

BAB III

ANALIS DAN PERANCANGAN SISTEM

Bab analisis dan perancangan sistem berisi pembahasan analisis dan perancangan aplikasi sistem. Pembahasan ditujukan untuk menguraikan kebutuhan-kebutuhan dalam pengembangan aplikasi.

3.1 Analisis Sistem

Sebelum menjalankan aplikasi sistem parkir, ada beberapa faktor yang harus dilakukan untuk memastikan aplikasi sistem parkir dapat berjalan (*running*), antara lain:

1. Memastikan database dapat bekerja dengan cara sinkronisasi dengan program, selanjutnya jalankan program/ sistem parkir,
2. Membuat master user untuk dapat menjalankan aplikasi sistem,
3. Login user,
4. Membuat beberapa master data, misal nomor kartu parkir,
5. Melakukan transaksi parkir masuk maupun parkir keluar,
6. Print laporan administrasi bukti operasional parkir masuk maupun keluar.

3.1.1 Evaluasi Sistem

Analisis Keunggulan Program

Aplikasi sistem kendali parkir menggunakan bahasa pemrograman Delphi merupakan aplikasi sistem yang dapat membantu sistem keamanan dan operasional perparkiran. Dengan adanya tambahan menu 'daftar *blacklist*/ khusus', aplikasi sistem kendali parkir diharap mampu mengurangi tindakan kriminalitas dibidang pencurian kendaraan bermotor.

Menu daftar *blacklist*/ khusus, berfungsi mengfilter setiap kendaraan yang hendak keluar dari tempat atau area parkir, sehingga jika ada kendaraan yang mana plat nomornya terdeteksi bermasalah, maka kendaraan tidak dapat diijinkan keluar dari tempat parkir.

3.1.2 Analisis Kebutuhan

Analisa kebutuhan sistem ini meliputi kebutuhan perangkat lunak dan kebutuhan perangkat keras.

Kebutuhan Perangkat Lunak:

1. Sistem Operasi Microsoft Windows 7

2. Software Delphi 6.0
3. MySQL 3.23.52-Max
4. ZeosDBO

Kebutuhan Perangkat Keras:

1. PC/ Laptop prosesor intel core 2 duo keatas
2. RAM 2Gb

3.1.3 Spesifikasi

Secara umum aplikasi sistem kendali parkir ini mempunyai spesifikasi sebagai berikut :

Sistem kendali parkir dibuat menggunakan pc/ laptop Toshiba satellite L830 dengan prosesor core i3 dan windows 7 ultimate sebagai operating sistem.

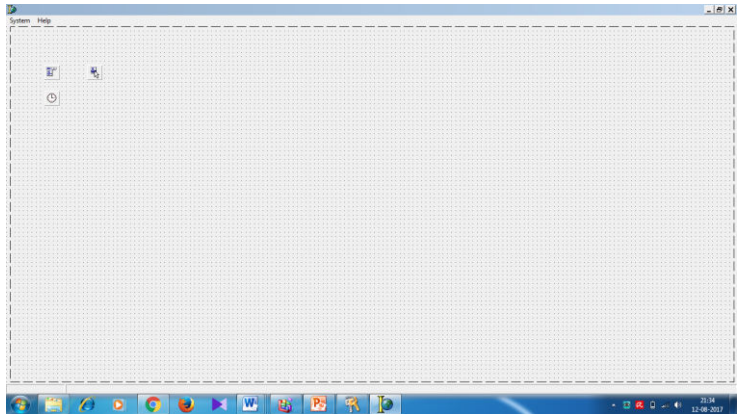
Sistem kendali parkir dibuat dengan menggunakan *Software* Delphi 6.0 dan MySQL sebagai database. MySQL yang digunakan adalah versi 3.23.52-Max.

Selain menggunakan software Delphi dan MySQL sebagai database, aplikasi ini juga dirancang dengan menggunakan zeos sebagai komponen library delphi untuk mengakses database mysql.

3.1.4 Desain Sistem

Setelah jelas apa saja yang menjadi spesifikasi kebutuhan, selanjutnya adalah membuat desain sistem kendali parkir. Adapun desain yang sudah dibuat ada di Gambar 3.1.

Desain dibuat seminimalis mungkin, tetapi secara garis besar aplikasi yang dibuat cukup kompleks.



Gambar 3.1. Desain Tampilan Aplikasi Sistem Kendali Parkir

3.1.5 Verifikasi

Pada tahapan uji coba berguna untuk mengetahui kesalahan atau kekurangan pada sistem, sehingga dapat dengan segera diperbaiki sebelum menuju ke tahap berikutnya.

