.

1.5. Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penulisan tugas akhir ini meliputi Sembilan bagian, yaitu tinjauan pustaka, pengumpulan data, perencanaan perangkat keras, perencanaan perangkat lunak, pembuatan perangkat keras, pembuatan perangkat lunak, pengujian sistem, pengambilan kesimpulan, dan penulisan laporan tugas akhir.

1.5.7. Pengujian Sistem

Pengujian sistem pada tugas akhir ini di lakukan dengan mengoprasikan sistem parkir secara keseluruhan seperti saat di gunakan dalam sistem parkir yang sesungguhnya dan di lakukan pengujian proses transaksi parkir.

1.5.8. Pengambilan Kesimpulan

Pengambilan kesimpulan berdasarkan hasil pengujian sistem yang telah dilakukan pada sistem parkir prabayar yang telah dibuat meliputi antara lain: kesimpulan terhadap identifikasi sistem parkir prabayar terhadap kartu parkir dan pemotongan jumlah uang yang tersimpan terhadap perhitungan biaya parkir kendaraan.

1.5.9. Penulisan Laporan Akhir

1.6. Sistematika Penulisan

- **Bab 1 : PENDAHULUAN**

Dalam bab ini di paparkan hal – hal yang berhubungan dengan latar belakang, tujuan, perumusan masalah, ruang lingkup pembahasan serta metodologi penelitian yang digunakan dalam pembahasan tugas akhir ini.

- **Bab 2 : TEORI PENUNJANG**

Dalam bab ini akan dijelaskan mengenai RFID reader (ID 12).

- **Bab 3 : PERENCANAAN SISTEM**

Dalam bab ini akan dibahas mengenai perencanaan dan perealisasiian sistem parkir prabayar secara keseluruhan baik perangkat keras maupun perangkat lunak.

- **Bab 4 : PENGUJIAN SISTEM**

Dalam bab ini membahas tentang pengujian dan hasil yang diperoleh dari sistem parkir prabayar yang telah dibuat.

- **Bab 5 : KESIMPULAN DAN SARAN**

Dalam bab ini berisi kesimplan – kesimpulan yang diperoleh dari perencanaan tugas akhir ini serta saran – saran yang diberikan untuk penyempurnaan dan pengembangan sistem lebih lanjut.



BAB II

TEORI PENUNJANG

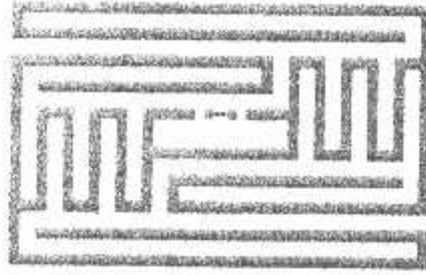
2.1 Radio Frekuensi Identification (RFID)

RFID (Radio Frekuensi Identification) merupakan identifikasi yang dilakukan dengan menggunakan gelombang RF (Radio Frekuensi). Sistem RFID sendiri terdiri dari dua bagian besar yaitu:

- *Transponders* atau sering disebut dengan *tag* (berisikan *microchip* dan *coil* antenna).
- *Readers* (biasa dikenal sebagai sensor RFID), yang digunakan untuk membaca identitas dari *transponders*.

2.1.1. RFID tag

Berdasarkan keaktifannya, *tag* RFID dapat dibedakan menjadi dua, yaitu *tag aktif* dan *tag pasif*. *Tag aktif* cenderung berukuran besar dan lebih mahal harganya karena *tag aktif* secara aktif memancarkan data kepada *RFID reader*. Sedangkan *tag pasif* mendapatkan power supply dari medan magnet yang dihasilkan *tag* itu sendiri dengan *RFID reader* untuk membangkitkan rangkaian *microchip* yang terdapat dalam *tag* tersebut dan akan memantulkan kembali gelombang RF yang ditransmisikan oleh *RFID reader* kepadanya, beserta informasi pada *tag* tersebut dengan melakukan modulasi pada gelombang yang dipantulkan tersebut.



Gambar 2.1. *Transponders* atau *tag*

(www.google.com *RFID Tag*)

Bagian *RFID tag* sebuah sirkuit terpadu (juga dikenal sebagai IC, microcircuit, microchip, silicon chip, atau chip) adalah miniaturized sirkuit elektronik (terutama terdiri dari perangkat semikonduktor, serta komponen pasif) yang telah diproduksi di permukaan yang tipis substrat dari bahan semikonduktor.

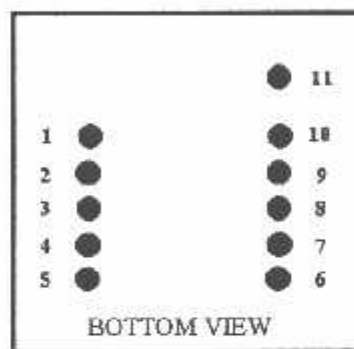
2.1.2. RFID Reader

Pada tugas akhir ini, jenis *RFID reader* yang digunakan adalah ID – 12 dengan spesifikasi sebagai berikut:

- Berbasis RFID reader ID-12 dengan frekuensi kerja 125 kHz untuk kartu berformat EM4001/sejenis dan memiliki jarak baca maksimal 8 cm.
- Kompatibel dengan varian RFID reader lainnya, antara lain: ID-2, ID-10, dan ID-20.
- Mendukung Varian RFID reader/writer, antara lain : ID-2RW, ID-12RW, dan ID-20RW.
- Mendukung varian RFID reader/writer, antara lain: ID-2RW, ID-12RW, dan ID-20RW.

- Mendukung format data ASCII (UART TTL/RS-232), Wiegand26, maupun Magnetic ABA Track2 (Magnet Emulation).
- Dilengkapi dengan buzzer sebagai indikator baca, serta LED sebagai indikator tulis.
- Tegangan input catu daya 9 - 12 VDC.

Untuk karakteristik fisik dan operasional pada ID-12, dapat dilihat pada gambar 2.1 dan table 2.1.. sedangkan untuk format data ID-12 baik dalam ASCII maupun *wiegand26*, dapat dilihat pada tabel 2.2.



Gambar 2.2. Konfigurasi Pin ID – 12

(www.google.com *RFID ID-12*)

Tabel 2.1 Karakteristik Fisik

Parameters	ID-2	ID-12	ID-20
Read Range	N/A (no internal antenna)	8+ cm	16+ cm
Dimensions	21 mm x 19 mm x 6 mm	26 mm x 25 mm x 7 mm	40 mm x 40 mm x 9 mm
Frequency	125 kHz	125 kHz	125 kHz
Card Format	EM 4001 or compatible	EM 4001 or compatible	EM 4001 or compatible
Encoding	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64
Power Requirement	5 VDC @ 13mA nominal	5 VDC @ 30mA nominal	5 VDC @ 65mA nominal
I/O Output Current	+/-200mA PK	-	-
Voltage Supply Range	+4.6V through +5.4V	+4.6V through +5.4V	+4.6V through +5.4V

(www.google.com/DataSheet/ID-12)

Tabel 2.2 Pin Description & Output Data Format

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 function	Card Present *	No function
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 (inverted)	Data *	Zero Output *
Pin 10	3.1 kHz Logic	Beeper / LED	Beeper / LED	Beeper / LED
Pin 11	DC Voltage Supply	+5V	+5V	-5V

(www.google.com DataSheet ID-12)

Sistem RFID terdiri dari tiga komponen, diantaranya seperti berikut :

- Tag : Ini adalah devais yang menyimpan informasi untuk identifikasi objek. Tag RFID sering juga disebut sebagai *transponder*. Format dari Tag pada perancangan ini adalah EM4001 atau tag kompatibel lainnya.
- Antena : Untuk menstransmisikan 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. Digunakan ID-12 sebagai RFID reader pada perancangan ini.