3.1.6 Validasi

Setelah melewati tahap implementasi dan verifikasi maka tahap selanjutnya adalah validasi. Pada tahap ini dilakukan pengujian sistem secara menyeluruh, meliputi pengujian fungsional dan ketahanan sistem. Dari validasi ini dapat diketahui kesesuaian hasil perancangan dengan analisis kebutuhan yang diharapkan.

3.1.7 Finalisasi

Pada tahapan ini adalah tahapan hasil dari sistem yang sudah dirancang dan berjalan sesuai dengan rencana.

3.2 Gambaran Sistem

Sistem kendali parkir menggunakan bahasa pemrograman Delphi 6.0 dan database Mysql ini di buat seperti pada umumnya aplikasi parkir, yang mana didalamnya memiliki terdapat beberapa tools. Tools login, logout dan exit, ketiga tools tersebut masuk ke dalam menu system. Tool log in sangat mempengaruhi fungsi tools yang lain, karena saat tool log in di input, aplikasi akan melakukan filter system yang akan ditampilkan sesuai dengan akses yang diberikan kepada user. Jika user adalah orang yang mempunyai akses penuh, maka akan tampil semua menu dengan

tools yang lengkap. Adapun selain menu system yang di buat di dalam aplikasi ini adalah menu daftar, transaksi, laporan, master dan help.

Daftar menu mempunyai beberapa sub menu, yaitu daftar member dan daftar blacklist/spesial. Daftar member adalah pelanggan parkir yang mendaftarkan keanggotannya, sehingga nomor plat kendaraan akan diberikan nomor member sehingga akan mendapatkan nomor paten, untuk keistimewaan yang didapatkan disesuaikan dengan kebijakan peraturan perusahaan nantinya. Sedangkan untuk daftar blacklist/spesial ini yang membuat aplikasi ini berbeda, karena submenu inilah yang dapat meminimalisir tindak kejahatan. Karena pada sub menu ini dapat di input serial plat nomor kendaraan hasil curian atau kendaraan yang hilang, yang mana si pemilik lahan harus bekerjasama dengan kepolisian untuk mendapat daftar kendaraan yang hilang.

Di dalam menu transaksi ada sub menu transaksi parkir masuk dan parkir keluar. Menu laporan ada sub menu report parkir, report administrasi dan report real time. Menu master terdapat sub menu master user, master reason, master posisi parkir, master kartu parkir, master jenis kendaraan, master setup dan master modul. Sedangkan untuk menu terakhir ada menu help, yang berisi sub menu about dan help.

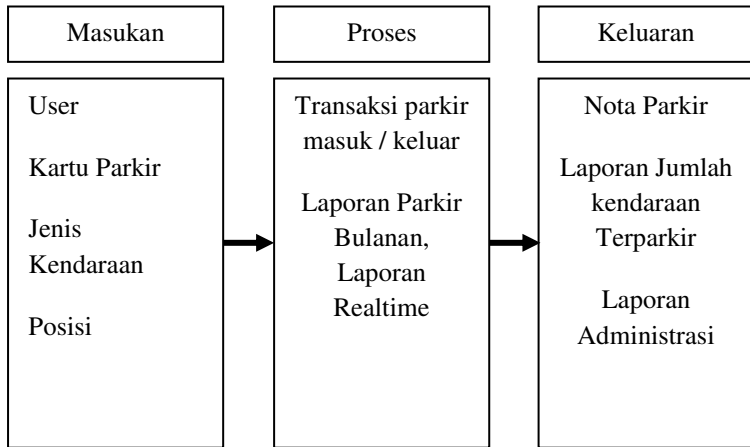
3.3 Perancangan Proses

Berikut ini dijelaskan mengenai alur sistem yang terdiri dari bagian-bagian sebagai berikut :

1. *Diagram Block*
2. *System Flow*
3. *Data Flow Diagram (DFD)*
4. *Entity Relationship Diagram (ERD)*
5. Struktur Database
6. Integrasi dengan Zeos

3.3.1 Block Diagram

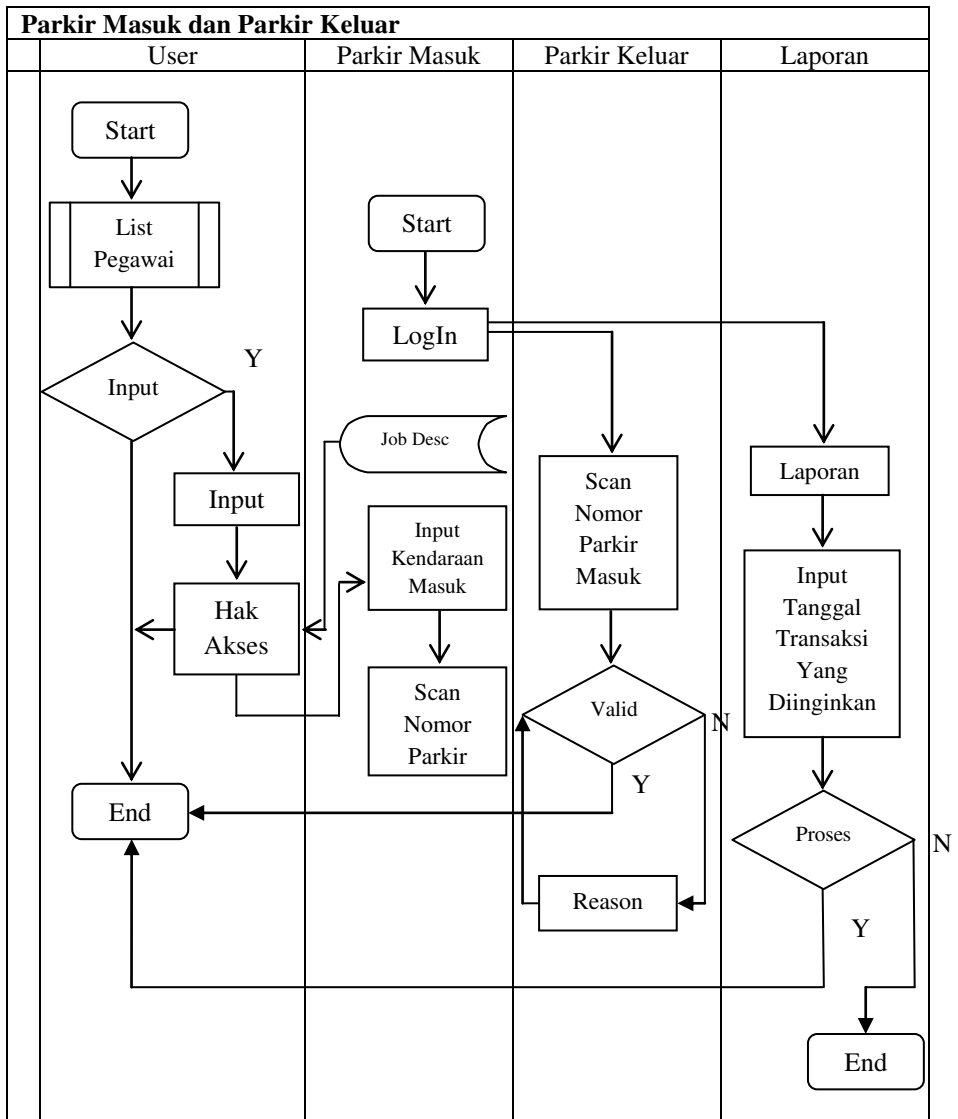
Block diagram dalam penelitian ini terdiri dari blok masukan, proses dan blok keluaran.



Gambar 3.2. Diagram Block Aplikasi

3.3.2 System Flow

Penggambaran arus informasi akan dijabarkan pada alur sistem yang akan diimplementasikan dengan komputer berupa penjaluran antara data, proses dan laporan. Bentuk desain umum Sistem kendali parkir masuk dan parkir keluar adalah sebagai berikut :



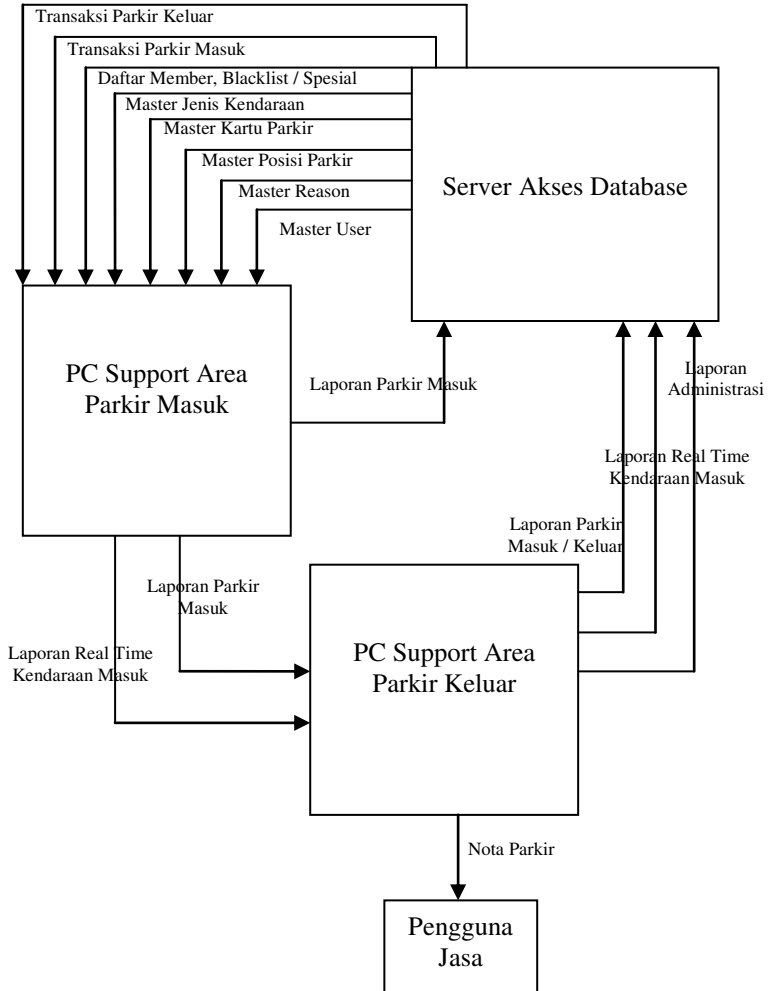
Gambar 3.3. System Flow Parkir Masuk dan Parkir Keluar