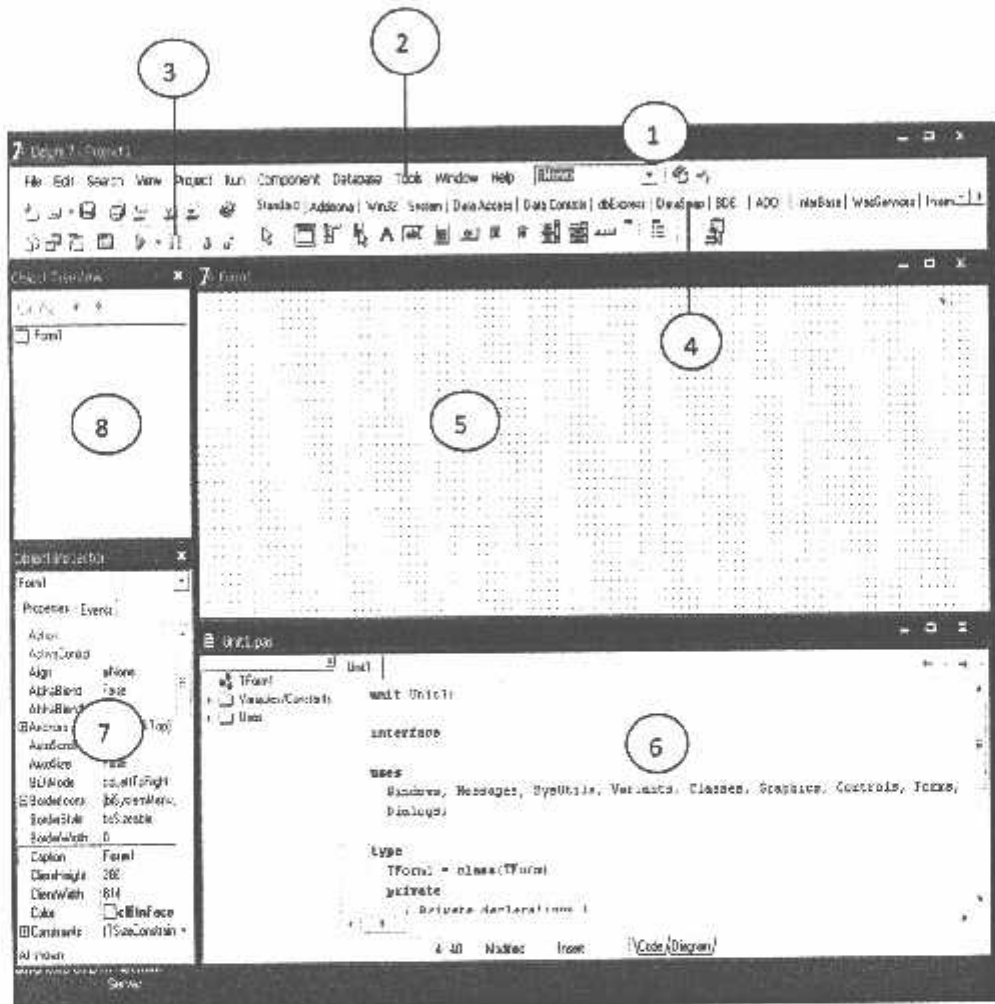
2.2 Borland Delphi 7.0

Delphi adalah salah satu pemrograman visual dilingkungan sitem operasi Microsoft Windows, berbasis bahasa PASCAL, sehingga bila kita telah mengetahui konsep dasar dan aturan-aturan yang berlaku dalam pemrograman PASCAL maka untuk masuk ke Delphi akan lebih mudah. Untuk mempermudah pemrograman dalam membuat suatu 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, object biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan *bahasa pemrograman* dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu.

2.2.1. IDE Delphi

Lingkungan pengembangan terpadu atau *Integrated Development Environment* (IDE) dalam program Delphi terbagi menjadi enam bagian utama yaitu *Main Window*, *ToolBar*, *Componen Palette*, *Form Designer*, *Code Editor* dan *Object Inspector*. Untuk lebih jelasnya terlihat pada gambar dibawah ini:



Gambar 2.3. Lembar Kerja Delphi

(M. Agus J. Alam, " Mengelola Database dengan Borland Delphi 7"
PT. Elex Media Komputindo)

APLIKASI RFID PADA SISTEM PARKIR PRABAYAR

TUGAS AKHIR

Disusun oleh :

NAMA : Bayu Bagus Triyo Wibowo

NIM : 05.52.206



**JURUSAN TEKNIK ELEKTRO D-3
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2009**

IDE merupakan sebuah lingkungan dimana semua tombol perintah yang diperlukan untuk mendesain aplikasi, menjalankan dan menguji sebuah aplikasi disajikan dengan baik untuk memudahkan mengembangkan program.

Jendela IDE Delphi 7.0 mempunyai perangkat-perangkat yang dapat dipergunakan untuk memudahkan seseorang programmer dalam membuat program. Perangkat-perangkat tersebut seperti terlihat pada gambar 2.3 diantaranya adalah:

1. *Main Window*

Jendela Utama ini adalah bagian dari IDE yang mempunyai fungsi yang sama dengan semua fungsi utama dari program aplikasi Windows lainnya. Jendela utama Delphi terbagi menjadi tiga bagian, diantaranya Main Menu, Toolbar, dan Component Palette.

2. *Main Menu*

Menu Utama pada Delphi memiliki kegunaan yang sama seperti program aplikasi windows lainnya. Dengan menggunakan fasilitas menu, dapat memanggil atau menyimpan program. Pada dasarnya semua perintah yang diberikan dapat ditemukan pada bagian menu utama.

3. *Toolbar*

Dengan *toolbar* dapat melakukan beberapa operasi pada menu utamadengan sebuah klik tunggal. Setiap tombol pada *toolbar* mempunyai sebuah *tooltip* yang berisi informasi mengenai fungsi dari tombol tersebut.

4. *Component Palette*

Component Palette merupakan bagian yang digunakan untuk meletakkan

berbagai komponen yang sesuai dengan kategorinya. Komponen tersebut berguna untuk mendesain user interface (antarmuka pemakai)dari program yang sedang dibuat.

5. *Form Designer*

Merupakan sebuah bidang jendela yang masih kosong. Ketika menjalankan Borland Delphi 7, secara otomatis *Form Designer* akan memanggil sebuah form yang bernama Form1.

6. *Code Editor*

Pada bidang ini dapat menuliskan kode-kode program dan logika program dan bahasa Delphi untuk mengatur jalannya program. Antara Form Designer dan Code Editor merupakan dua bagian yang berkaitan.

7. *Object Inspector*

Dengan perangkat ini dapat mengubah *property* dan *event* pada setiap object atau komponen. Object atau komponen yang satu dengan yang lain mempunyai *property* dan *event* yang berbeda. Jika menempatkan sebuah komponen pada suatu Form, komponen tersebut akan berisi nilai *default* dari Delphi. Misalnya komponen Label akan berisi nilai 13 dan 32 untuk *property* height dan width. Nilai *property* tersebut dapat diubah baik pada saat perancangan interface program maupun pada saat program berjalan dengan menggunakan kode program.

8. *Object Tree View*

Merupakan sebuah diagram pohon yang menggambarkan hubungan logis antara komponen visual dan non visual yang terletak pada form, data module

atau frame. Semua object yang anda pakai pada form, data module atau frame akan muncul pada *object Tree View*.

2.3.2. Menu Borland Delphi

1. *Menu File*

Berisi Perintah dasar menu yang sering digunakan yang berhubungan dengan pengoperasian file.

2. *Menu Edit*

Berisi perintah yang digunakan untuk menyunting teks program dalam jendela code editor, menyunting komponen-komponen yang terletak pada bagian form design dan beberapa item lainnya.

3. *Menu Search*

Berisi perintah yang digunakan untuk menyunting teks program dalam jendela code editor, menyunting komponen-komponen yang terletak pada bagian form designer dan beberapa item lainnya.

4. *Menu View*

Berisi tentang perintah yang digunakan untuk menampilkan atau menyembunyikan jendela-jendela tertentu dalam program Delphi.

5. *Menu Project*

Berisi tentang perintah yang berhubungan dengan pengolahan suatu project.

6. *Menu Run*

Digunakan untuk menjalankan program dan melihat jalannya program. Dan dapat memantau jalannya program dengan memperhatikan procedure yang

dijalankan.

7. *Menu Component*

Digunakan untuk menambah atau menginstal komponen baru.

8. *Menu Database*

Digunakan untuk membuat, mengubah atau melihat database.

9. *Menu Tools*

Digunakan untuk mengubah option atau memanggil database desktop dari menu Delphi.

10. *Menu Help*

Menu Help digunakan untuk membuka lembaran kerja bantu Delphi.

2.3.3. Komponen Visual dan Nonvisual

Visual Component Library (VCL) adalah Library Delphi dalam lingkungan windows. Delphi membagi komponen-komponen ke dalam dua kelompok besar berdasarkan visualisasinya, yaitu:







1. Komponen Visual







Yaitu komponen yang akan ditampilkan dilayar pada saat aplikasi dijalankan. Komponen visual dipakai untuk membangun antarmuka dengan pemakai. Komponen ini dapat diubah ukurannya dengan cara drag pada salah satu kotak kecil yang ada disekeliling komponen. Contoh komponen visual adalah Button, Edit, Memo, Label, dan lain-lain.






2. Komponen Nonvisual

Yaitu komponen yang tidak ditampilkan dilayar pada saat aplikasi dijalankan, tetapi diperlukan dalam aplikasi. Ukuran komponen ini tidak dapat diubah.

Tabel 2.3 Macam-macam Komponen Standar Delphi

Komponen	Fungsi
 Pointer	Pointer adalah komponen khusus dan terdapat disetiap tab dalam component palatte. Komponen pointer adalah komponen select yang digunakan untuk memilih komponen dalam form designer.
 Frames	Sebuah kontainer yang digunakan untuk menampung komponen frame dapat diletakkan dalam form atau frame-frame yang lain.
 Main Menu	Komponen yang digunakan untuk membuat menu bar dan menu drop down. Komponen ini bersifat invisible.
 PopUp Menu	Komponen yang digunakan untuk membuat menu popup yang akan muncul jika pemakai melakukan proses klik kanan. Komponen ini bersifat invisible.
 Label	Komponen ini hanya digunakan untuk menambahkan teks di dalam form
 Edit	Komponen yang digunakan untuk menerima satu baris teks yang merupakan data input pemakai. Komponen ini juga dapat digunakan untuk menampilkan teks.