3.3.3 Data Flow Diagram (DFD)

DFD merupakan representasi grafik dalam menggambarkan arus data sistem secara terstruktur dan jelas sehingga dapat menjadi sarana dokumentasi yang baik.

A. Context diagram

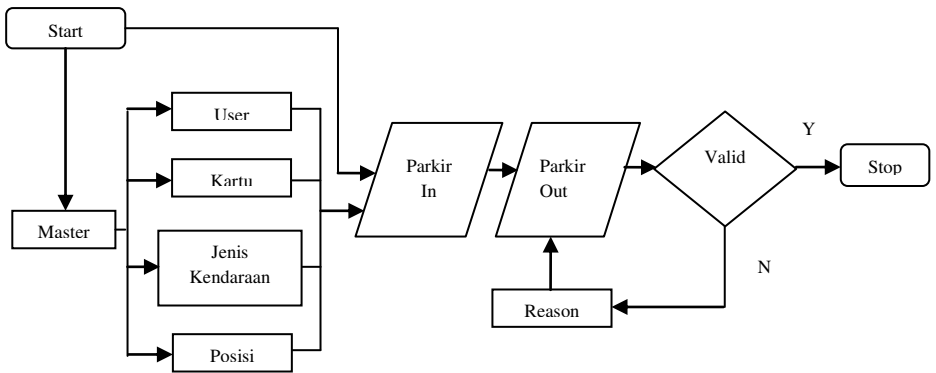
Diagram ini menggambarkan rancangan global/keseluruhan dari proses yang ada pada DFD. Kemudian digambarkan juga diagram berjenjang untuk context diagram yang dibuat seperti Nampak pada gambar 3.4.



Gambar 3.4. Context Diagram dari DFD

B. Diagram Alir

Berikut ini adalah diagram alir yang menggambarkan proses-proses secara umum yang akan dilalui dalam aplikasi ini, serta proses-proses utama sebagai berikut :



Gambar 3.5. Diagram Alir Aplikasi Sistem Parkir

3.3.4 Entity Relationship Diagram (ERD)

ERD Merupakan suatu desain sistem yang digunakan untuk merepresentasikan, menentukan dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan database. ERD juga menyediakan bentuk untuk menunjukkan struktur keseluruhan data dari pemakai. Dalam perancangan sistem ini telah dibuat ERD yang merupakan lanjutan dari pembuatan desain dengan menggunakan DFD.

3.3.5 Struktur Database

Struktur database menggambarkan data-data yang ada dalam database beserta tipe dan kegunaannya.

1. Tabel Blacklist

```
CREATE TABLE `t_blacklist` (  
  `fc_listcode` varchar(10) NOT NULL default "",  
  `fc_flaglist` char(1) default 'Y',  
  `fv_description` varchar(30) default NULL,  
  PRIMARY KEY (`fc_listcode`)  
) TYPE=MyISAM;
```

Name	Type	NULL	Default
fc_listcode	varchar(10)	No	
fc_flaglist	char(1)	Yes	Y
fv_descript...	varchar(30)	Yes	<NULL>

Tabel 3.7. Tabel Blacklist

2. Tabel Coba1

```
CREATE TABLE `t_coba1` (  
  `fv_description` char(30) default NULL,  
  `ttl_biaya` double(19,2) default NULL,  
  `ttl_jenis` bigint(21) NOT NULL default '0'  
) TYPE=MyISAM;
```

Name	Type	NULL	Default
fv_description	char(30)	Yes	<NULL>
ttl_biaya	double(19,2)	Yes	<NULL>
ttl_jenis	bigint(21)	No	0

Tabel 3.8. Tabel Coba1

3. Tabel Coba2

```
CREATE TABLE `t_coba2` (  
  `fv_description` char(30) default NULL,  
  `ttl_biaya` double(19,2) default NULL,  
  `ttl_jenis` bigint(21) NOT NULL default '0'  
) TYPE=MyISAM;
```

Name	Type	NULL	Default
fv_descript...	char(30)	Yes	<NULL>
ttl_biaya	double(19,2)	Yes	<NULL>
ttl_jenis	bigint(21)	No	0

Gambar 3.9. Tabel Coba2

4. Tabel Data Tes

```
CREATE TABLE `t_datetest` (  
  `fd_datetest` date default NULL  
) TYPE=MyISAM;
```

Name	Type	NULL	Default
fd_datetest	date	Yes	<NULL>

Gambar 3.10. Tabel Data Tes

5. Tabel Jenis

```
CREATE TABLE `t_jenis` (  
  `fc_jenis` char(2) NOT NULL default '',  
  `fv_description` varchar(30) default NULL,  
  `fm_harga` decimal(18,2) NOT NULL default '0.00',  
  `fn_countafterminute` int(3) unsigned default '0',  
  `fn_increment` int(10) unsigned default '0',  
  PRIMARY KEY (`fc_jenis`)  
) TYPE=MyISAM;
```

Name	Type	NULL	Default
Primary Index	fc_jenis		
fc_jenis	char(2)	No	
fv_description	varchar(30)	Yes	<NULL>
fn_harga	decimal(18,2)	No	0.00
fn_countaft...	int(3) unsigned	Yes	0
fn_increment	int(10) unsigned	Yes	0

Gambar 3.11. Tabel Jenis

6. Tabel Mdlprg

```
CREATE TABLE `t_mdprg` (
  `fc_flag` char(1) NOT NULL default 'N',
  `fc_mdprg` varchar(6) NOT NULL default "",
  `fc_namaprg` varchar(20) NOT NULL default "",
  `fc_description` varchar(30) default NULL,
  PRIMARY KEY (`fc_mdprg`)
) TYPE=MyISAM;
```

Name	Type	NULL	Default	Extras
Primary Index	fc_mdprg			unique
fc_flag	char(1)	No	N	
fc_mdprg	varchar(6)	No		
fc_namaprg	varchar(20)	No		
fc_description	varchar(30)	Yes	<NULL>	

Gambar 3.12. Tabel Mdlprg

7. Tabel Member

```
CREATE TABLE `t_member` (
  `fc_regcode` varchar(12) NOT NULL default "",
  `fc_nopol` varchar(10) NOT NULL default "",
  `fd_begindate` date default NULL,
  `fd_untildate` date default NULL,
  `fv_description` varchar(30) default NULL,
  `fc_hold` char(1) NOT NULL default 'N',
  PRIMARY KEY (`fc_regcode`,`fc_nopol`)
) TYPE=MyISAM;
```

Name	Type	NULL	Default	Extras
Primary Index	fc_regcode,fc...			unique
fc_regcode	varchar(12)	No		
fc_nopol	varchar(10)	No		
fd_begindate	date	Yes	<NULL>	
fd_untildate	date	Yes	<NULL>	
fv_description	varchar(30)	Yes	<NULL>	
fc_hold	char(1)	No	N	

Gambar 3.13. Tabel Member

8. Tabel Parkappr

```
CREATE TABLE `t_parkappr` (
  `fc_regcode` varchar(12) NOT NULL default "",
  `fc_nopol` varchar(10) NOT NULL default "",
  `fc_reasoncode` char(2) default NULL,
  `fc_jenis` char(2) default NULL,
  `fd_intrans` datetime default NULL,
  `fd_outtrans` datetime default NULL,
  `fm_totalpay` decimal(18,2) default NULL,
  `fd_totalwaktu` datetime default NULL,
  `fc_inposcode` char(2) default NULL,
  `fc_outposcode` char(2) default NULL,
  `fc_flagtrans` char(1) default NULL,
  PRIMARY KEY (`fc_regcode`,`fc_nopol`)
) TYPE=MyISAM;
```

Name	Type	NULL	Default	Extras
Primary Index	fc_regcode,fc...			unique
fc_regcode	varchar(12)	No		
fc_nopol	varchar(10)	No		
fc_reasoncode	char(2)	Yes	<NULL>	
fc_jenis	char(2)	Yes	<NULL>	
fd_intrans	datetime	Yes	<NULL>	
fd_outtrans	datetime	Yes	<NULL>	
fm_totalpay	decimal(18,2)	Yes	<NULL>	
fd_totalwaktu	datetime	Yes	<NULL>	
fc_inposcode	char(2)	Yes	<NULL>	
fc_outposcode	char(2)	Yes	<NULL>	
fc_flagtrans	char(1)	Yes	<NULL>	