 Memo	<p>Komponen Memo dipakai untuk memasukkan atau menampilkan beberapa baris teks di dalam form.</p>
 Button	<p>Komponen yang dipakai untuk membuat button yang akan dipakai untuk memilih pilihan di dalam aplikasi. Jika mengklik komponen button tersebut maka suatu perintah atau kejadian akan dijalankan.</p>
 Check Box	<p>Komponen yang digunakan untuk memilih atau membatalkan suatu pilihan, yaitu dengan cara mengklik komponen.</p>
 Radio Button	<p>Komponen yang digunakan untuk memberikan sekelompok pilihan dan hanya ada satu pilihan yang dapat dipilih. Untuk memilih salah satu pilihan adalah dengan mengklik tombol pilihan yang diinginkan.</p>
 List Box	<p>Komponen yang digunakan untuk membuat sebuah daftar pilihan, dimana hanya ada satu pilihan yang dapat dipilih. Untuk mencari dan kemudian memilih salah satu pilihan yang terdapat di dalam daftar dapat menggunakan batang penggulung.</p>
 Combo Box	<p>Komponen yang digunakan untuk membuat sebuah daftar pilihan, dimana hanya ada satu pilihan yang dapat dipilih. Untuk mencari dan kemudian memilih salah satu pilihan yang terdapat di dalam daftar dapat menggunakan mengklik tombol drop down.</p>

 Scroll Bar	Komponen ini mempunyai fungsi yang sama seperti batang penggulung yang terdapat dalam program-program berbasis Windows pada umumnya.
 Group Box	Sebuah kontainer yang dapat digunakan untuk mengelompokkan komponen-komponen lain seperti Radio Button, CheckBox dan komponen kontainer yang lain.
 Radio Group	Komponen ini merupakan komponen kombinasi dari GroupBox dan didesain untuk membuat sekelompok Radio Button
 Panel	Sebuah kontainer yang dapat digunakan untuk membuat StatusBar, ToolBar dan ToolPalette.
 Action List	Komponen yang berisi daftar action yang digunakan bersama-sama dengan komponen dan control seperti item menu dan button.

(M. Agus J. Alam, " Mengelola Database dengan Borland Delphi 7"

PT. Elex Media Komputindo)

2.3.4. Informasi Sintak Umum

Untuk memulai penulisan listing program pada Delphi 7,terdapat hal-hal yang perlu diperhatikan dan harus dimengerti terlebih dahulu. Sehingga dalam membuat sebuah program atau aplikasi yang telah direncanakan, dapat berjalan dengan baik. Hal-hal tersebut antara lain;

2.3.4.1.Komentar

Komentar Dipakai untuk memberikan penjelasan atau keterangan di dalam baris program. Teks yang ditulis sebagai komentar tidak akan dikompilasi oleh kompiler pada saat program aplikasi di jalankan.

Untuk menuliskan sebuah komentar, programmer dapat menggunakan salah satu tanda dari tiga bentuk tanda yang disediakan, yaitu :

- Kurung Kurawal => { Komentar Program }
- Kurung Bintang => (* Komentar Program *)
- Slash Ganda -> // Komentar Program

Komentar dengan tanda kurung kurawal dan kurung bintang menggunakan tanda pembuka dan tanda penutup, sehingga teks yang dianggap sebagai komentar adalah teks yang terletak di antara tanda pembuka dan tanda penutup. Sedangkan tanda komentar slash ganda tidak menggunakan tanda penutup, sehingga semua teks yang terletak di belakan tanda slash ganda dianggap sebagai komentar.

2.3.4.2. Tipe Data

Pemilihan tipe data yang tepat akan sangat berguna dalam penghematan memori, kecepatan proses, ketelitian penghitungan dan lain-lain. Ada beberapa hal yang perlu diperhatikan dalam menentukan tipe data :

1. Penggunaan Memori

Masing-masing tipe data memiliki perbedaan dalam hal penggunaan memori. Dalam hal ini usahakan untuk menggunakan tipe data yang memiliki memori yang kecil.

2. Ketelitian Penghitungan

Walaupun unsur penghematan memori sangat penting untuk diperhatikan namun kebenaran dalam hal ketelitian penghitungan jauh lebih penting. Oleh sebab itu gunakan tipe data yang mempunyai tingkat ketelitian (presisi) paling tinggi.

Borland Delphi menyediakan banyak tipe data. Jenis maupun tipe data yang dapat digunakan antara lain adalah :

- **Tipe Integer**

Tipe data Integer digunakan untuk bilangan bulat atau bilangan yang tidak memiliki angka desimal. Tipe data Integer memiliki beberapa tipe yang tergantung pada rentang nilai dan ukuran penggunaan memori.

Tabel 2.4 Tabel Tipe Data Integer

Tipe	Rentang Nilai	Byte
Byte	0 .. 255	1
Word	0 .. 65535	2
ShortInt	-128 .. 127	2
SmallInt	-32768 .. 32767	2
Integer	-2147483648 .. 2147483648	4
Cardinal	0 .. 2147483647	4
LongInt	-2147483648 .. 2147483648	4

(M Agus J. Alam, " Mengelola Database dengan Borland Delphi 7"

PT Elex Media Komputindo)

Semakin besar rentang nilai bilangan, semakin besar pula memori yang diperlukan.

- **Tipe Real**

Tipe data Real digunakan untuk bilangan yang memiliki desimal (ditulis dalam $a \times 10^b$). Tipe data Real memiliki beberapa tipe yang tergantung pada rentang nilai dan ukuran penggunaan memori.

Tabel 2.5 Tabel Tipe Data Real

Tipe	Rentang Nilai	Byte
Real48	2.9×10^{-39} .. 1.7×10^{38}	1
Single	1.5×10^{-45} .. 3.4×10^{45}	2
Double	5.0×10^{-324} .. 1.7×10^{303}	2
Real	5.0×10^{-324} .. 1.7×10^{303}	2
Extended	3.6×10^{-496} .. 1.1×10^{486}	4
Comp	-2^{63+1} .. 2^{62-1}	4
Currency	-922337203685477.5808 .. 922337203685477.5807	4

(M Agus J. Alam, " Mengelola Database dengan Borland Delphi 7"

PT Elex Media Komputindo)

Tipe Double ekuivalen dengan tipe Real yang umum dipakai. Tipe Real48 digunakan untuk menjaga kompatibilitas dengan versi sebelumnya. Tipe Real48 menghasilkan proses yang lebih lambat. Tipe Extended menghasilkan ketelitian yang lebih baik tetapi kurang kompatibilitas jika program nantinya didesain untuk sharing dengan platform lain. Tipe Comp adalah tipe asli yang digunakan CPU Intel yang sebetulnya mewakili tipe Integer 64 bit, tetapi dikelompokkan dalam tipe Real karena tidak mempunyai sifat ordinal. Comp dipakai untuk mempunyai kompatibilitas dengan versi sebelumnya. Sebaiknya gunakan Int64 untuk kerja yang lebih baik. Keuntungan dari penggunaan tipe data Currency adalah :

- ❖ Tipe data Currency mempunyai ketelitian yang lebih tinggi dalam menangani bilangan yang cukup besar.
- ❖ Tipe data Currency dipakai dalam CurrencyField dan komponen lain dan kompatibel dengan tipe database yang menyatakan uang.

- **Tipe Boolean**

Tipe data Boolean digunakan untuk data logika yang hanya berisi True (Benar) dan False (Salah). Tipe data Boolean yang dapat digunakan antara lain :

Tabel 2.6 Tabel Tipe Data Boolean

Tipe	Byte
Boolean	1
ByteBool	1
WordBool	2
LongBool	4

(M.Agus J. Alam, " Mengelola Database dengan Borland Delphi 7"
PT. Elex Media Komputindo)

Dari beberapa tipe yang ada disarankan untuk menggunakan tipe Boolean, sedangkan untuk tipe lain hanya untuk menjaga kompatibilitas dengan program lain yang menggunakan tipe yang sama. Variabel tipe data Boolean dapat menerima penggunaan operator logika AND, OR, dan NOT.

- **Tipe Character**

Tipe data Character digunakan untuk menyatakan sederetan karakter satu huruf dalam bentuk tiga tipe, yaitu :

Tabel 2.7 Tabel Tipe Data Character

Tipe	Rentang Nilai	Byte
Char	1 Karakter ANSI	1
AnsiChar	1 Karakter ANSI	1
WideChar	1 Karakter Unicode	2

(M.Agus J. Alam, " Mengelola Database dengan Borland Delphi 7"
PT. Elex Media Komputindo)

- **Tipe String**

Tipe data string digunakan untuk menyatakan sederetan karakter, misalnya nama, alamat, kota, dan lain-lain. Adapun tipe-tipe dari data string, yaitu :

Tabel 2.8 Tabel Tipe Data String

Tipe	Rentang Nilai	Byte
ShortString	2 - 256	256 Karakter
AnsiString	4 - 2 GB	2 ³¹ Karakter
String	256 - 3 GB	2 ³¹ Karakter
WideString	4 - 2 GB	2 ³¹ Karakter

(M. Agus J. Alam, " Mengelola Database dengan Borland Delphi 7")

(PT Elex Media Komputindo)

Tipe ShortString berfungsi untuk menyesuaikan kompatibilitas dengan versi sebelumnya, sedangkan AnsiString dan WideString dapat digunakan untuk menyimpan karakter Unicode. Variabel dengan tipe data string mampu menangani data String yang hampir tidak terbatas (3 GB).