Gambar 3.14. Tabel Parkappr

9. Tabel Parkin

```
CREATE TABLE `t_parkin` (
  `fc_nopol` varchar(10) NOT NULL default "",
  `fc_regcode` varchar(12) NOT NULL default "",
  `fd_intrans` datetime NOT NULL default '0000-00-00
00:00:00',
  `fc_listcode` varchar(10) default NULL,
  `fc_useridIN` varchar(8) default NULL,
  `fc_inposcode` char(2) default NULL,
  `fc_member` char(1) default 'N',
  `fv_description` varchar(30) default NULL,
  `fc_jenis` char(2) default NULL,
  PRIMARY KEY (`fc_nopol`,`fc_regcode`)
) TYPE=MyISAM;
```

Name	Type	NULL	Default	Extras
Primary Index	fc_nopol,fc_re...			unique
fc_nopol	varchar(10)	No		
fc_regcode	varchar(12)	No		
fd_intrans	datetime	No	0000-00-00 00...	
fc_listcode	varchar(10)	Yes	<NULL>	
fc_useridIN	varchar(8)	Yes	<NULL>	
fc_inposcode	char(2)	Yes	<NULL>	
fc_member	char(1)	Yes	N	
fv_description	varchar(30)	Yes	<NULL>	
fc_jenis	char(2)	Yes	<NULL>	

Gambar 3.15. Tabel Parkin

10. Tabel Parkout

```
CREATE TABLE `t_parkout` (
  `fc_regcode` varchar(12) NOT NULL default "",
  `fc_nopol` varchar(10) NOT NULL default "",
  `fc_useridIN` varchar(8) default NULL,
  `fc_useridOUT` varchar(8) default NULL,
  `fd_intrans` datetime NOT NULL default '0000-00-00
00:00:00',
  `fd_outtrans` datetime NOT NULL default '0000-00-00
00:00:00',
  `fn_totalwaktu` float NOT NULL default '0',
  `fm_totalpay` decimal(18,2) NOT NULL default '0.00',
```

```

`fc_inposcode` char(2) default NULL,
`fc_outposcode` char(2) default NULL,
`fc_flagtrans` char(1) NOT NULL default 'N',
`fv_description` varchar(30) default NULL,
`fc_jenis` char(2) default NULL
) TYPE=MyISAM;

```

Name	Type	NULL	Default
fc_regcode	varchar(12)	No	
fc_nopol	varchar(10)	No	
fc_useridIN	varchar(8)	Yes	<NULL>
fc_useridOUT	varchar(8)	Yes	<NULL>
fd_intrans	datetime	No	0000-00-00 00...
fd_outtrans	datetime	No	0000-00-00 00...
fn_totalwaktu	float	No	0
fm_totalpay	decimal(18,2)	No	0.00
fc_inposcode	char(2)	Yes	<NULL>
fc_outposcode	char(2)	Yes	<NULL>
fc_flagtrans	char(1)	No	N
fv_description	varchar(30)	Yes	<NULL>
fc_jenis	char(2)	Yes	<NULL>

Gambar 3.16. Tabel Parkout

11. Tabel Parkpos

```

CREATE TABLE `t_parkpos` (
  `fc_poscode` char(2) NOT NULL default "",
  `fv_description` varchar(30) default NULL,
  PRIMARY KEY (`fc_poscode`)
) TYPE=MyISAM;

```

Name	Type	NULL	Default	Extras
Primary Index	fc_poscode			unique
fc_poscode	char(2)	No		
fv_description	varchar(30)	Yes	<NULL>	

Gambar 3.17. Tabel Parkpos

12. Tabel Reason

```

CREATE TABLE `t_reason` (

```



```

`fc_reasoncode` char(2) NOT NULL default "",
`fv_description` varchar(30) default NULL,
PRIMARY KEY (`fc_reasoncode`)
) TYPE=MyISAM;

```

Name	Type	NULL	Default	Extras
Primary Index	fc_reasoncode			unique
fc_reasoncode	char(2)	No		
fv_description	varchar(30)	Yes	<NULL>	

Gambar 3.18. Tabel Reason

13. Tabel Registration

```

CREATE TABLE `t_registration` (
`fc_regcode` varchar(12) NOT NULL default "",
`fv_description` varchar(30) default NULL,
`fc_hold` char(1) NOT NULL default 'N',
PRIMARY KEY (`fc_regcode`)
) TYPE=MyISAM;

```

Name	Type	NULL	Default	Extras
Primary Index	fc_regcode			unique
fc_regcode	varchar(12)	No		
fv_description	varchar(30)	Yes	<NULL>	
fc_hold	char(1)	No	N	

Gambar 3.19. Tabel Registration

14. Tabel Setup

```

CREATE TABLE `t_setup` (
`fc_parameter` varchar(12) NOT NULL default "",
`fc_value` varchar(50) NOT NULL default "",
PRIMARY KEY (`fc_parameter`)
) TYPE=MyISAM;

```

Name	Type	NULL	Default	Extras
Primary Index	fc_parameter			unique
fc_parameter	varchar(12)	No		
fc_value	varchar(50)	No		

Gambar 3.20. Tabel Setup

15. Tabel User

```
CREATE TABLE `t_user` (  
  `fc_userid` varchar(8) NOT NULL default "",  
  `fc_password` varchar(12) NOT NULL default "",  
  `fv_username` varchar(30) default NULL,  
  `fc_flagadm` char(1) NOT NULL default 'N',  
  `fc_flagfo` char(1) NOT NULL default 'Y',  
  `fc_hold` char(1) NOT NULL default 'N',  
  PRIMARY KEY (`fc_userid`)  
) TYPE=MyISAM;
```

Name	Type	NULL	Default	Extras
Primary Index	fc_userid			unique
fc_userid	varchar(8)	No		
fc_password	varchar(12)	No		
fv_username	varchar(30)	Yes	<NULL>	
fc_flagadm	char(1)	No	N	
fc_flagfo	char(1)	No	Y	
fc_hold	char(1)	No	N	

Gambar 3.21. Tabel User

16. Tabel Usermdl

```
CREATE TABLE `t_usermdl` (  
  `fc_userid` varchar(8) NOT NULL default "",  
  `fc_mdprg` varchar(6) NOT NULL default "",  
  PRIMARY KEY (`fc_userid`,`fc_mdprg`)  
) TYPE=MyISAM;
```

Name	Type	NULL	Default	Extras
Primary Index	fc_userid,fc_m...			unique
fc_userid	varchar(8)	No		
fc_mdprg	varchar(6)	No		

Gambar 3.22. Tabel Usermdl

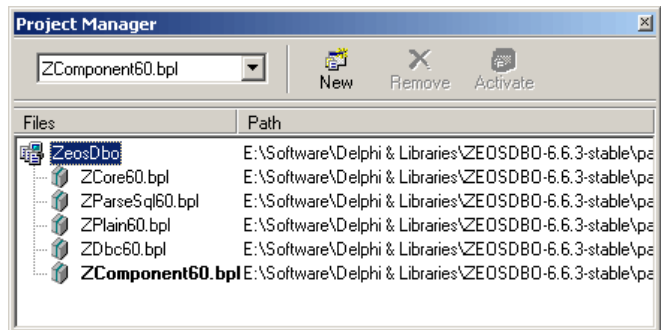
3.3.6 Integrasi Dengan Zeos

Seperti yang sudah dijelaskan sebelumnya, (pengertian zeos, Bab II), akan dijelaskan cara integrasi dengan zeos, yang mana zeos

merupakan komponen pendukung yang menghubungkan MySQL dengan Delphi.

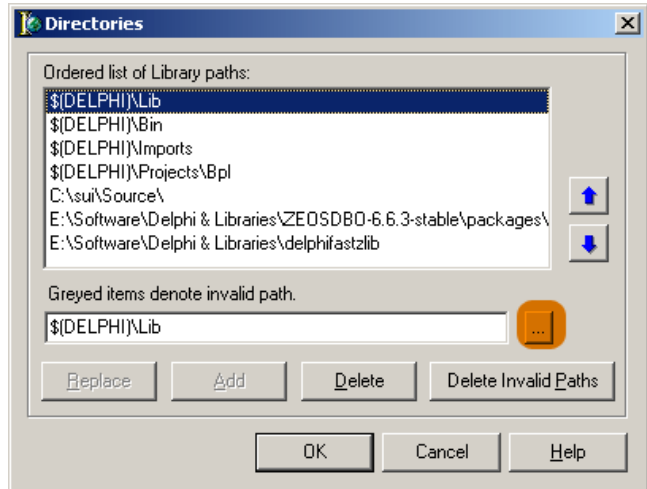
1. Instalasi Zeos

- a. Ekstrak file ZEOSDBO-6.6.3-stable.zip hasil download ke sebuah folder.
- b. Masuk ke dalam folder tersebut. Cek apakah folder Packages\DelphiX\Build sudah ada. DelphiX berarti Delphi 6.0, Delphi 7.0 atau versi lainnya sesuai versi Delphi yang akan digunakan. Karena disini menggunakan Delphi 6.0, maka folder yang harus di cek adalah Packages\Delphi6\Build.
- c. Jalankan Delphi. Dari Delphi, buka file ZeosDbo.bpg dari folder Packages\DelphiX. Kompile semua project dari menu Project>Compile All. Berikut Binary Package yang harus terkompilasi:
 - ZCore60.bpl
 - ZParseSql60.bpl
 - ZPlain60.bpl
 - ZDbc60.bpl
 - ZComponent60.bpl