3. Variabel

Variabel adalah suatu tempat yang dialokasikan dalam memori yang diberi nama (sebagai pengenal) untuk menampung suatu data. Pendeklarasian variabel mempunyai susunan sebagai berikut :

var

nama_variabel : tipe_variabel;

Contoh :

```
var
```

```
a : Integer;
```

Contoh diatas mendeklarasikan variabel a dengan tipe Integer. Untuk mendeklarasikan beberapa variabel, perintah `var` cukup dituliskan hanya satu kali.

Contoh :

```
var
```

```
a : Integer;
```

```
b : String;
```

```
c : Real;
```

Contoh diatas mendeklarasikan variabel a, b dan c yang masing-masing bertipe Integer, String dan Real. Untuk mendeklarasikan beberapa variabel dengan tipe yang sama, tipe datanya cukup dituliskan hanya satu kali.

Contoh :

```
var
```

```
a, b, c : Integer;
```

```
x, y, z : String;
```

4. Operator

Dalam melaksanakan proses pengolahan data, Delphi menyediakan berbagai operator dengan urutan atau derajat proses pelaksanaan yang berbeda untuk beberapa operator yang dilibatkan pada suatu proses. Berikut ini adalah derajat proses pengolahan data yang dimiliki delphi :

Apabila terdapat ekspresi $A + B * C$, maka operasi yang akan dikerjakan terlebih dahulu $B * C$, baru setelah itu hasilnya ditambahkan dengan A. Urutan proses tersebut terjadi karena proses perkalian mempunyai urutan derajat pengoperasian yang lebih tinggi dari penjumlahan. Namun jika bentuk ekspresi diubah menjadi $(A + B) * C$, maka operasi yang didahulukan $(A + B)$, kemudian hasilnya dikalikan dengan C. Hal ini menunjukkan bahwa tanda kurung digunakan untuk mendahulukan proses.

Apabila ada proses yang terdiri dari beberapa operator yang mempunyai urutan derajat yang sama, maka proses pengoperasian akan mendahulukan proses yang ditulis paling kiri.

- **Operator Assignment (Penugasan)**

Assignment atau operator penugasan dituliskan dengan bentuk “:=” (titik dua sama dengan) dan berfungsi untuk memasukkan suatu nilai data ke dalam sebuah variabel, dengan bentuk penulisan :

Nama_Variabel := Ekspresi;

Contoh :

Harga := 500;

Jumlah := 20;

Total := Harga * Jumlah;

- **Operator Aritmatika**

Operator aritmatika digunakan untuk melakukan operasi aritmatika. Delphi mempunyai beberapa operator aritmatika, yaitu seperti tabel dibawah ini.

Tabel 2.9 Tabel operator aritmatika pada Delphi

Operator	Fungsi	Tipe yang Diproses	Tipe Hasil Proses
*	Perkalian	Integer, Real	Integer, Real
/	Pembagian Real	Integer, Real	Integer, Real
+	Penjumlahan	Integer, Real	Integer, Real
-	Pengurangan	Integer, Real	Integer, Real
div	Pembagian Integer	Integer	Integer
mod	Sisa Hasil Bagi	Integer	Integer

(M Agus J. Alam, " Mengelola Database dengan Borland Delphi 7 "

PT. Elex Media Komputindo)

Contoh :

Angka := 15 * 2; //Hasil 30

Angka := 15 / 2; //Hasil 7.5

Angka := 15 + 2; //Hasil 17

Angka := 15 - 2; //Hasil 13

Angka := 15 div 2; //Hasil 7

Angka := 15 mod 2; //Hasil 1

Untuk operasi perpangkatan, Delphi tidak me-nyediakan operator untuk pemangkatan. Sebagai gantinya, dapat digunakan rumus :

$$ax = \exp(b * \ln(a))$$

Contoh :

Angka1 := 5;

Angka2 := 2;

Hasil1 := exp(Angka1 * ln(Angka2)); //Hasil 32

Hasil2 := exp(Angka2 * ln(Angka1)); //Hasil 25

- **Operator Relasi (Pembandingan)**

Operotor relasi berfungsi untuk membandingkan suatu nilai (ekspresi) dengan nilai (ekspresi) lain yang akan meng-hasilkan suatu nilai logika (Boolean) yaitu True atau False. Kedua data yang dibandingkan tersebut harus memiliki tipe data yang sama. Berikut ini adalah tabel operator relasi yang disediakan Delphi :

Tabel 2.10 Tabel operator relasi pada Delphi

Operator	Keterangan
=	Sama Dengan
>	Lebih Besar
<	Lebih Kecil
>=	Lebih Besar atau Sama Dengan
<=	Lebih Kecil atau Sama Dengan
<>	Tidak Sama Dengan

(M. Agus J. Alami, " Mengelola Database dengan Borland Delphi 7 "

PT. Elex Media Komputindo)

Contoh :

$x := 21 + 9 > 20;$ // menghasilkan *True*

$x := 21 - 9 > 20;$ // menghasilkan *False*

- **Operator Logika**

Operator logika digunakan untuk mengekspresikan suatu satu atau lebih data atau ekspresi logika (Boolean), menghasilkan data logika (boolean) baru. Tabel operator logika dengan hierarki dari atas ke bawah adalah sebagai berikut :

Tabel 2.11 Tabel operator logika pada Delphi

Operator	Keterangan
not	Tidak
and	Dan
or	Atau
xor	Exclusive Or

(M.Agus J. Alam, "Mengelola Database dengan Borland Delphi 7"
PT Elex Media Komputindo)

- ❖ **Operator Logika Not**

Operator logika Not merupakan operator yang menyatakan kondisi kebalikan dari suatu ekspresi.

Contoh :

$x := \text{not}(21 > 9);$ Hasil *False*

$x := \text{not}(21 < 9);$ Hasil *True*

$x := \text{not}(21 = 9);$ Hasil *True*

❖ Operator Logika And

Operator logika And hanya akan menghasilkan nilai True jika lebih dari satu ekspresi yang menggunakan operator And bernilai True. Apabila ada satu ekspresi atau lebih bernilai False maka operator logika And akan menghasilkan nilai False.

Contoh :

$x := (21 > 9) \text{ and } (19 < 71); // \text{Hasil True}$

$x := (21 < 9) \text{ and } (19 < 71); // \text{Hasil False}$

$x := (21 < 9) \text{ and } (19 > 71); // \text{Hasil False}$

❖ Operator Logika Or

Operator logika Or akan menghasilkan nilai True jika salah satu atau seluruh ekspresi yang menggunakan operator Or bernilai True. Operator Or hanya akan bernilai False jika semua ekspresi yang menggunakan operator Or bernilai False.

Contoh :

$x := (21 > 9) \text{ or } (19 < 71); // \text{Hasil True}$

$x := (21 < 9) \text{ or } (19 < 71); // \text{Hasil True}$

$x := (21 < 9) \text{ or } (19 > 71); // \text{Hasil False}$

❖ Operator Logika Xor

Operator logika Xor akan menghasilkan nilai True jika ekspresi yang terletak di kiri operator Xor berbeda dengan ekspresi disebelah kanannya. Operator Xor hamper memiliki fungsi yang sama dengan operator **tidak sama dengan**.

Contoh :

```
x := (9 < 21) xor (19 < 71); //Hasil False
```

```
x := (9 > 21) xor (19 > 71); //Hasil False
```

```
x := (9 < 21) xor (19 > 71); //Hasil True
```

```
x := (9 > 21) xor (19 < 71); //Hasil True
```

• Operator String

Operator String fungsinya untuk menggabungkan dua data string. Operator setring dilambangkan dengan bentuk "+".

Penggabungan dua buah string tersebut menghasilkan data yang berjenis string juga. Contoh :

```
x := 'Mata Kuliah ';
```

```
y := 'Pemrograman Lanjutan';
```

```
z := x + y;
```

Maka *z* akan berisi 'Mata Kuliah Pemrograman Lanjutan'

2.4. Dasar Basis Data

2.4.1. Konsep Basis Data

Basis Data (Database) dapat dibayangkan sebagai sebuah lemari arsip. Jika kita memiliki sebuah almari arsip dan berwenang untuk mengelolanya, maka kemungkinan besar kita akan melakukan hal-hal sebagai berikut:

- Memberi sampul/map pada kumpulan/bundel arsip yang akan disimpan
- Menentukan kelompok/jenis arsip
- Memberi penomoran dengan pola tertentu yang nilainya unik pada setiap sampul/map
- Menerapkan arsip-arsip tersebut dengan cara/urutan tertentu dalam almari dll

Upaya penyusunan/penempatan ini baru kita lakukan jika kita rasakan bahwa buku tersebut sudah cukup banyak.

2.4.2. Definisi

Basis data terdiri dari 2 kata, yaitu basis dan data. Basis adalah markas/gudang, tempat bersarang atau berkumpul, sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi dan kombinasinya.

Basis data dapat didefinisikan dari beberapa sudut pandang:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali secara cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Prinsip kerja dari basis data dan lemari arsip adalah sama, terutama adalah untuk pengaturan data/arsip. Selain itu, tujuan utamanya juga sama, yaitu untuk kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari besi atau kayu, maka basis data menggunakan media penyimpanan elektronik seperti disk (disket atau harddisk).