Gambar 3.23. Project Manager Zeos

- d. Jika sudah, tutup Project Manager. Tambahkan path folder C:\ZEOSDBO-6.6.3-stable\Packages\DelphiX\Build ke Delphi Library Path. Caranya masuk ke menu Tools>Environment Options. Pada dialog yang muncul, klik tab Library, dan klik tombol elipsis di samping kanan combobox Library path. Dialog Directories akan muncul.



Gambar 3.24. Gambar Direktori Zeos

- e. Klik tombol elipsis yang dilingkari, pilih folder C:\ZEOSDBO-6.6.3-stable\Packages\DelphiX\Build di mana diletakkan Zeos DBO hasil kompilasi. Tombol Add akan menjadi aktif. Klik Tombol Add, kemudian klik OK. Kembali ke dialog Environment Options. Klik OK lagi.
- f. Install komponen-komponen akses data di Component Palette. Buka file ZComponentDesign.dpk dalam folder C:\ZEOSDBO-6.6.3-stable\Packages\DelphiX. Jika file ini tidak ditemukan, pilih saja file ZComponent.dpk. Klik tombol Install. Jika tak ada error melintang, Zeos DBO telah terinstall di Component Palette Delphi.

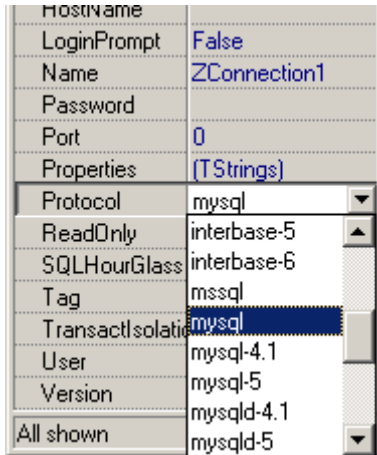
2. Sistem Kendali Parkir dengan Zeos Database

- a. Tambahkan sebuah TZConnection, TZTable dari page Zeos Access ke form.
- b. Tambahkan sebuah TZConnection, TZTable dari page Zeos Access ke form.
- c. Tambahkan juga sebuah TDataSource dari page Data Access. Juga sebuah TDBGrid dan sebuah TDBNavigator, keduanya dari page Data Controls.



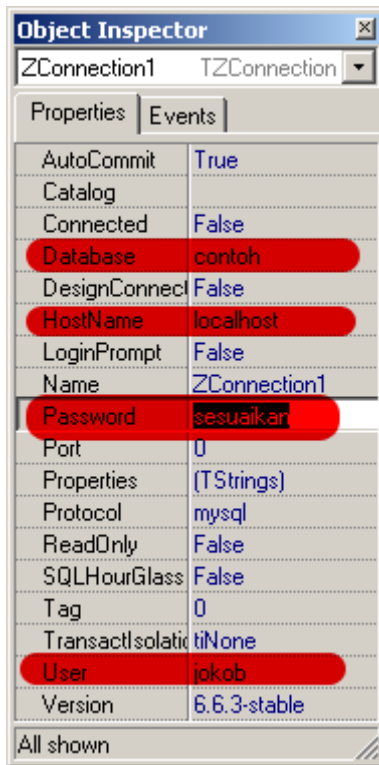
Gambar 3.25. Komponen Zeos di Toolbar Delphi

- d. Atur setting koneksi TZConnection. Protocol diset ke MySQL. Ada beberapa pilihan, tapi gunakan MySQL.



Gambar 3.26. Koneksi TZConnection

- e. Atur juga properti TZConnection yang lain seperti gambar berikut:



Gambar 3.27. Setting Properti TZConnection

- f. Jika sudah, set properti Connected ke True. Kini atur proprerti connection milik TZTable ke ZConnection1. Atur properti TableName menjadi pelanggan. Set properti Active ke True. Lanjutkan dengan mengeset properti DataSet milik DataSource1 menjadi ZTable1. Terakhir, atur properti DataSource milik DBNavigator1 dan DBGrid1 menjadi DataSource1.
- g. Jika tidak ada error, aplikasi telah terkoneksi dengan MySQL server via Zeos DBO.