Satu hal yang harus diperhatikan bahwa basis data bukan hanya sekedar media penyimpanan data secara elektronik (dengan bantuan komputer), artinya tidak semua bentuk penyimpanan data elektronik dapat disebut basis data. Kita dapat menyimpan dokumen berisi data dalam file teks (dengan program pengolah kata), file spreadsheet dll, tetapi tidak dapat disebut sebagai basis data. Karena didalamnya tidak ada pemilahan dan pengelompokan data sesuai jenis/fungsi data, sehingga akan menyulitkan pencarian data kelak. Yang sangat ditonjolkan dalam basis data adalah pengaturan/ pemilahan/ pengelompokan/ pengorganisasian data yang akan kita simpan sesuai fungsi atau jenisnya.

Pemilahan/ pengelompokan/ pengorganisasian ini dapat berbentuk sejumlah file/ tabel terpisah atau dalam bentuk pendefinisian kolom-kolom/ field-field data setiap file/tabel.

2.4.3. Operasi Dasar Basis Data

Dalam sebuah disk, basis data dapat diciptakan dan dapat pula ditiadakan. Di dalam sebuah disk, kita dapat pula menempatkan beberapa (lebih dari satu) basis data. Sementara dalam sebuah basis data kita dapat menempatkan satu atau lebih file/tabel. Pada file/tabel inilah sesungguhnya data disimpan/ditempatkan.

Sebuah basis data umumnya dibuat untuk mewakili sebuah semesta data yang spesifik. Misalnya basis data kepegawaian, basis data akademik, basis data inventori (pergudangan) dan sebagainya. Dalam basis data akademik, misalnya kita dapat menempatkan file mata pelajaran, file guru, file jadwal, file nilai dan seterusnya.

Oleh karena itu operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data dapat meliputi:

- Pembuatan basis data baru (*create database*) yang identik dengan pembuatan lemari rasip yang baru.
- Penghapusan basis data (*drop database*) yang identik dengan perusakan lemari rasip.
- Pembuatan file/ tabel baru ke suatu basis data (*create table*), identik dengan penambahan map arsip baru.
- Penghapusan file/tabel dari suatu basis data (*drop tabel*).

- Penambahan/pengisian data baru ke sebuah file/table di sebuah basis data.
- Pengambilan data dari sebuah file/tabel.
- Perubahan data dari sebuah file/tabel.
- Penghapusan data dari sebuah file/tabel.

Operasi-operasi yang berkaitan dengan pembuatan objek (basis data dan tabel) merupakan operasi awal yang hanya dilakukan sekali dan berlaku seterusnya, sedangkan operasi-operasi yang berkaitan dengan pengisian tabel merupakan operasi-operasi rutin yang dilakukan berulang-ulang dan karena itu operasi-operasi inilah yang lebih tepat mewakili aktivitas pengelolaan (management) dan pengolahan (processing) data dalam basis data

2.5. Microsoft Access

Microsoft Access (atau Microsoft Office Access) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi terakhir adalah Microsoft Office Access 2007 yang termasuk ke dalam Microsoft Office System 2007.

Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/ *programmer* yang mahir dapat menggunakannya untuk

mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.

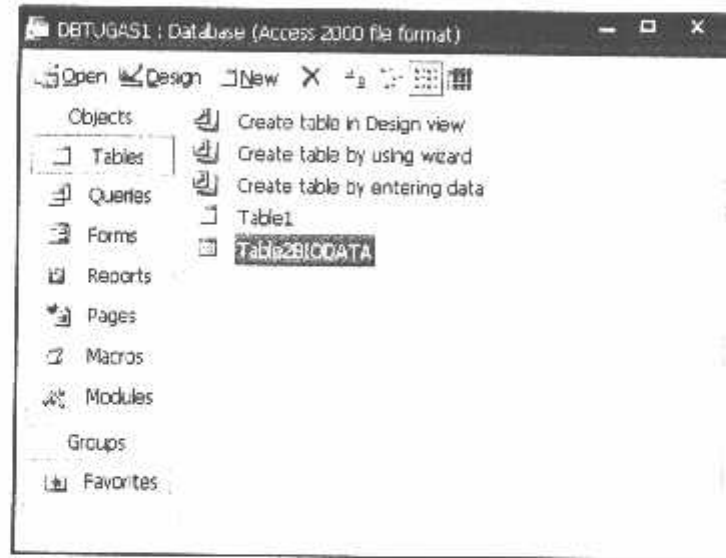
2.5.1. Keunggulan Microsoft Access

Salah satu keunggulan Microsoft Access dilihat dari perspektif *programmer* adalah kompatibilitasnya dengan bahasa pemrograman Structured Query Language (SQL); query dapat dilihat dan disunting sebagai statemen-statement SQL, dan statemen SQL dapat digunakan secara langsung di dalam Macro dan VBA Module untuk secara langsung memanipulasi tabel data dalam Access. Para pengguna dapat mencampurkan dan menggunakan kedua jenis bahasa tersebut (VBA dan Macro) untuk memprogram *form* dan logika dan juga untuk mengaplikasikan konsep berorientasi objek.

Microsoft SQL Server Desktop Engine (MSDE) 2000, yang merupakan sebuah versi mini dari Microsoft SQL Server 2000, dimasukkan ke dalam Office XP Developer Edition dan dapat digunakan oleh Microsoft Access sebagai alternatif dari Microsoft Jet Database Engine.

2.5.2. Database di dalam MS Access

Pengertian *database* pada MS Access adalah sekumpulan objek yang terdiri dari *Tabel, Query, Form, Report, Pages, Macro* dan *Module*. Object-object ini ditampung dalam satu wadah atau database. Jadi MS Access yang dimaksud database adalah kumpulan dari *Tabel, Query, Form, Report, Pages, Macro* dan *Module*. Database di MS Access selalu terdiri dari objek terdapat juga groups.



Gambar 2.4. Database dalam MS Access

(www.google.com Tutorial Microsoft Acces 2007)



Gambar 2.5. Object Tabel

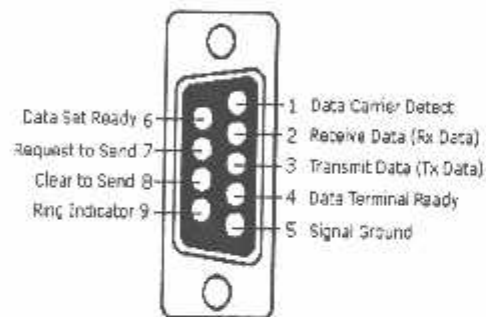
(www.google.com Tutorial Microsoft Acces 2007)

Didalam MS Access sebuah database disimpan dalam *satu file* berekstensi *mdb*. Jadi dalam Access satu file mewakili atau mempresentasikan satu database, dimana satu database ini bisa terdiri atas beberapa *Tabel, Query, Form, Report, Pages, Macro* dan *Module*.

2.6. RS-232

Karakteristik elektris dari RS-232 meliputi level tegangan, yaitu mempunyai tegangan keluaran +5V sampai +15V untuk mewakili level rendah (logika '0') dan -5V sampai -15V untuk mewakili level tinggi (logika '1').

Untuk kebutuhan *interface*, sinyal RS-232 memanfaatkan sebuah konektor dengan jumlah pin 9 atau 25. Konektor tersebut sering disebut DB 9 *conector*. Gambar tipe konektor DB 9 ditunjukkan dengan gambar di bawah ini.



Gambar 2.6. Konfigurasi Pin Konektor RS-232

(www.google.com Tutorial DB9)

Fungsi masing-masing pin adalah sebagai berikut:

Tabel 2.12. Fungsi Pin dalam DB 9

No	Nama	Fungsi
1	DCR (<i>Data carrier Detect</i>)	Mendeteksi sinyal carrier dari modem lain
2	RD (<i>Receive Data Line</i>) / (RxD)	Pengiriman data serial dari DCE ke DTE
3	TD (<i>Transmit Data Line</i>) / (TxD)	Pengiriman data serial dari DTE ke DCE
4	DTR (<i>Data Terminal Ready</i>)	Memberitahukan DCE bahwa DTE telah aktif dan siap untuk bekerja
5	Ground	Referensi semua tegangan antarmuka
6	DSR (<i>Data Set Ready</i>)	Memberitahukan DTE dan DCE telah aktif dan siap untuk bekerja
7	RTS (<i>Request To Send</i>)	Memberitahukan DCE bahwa DTE akan mengirim data
8	CTS (<i>Clear To Send</i>)	Memberitahukan DTE bahwa DCE siap menerima data
9	RI (<i>Ring Indikator</i>)	Aktif jika modem menerima sinyal ring pada jalur telepon

(www.google.com Tutorial DB9)

Untuk spesifikasi RS-232 dapat dilihat dalam tabel 2.13

Tabel 2.13. Spesifikasi RS-232

Keistimewaan	Karakteristik
Jenis Operasi	Single ended (tak seimbang)
Jenis Penggerak dan Penerimaan per jalur	1 driver 1 receiver
Data rate maksimum	20 kbps
Panjang saluran maksimum	50 ft (15 m)
Tegangan keluaran penggerak	$\pm 5 - \pm 15$ volt
Sensitivitas penerima	± 3 volt

(www.google.com Tutorial RS-232)

BAB III

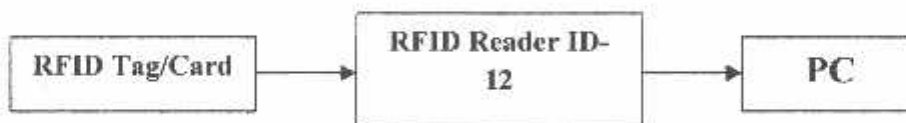
PERANCANGAN DAN PEMBUATAN ALAT

3.1. Pendahuluan

Dalam bab ini akan membahas tentang perencanaan dan pembuatan sistem yang digunakan untuk pengoperasian RFID pada suatu Perguruan Tinggi terutama di ITN Malang.

3.1.1. Blok Diagram Keseluruhan

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



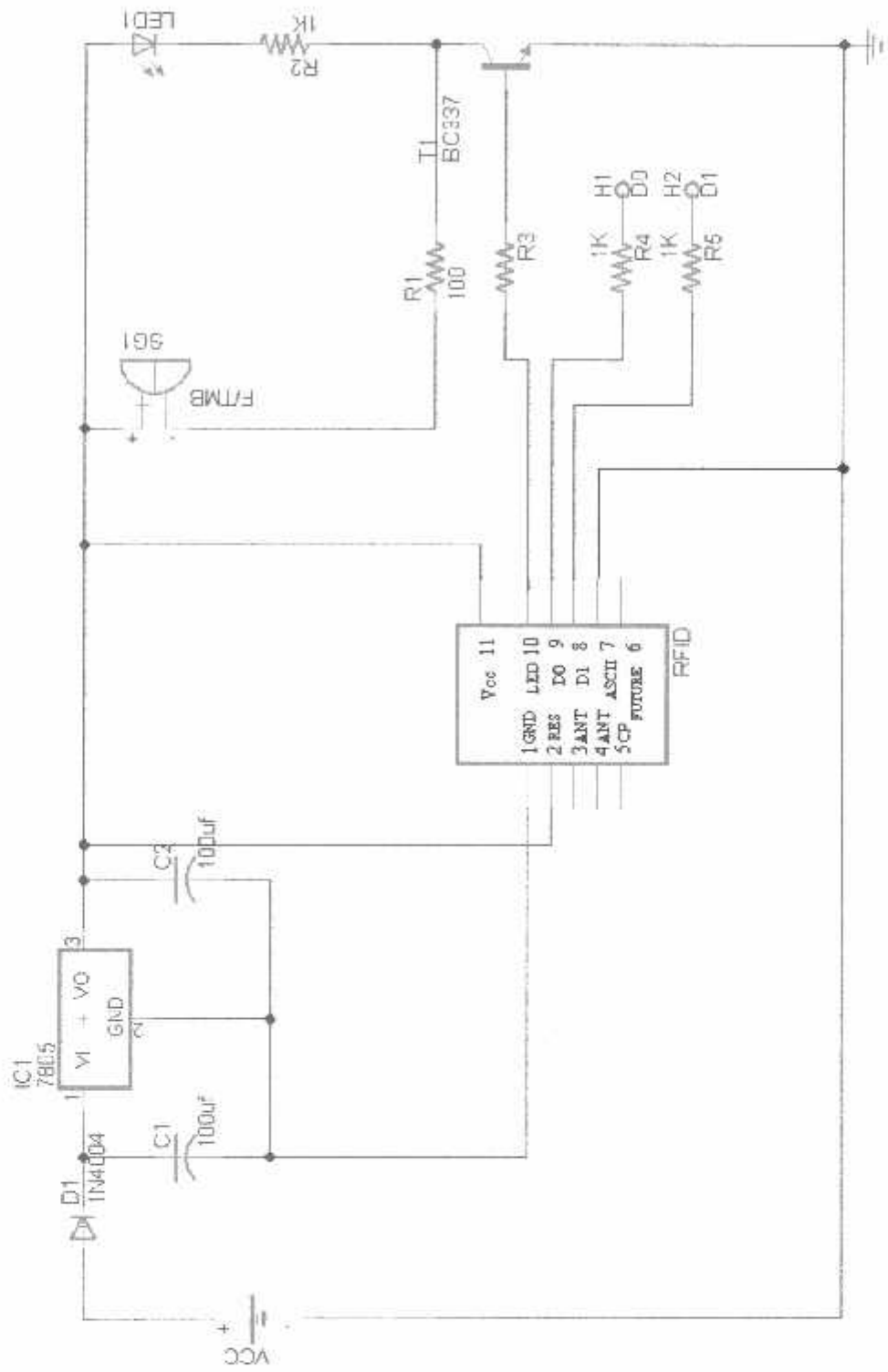
Gambar 3.1 Blok Diagram

RFID dihubungkan ke komputer melalui *COM port* dengan menggunakan kabel serial RFID Reader yang tersedia, RFID reader ID-12 yang digunakan di dalam rangkaian ini dapat diganti dengan ID-2 atau ID-20. Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, program RFID dapat dijalankan dan pilih *COM port* yang digunakan.

3.2. Perancangan Alat

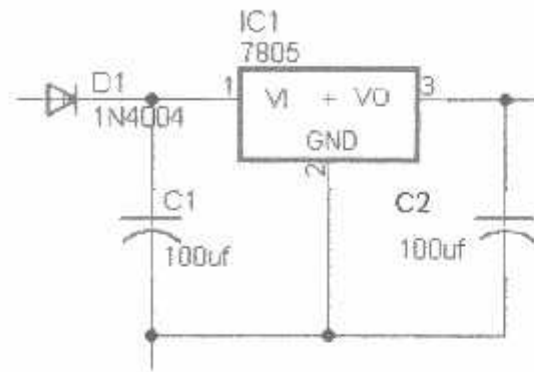
Kerja dari alat ini merupakan sebuah aplikasi sederhana yang menggunakan RFID Reader untuk membaca nomor ID dari *RFID transponder* dengan bantuan komputer. RFID Reader dihubungkan ke komputer melalui *COM port* dan nomor ID dari *RFID transponder* akan ditampilkan pada sebuah jendela aplikasi yang dikembangkan dalam bahasa Borland Delphi. Aplikasi ini hanya membutuhkan RFID Reader dengan kabel serialnya serta bantuan sebuah komputer.

Pada umumnya RFID tidak memancarkan sinyal terus-menerus, RFID hanya akan memancarkan sinyalnya apabila ada sinyal pemicu yang sesuai dengan tata cara pengiriman dan penerimaannya (protokol). Sinyal pemicu ini biasanya ditempatkan menjadi satu pada alat pemancar/penerima (reader dan antena) dengan range kisaran pembaca 8 cm serta bekerja pada frekuensi 125 KHz.



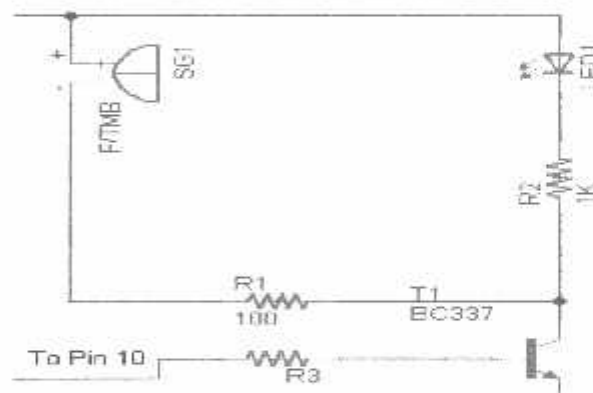
Gambar 3.2. Rangkaian RFID

3.2.1.1 Rangkaian Regulator



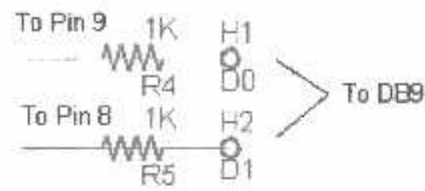
Rangkaian yang berfungsi sebagai penyetabil tegangan pada keseluruhan rangkaian RFID.

3.2.1.2 Rangkaian Driver Led dan Buzzer



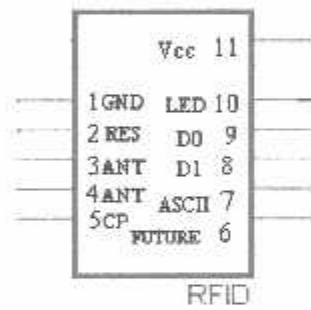
Rangkaian yang berfungsi sebagai switching LED dan juga Buzzer yang berfungsi sebagai indikator apabila Rangkaian bekerja.

3.2.1.3 Rangkaian Driver To DB9



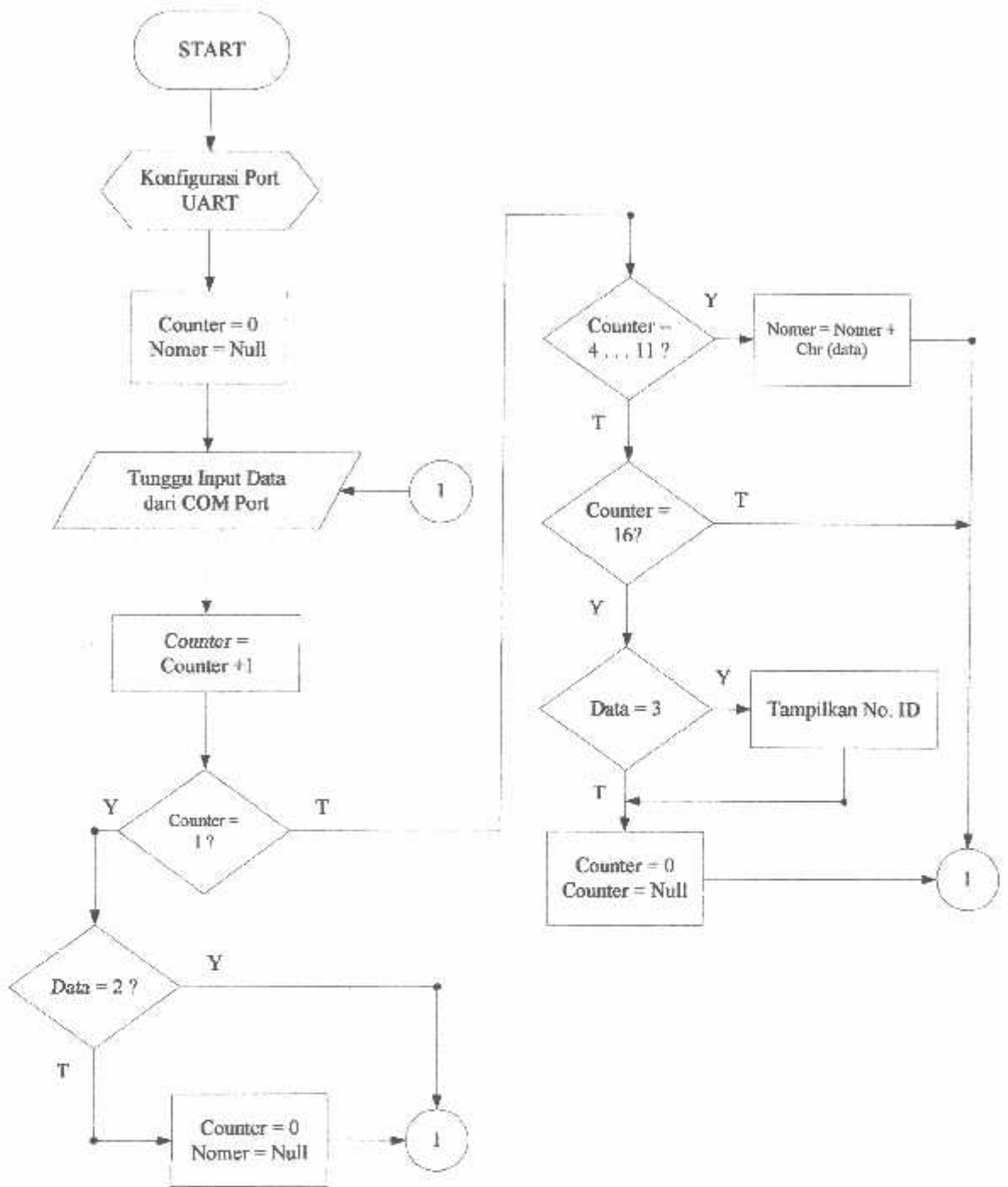
Rangkaian yang berfungsi sebagai penyetabil tegangan pada DB9 sehingga memungkinkan tidak akan terjadinya OverLoad tegangan pada DB9.

3.2.1.4 RFID ID 12



ID 12 Radio Frekuensi Identifikasi dengan antena internal yang sanggup membaca dengan Frekuensi 125 Khz dan dengan jarak baca antara ± 8 cm.

3.4. FlowChart



Gambar 3.7 FlowChart Program RFID

Program RFID ini secara garis besar bekerja sebagai berikut:

1. Program melakukan inisialisasi *COM port* yaitu pada *baudrate* 9600 bps, 8 bit data, 1 bit stop, tanpa bit *parity*, tanpa *flow control*. Kemudian program melakukan deklarasi variabel yang akan digunakan antara lain:
 - Counter adalah variabel bertipe *byte* dan digunakan sebagai penghitung jumlah byte data yang diterima.
 - Nomor adalah variabel bertipe *string* dan digunakan untuk menampung data ID dari *RFID transponder*.
 - Data adalah variabel bertipe *byte* yang digunakan untuk menampung data serial dari *COM port*.
2. Kemudian program memberi nilai nol pada variabel counter dan membuat variabel nomor menjadi *null* (kosong).
3. Program menunggu datangnya data serial dari *COM port*. Bila data serial datang maka program akan menyimpannya pada variabel data dan melanjutkan ke langkah berikutnya.
4. Program akan menambah variabel counter dengan 1, dan memeriksa nilai variabel counter. Jika variabel counter bernilai 1 maka program akan memeriksa apakah variabel data bernilai 2. Jika variabel data tidak bernilai 2 maka program akan memberi nilai variabel counter dengan 0 dan membuat variabel nomor menjadi *null*.

5. Jika variabel counter bernilai 4 hingga 11 maka program akan mengkonversi nilai variabel data menjadi *string* dan menambahkannya ke variabel nomor (sintaksis \rightarrow nomor:= nomor + chr[data]).
6. Jika nilai variabel counter adalah 16 maka program akan memeriksa apakah variabel data bernilai 3. Jika variabel data bernilai 3 maka program akan mengkonversi data dalam variabel nomor yang berformat heksadesimal menjadi format desimal lalu menampilkannya pada jendela program (melalui label1). Jika data tidak bernilai 3 maka program akan langsung melanjutkan ke langkah berikutnya.
7. Kemudian program memberi nilai nol pada variabel counter dan membuat variabel nomor menjadi *null* (kosong).
8. Lalu program kembali ke langkah 3.

3.5. Perancangan Perangkat Lunak (*Software*)

Perangkat lunak adalah sebuah jembatan yang menghubungkan keseluruhan komponen yang ada pada sebuah computer. Pada perancangan ini akan membahas tentang perangkat lunak untuk computer menggunakan *MS Access* dan *Delphi7*

3.5.1. Program Aplikasi Komputer

Program aplikasi adalah *software* di dalam komputer yang berfungsi untuk melakukan pengendalian. Program ini bertujuan untuk mengorganisasi komunikasi antara *MS Access* dengan *Delphi7*, sehingga perangkat lunak dalam program komputer perlu diketahui terlebih dahulu aturan-aturan atau protocol

komunikasi yang digunakan untuk mengatur jalannya komunikasi antara *MS Access* dengan *Delphi7*.

3.5.2. Perancangan Pembuatan Software Menggunakan Bahasa Pemrograman *Borland Delphi7* dan *MS Access*.

3.5.2.1. Penyusunan Desain Software

Dalam Penyusunan desain tampilan software perlu diperhatikan dengan teliti terutama pada saat membuat database pada *MS Access*. Sebelum masuk ke *Delphi7* terlebih dahulu buat database.

1. Membuat *Database*

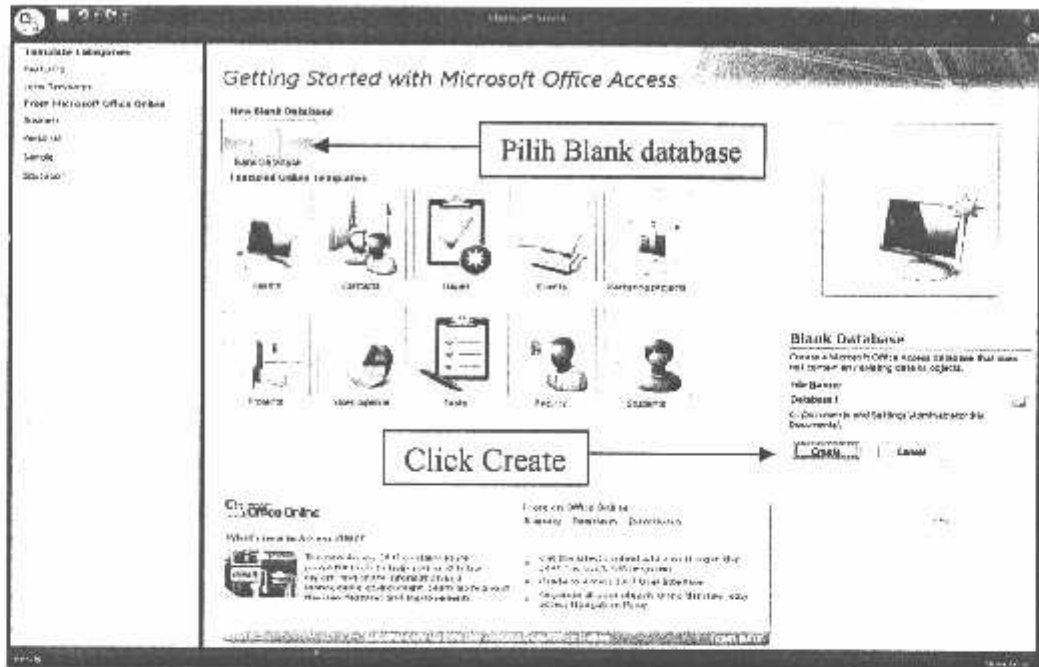
Langkah pertama yang dilakukan dalam membuat suatu Aplikasi *Database* adalah membuat *database*. Adapun langkah-langkah membuat database sebagai berikut:

- Membuka aplikasi *MS Access* akan muncul tampilan seperti gambar 3.8



Gambar 3.8. Tampilan Utama *MS Access*

- Setelah muncul tampilan utama pilihlah 'Blank Database' untuk membuat Database baru dan nama file/nama database yang akan dibuat.



Gambar 3.9. Blank Database

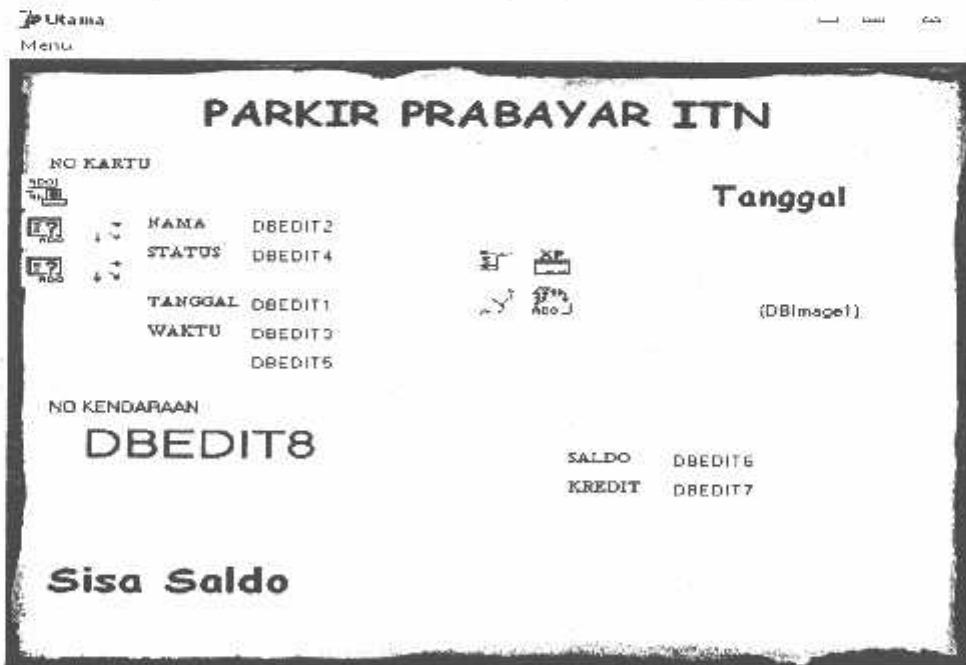
- Setelah sudah diberi nama misalnya 'Database1' Click tombol *create* untuk membuat database dengan nama 'Database 1'. Didalam MS Access satu database disimpandalam satu file. Dengan kata lain file Database 1 adalah database 1.
Setelah membuat database langkah selanjutnya adalah membuat isi dari database Database 1 yaitu berupa tabel-tabel.



Gambar 3.11. Tampilan tabel database

2. Membuat tampilan Desain koneksi dengan menggunakan Delphi7

- ❖ Menyusun desain Menu Utama seperti gambar dibawah ini :



Gambar 3.14. Tampilan Design Menu Utama

Untuk Mendesain Tampilan Design Menu Utama diperlukan komponen *ADO Connection*, *ADO Table*, *Data Source*, *Label 11* buah, *DBEdit 8* buah, satu buah *Dbimage1* satu buah *Edit1*, *DBQuery1*, *ADOCCommand1*, *Va Comm1* dan *DBnafigator1*. Kemudian merubah properti dari masing-masing komponen seperti dibawah ini.

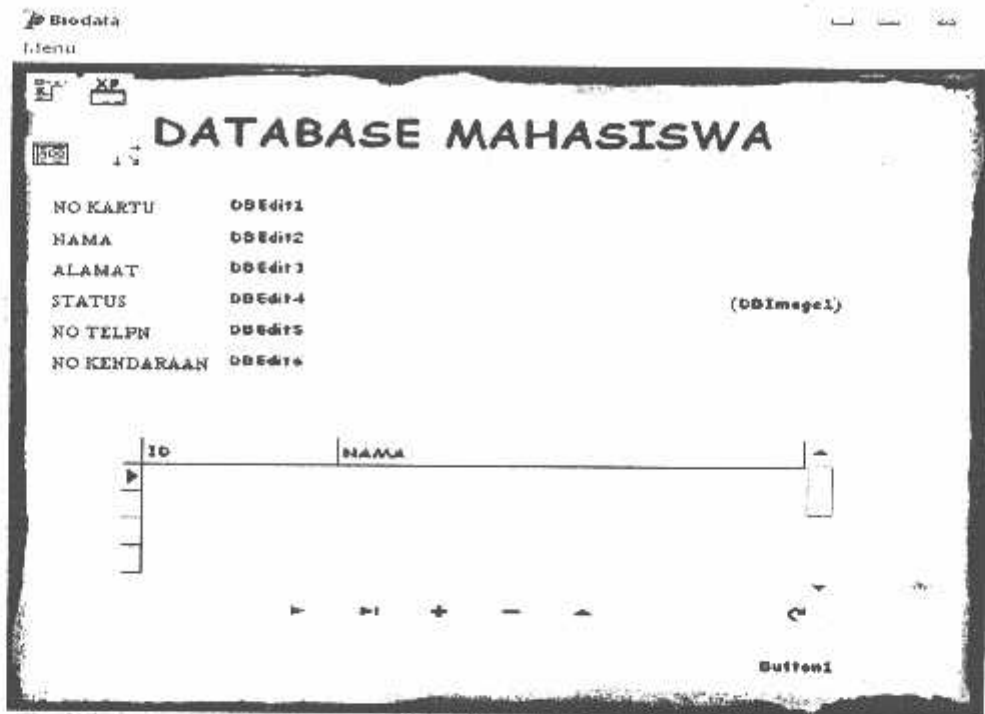
Tabel 3.1 Merubah isi Properti pada Layar Utama

Komponen	Properti	isi
ADOConection1 (Pada palet ADO)	Name	ADOConection1
	Connection string	Microsoft jet 4.0 OLE DB Provider
	LoginPrompt	False
	Connected	True
ADOQuery1 (Pada Palet ADO)	Connection	ADOConection1
	Name	ADOTable1
DataSource1	Name	DataSource1
	DataSet	ADOTable1
VaComm1 (pada palet TMS Async 32)	Name	VaCom1
	Devais Name	COM1
Label (pada palet standart)	Name	Label1
	Caption	No kartu
Label (pada palet standart)	Name	Label2
	Caption	NAMA
	FocusControl	DBEdit2
Label (pada palet standart)	Name	Label3
	Caption	Status
	FocusControl	DBEdit3
Label (pada palet standart)	Name	Label4
	Caption	Tanggal

	FocusControl	DBEdit5
Label (pada palet standart)	Name	Label5
	Caption	Waktu
	FocusControl	DBEdit1
Label (pada palet standart)	Name	Label6
	Caption	Saldo
	FocusControl	DBEdit4
Label (pada palet standart)	Name	Label7
	Caption	Kredit
	FocusControl	DBImage1
Label (pada palet standart)	Name	Label8
	Caption	Tanggal
	FocusControl	DBImage1
Label (pada palet standart)	Name	Label9
	Caption	Nominal
	FocusControl	DBImage1
Label (pada palet standart)	Name	Label10
	Caption	Parkir Prabayar ITN
	FocusControl	DBImage1
Label (pada palet standart)	Name	Label11
	Caption	Plat Nomor
	FocusControl	DBImage1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit1
	Data field	Tanggal
	Data Source	Data Source1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit2
	Data field	Nama
	Data Source	Data Source1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit3
	Data field	Waktu
	Data Source	Data Source1

DBEdit1 (Pada Palet Data Controls)	Name	DBEdit4
	Data field	Status
	Data Source	Data Source1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit5
	Data field	Nominal
	Data Source	Data Source1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit6
	Data field	Saldo
	Data Source	Data Source1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit7
	Data field	Kredit
	Data Source	Data Source1
DBEdit1 (Pada Palet Data Controls)	Name	DBEdit8
	Data field	Plat nomor
	Data Source	Data Source1
DBImage1 (pada palet Data Control)	Name	DBImage1
	Data Source	Data Source1
	Data Field	FOTO
Edit1 (pada palet Standart)	Name	Edit1
	Teks	(kosong)
MainMunu1 (pada palet Standart)	Name	Main Menu1

❖ Menyusun Desain Tampilan Pada Biodata Karyawan



Gambar 3.14. Tampilan Design Menu Biodata

Untuk mendesain Tampilan seperti diatas perlu beberapa komponen diantaranya ADOTable1, Data Source1, 8 buah Label, 6 buah DBEdit, DBImage, DBGrid, DBNavigator dan button1. Kemudian mengubah properti seperti tabel dibawah ini :

Tabel 3.2 Merubah isi Properti pada Layar Biodata

Komponen	Property	Isi
ADOTabel (Pada palet ADO)	Name	ADOTabel1
	Connection	Form_Utama.ADOConnection1
	Aktif	True
	Tabel Name	BIODATA
Data Source1	Name	Data Source1
	DataSet	ADOTable1
Label (pada palet standart)	Name	Label1
	Caption	No kartu
	FocusControl	DBEdit1
Label (pada palet standart)	Name	Label2
	Caption	Nama
	FocusControl	DBEdit2
Label (pada palet standart)	Name	Label3
	Caption	Alamat
	FocusControl	DBEdit3
Label (pada palet standart)	Name	Label4
	Caption	Status
	FocusControl	DBEdit4
Label (pada palet standart)	Name	Label5
	Caption	No_Telepon
	FocusControl	DBEdit5
Label (pada palet standart)	Name	Label6
	Caption	Nama
	FocusControl	DBEdit6
Label (pada palet standart)	Name	Label7
	Caption	Data base mahasiswa
	FocusControl	DBEdit7