BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1. Implementasi

Tahap implementasi sistem merupakan tahap lanjutan dari perancangan sistem yang bersifat objektif terhadap cara pandang pada suatu sistem informasi atau perangkat lunak, hal ini bertujuan untuk mempermudah dan mempersiapkan supaya sistem informasi yang dibangun tersebut siap untuk dioperasikan.

Proses implementasi dilakukan sebagai akhir dari desain sistem informasi penyewaan bus dengan konsep *client-server* pada marissa holiday tour and travel cianjur dibangun dengan menggunakan bahasa pemrograman Visulal Basic 6.0. Basis data yang digunakan adalah SQL Server 2005, Implementasi dan pengujian dilakukan di perangkat keras PC (*Personal Computer*) dengan sistem operasi Micrososft Windows XP.

Salah satu hal yang harus diperhatikan dalam mengembangkan suatu sistem manual menjadi sistem yang terkomputerisasi adalah tahap implementasi sistem. Penerapan program aplikasi penyewaan bus yang terbiasa menggunakan sistem pengolahan data secara manual membutuhkan pendekatan yang baik seperti diadakannya pelatihan pada petugas yang akan bertanggung jawab atas kegiatan yang berhubungan dengan program aplikasi penyewaan bus ini.

5.1.1. Batasan Implementasi

Dalam mengimplementasikan Sistem Informasi penyewaan bus di Marissa holiday tour and travel cianjur ini.ada beberapa hal yang perlu diperhatikan. diantaranya adalah batasan-batasan implementasi yang meliputi :

1. Basis Data yang digunakan dalam pengimplementasian ini adalah basis data SQL Serever 2005 sebagai basisdata utama. Pengaksesan basisdata di *server* dilakukan langsung dari sistem aplikasi secara modular (agar tidak perlu mengubah keseluruhan perangkat lunak yang dibangun).
2. Tidak semua kebutuhan sistem informasi penyewaan bus ini terpenuhi. Tetapi hanya pada subsistem, sistem hanya dibuat untuk pengolahan data bus,transaksi bus,penjadwalan sopir dan pemandu, perawatan bus.

5.1.2. Implementasi Perangkat Lunak

Perangkat lunak (*Software*) digunakan untuk mendukung dari sistem operasi (SO) dan aplikasi *database*. Adapun perangkat lunak pengembangan yang digunakan adalah sebagai berikut :

a. Komputer *Server*

Server adalah komputer yang mempunyai kemampuan yang lebih dari komputer *client*, dimana didalamnya tersimpan program dan data-data yang akan didistribusikan.

1. *Software* Sistem Operasi Windows XP.
2. *Software Database* SQL Server 2005
3. *Software Programming* Visual Basic 6.0

b. Komputer *Client*

Client adalah komputer yang didalamnya tersimpan data-data yang dapat digunakan secara perorangan, mengumpulkan dan menampilkan data, serta menyimpan data ke *server*.

1. *Software* Sistem Operasi Windows XP.
2. *Software Programming* Visual Basic 6.0

5.1.3. Implementasi Perangkat Keras

Kebutuhan perangkat keras meliputi pengadaan perangkat keras yang menunjang terhadap berjalannya perangkat lunak. Adapun kebutuhan minimal perangkat keras yang digunakan adalah sebagai berikut :

a. Komputer *Server*
   1. *Harddisk* 80 Giga Byte
   2. RAM minimal 256 Megabyte
   3. Processor minimal Pentium IV
   4. Monitor berwarna minimal 14 inc
   5. NIC Gigafast Ethernet 10/100 Mbps
   6. Mouse dan keyboard Standard
   7. Printer
   8. Hub

b. Komputer *Client*
   1. *Harddisk* 40 Giga Byte
   2. RAM minimal 128 Megabyte
   3. Proccesor minimal Pentium III

4. Monitor berwarna minimal 14 inc
5. NIC Gigafast Ethernet 10/100 Mbps
6. Mouse dan keyboard Standard

5.1.4. Implementasi Basis Data (Sintaks SQL Server 2005)

Pembuatan basis data dilakukan dengan mengunakan bahasa SQL Server 2005, implementasi basis datanya adalah sebagai berikut

```
CREATE LOGIN DATABASE [SQL]
IF NOT EXISTS(SELECT name FROM master.dbo.syslogins WHERE name =
'Marissa')
BEGIN
      CREATE LOGIN [Marissa] WITH PASSWORD='Marissa',
      DEFAULT_DATABASE=[MASTER], DEFAULT_LANGUAGE=[us_english],
      CHECK_EXPIRATION=ON, CHECK_POLICY=ON;
      EXEC sys.sp_addsrvrolemember @loginame = N'Marissa',
      @rolename = N'sysadmin';
      EXEC sys.sp_addsrvrolemember @loginame = N'Marissa',
      @rolename = N'securityadmin';
      EXEC sys.sp_addsrvrolemember @loginame = N'Marissa',
      @rolename = N'serveradmin';
      EXEC sys.sp_addsrvrolemember @loginame = N'Marissa',
      @rolename = N'setupadmin';
      EXEC sys.sp_addsrvrolemember @loginame = N'Marissa',
      @rolename = N'processadmin';
      EXEC sys.sp_addsrvrolemember @loginame = N'Marissa',
      @rolename = N'diskadmin';
      EXEC sys.sp_addsrvrolemember @loginame = N'Marissa',
      @rolename = N'dbcreator';
      EXEC sys.sp_addsrvrolemember @loginame = N'Marissa',
      @rolename = N'bulkadmin';
      ALTER LOGIN [Marissa] ENABLE;
      SELECT name FROM master.dbo.syslogins WHERE name =
      'Marissa';
END
ELSE
      SELECT name FROM master.dbo.syslogins;

CREATE DATABASE [SQL]
IF NOT EXISTS(SELECT name FROM sys.databases WHERE name =
'marissa_db')
BEGIN
      CREATE DATABASE [marissa_db];
      SELECT name FROM sys.databases WHERE name = 'marissa_db';
END;
```

```
ELSE
      SELECT name FROM sys.databases;
```

CREATE TABLE AND RELATION [SQL]

```
IF NOT EXISTS (SELECT name FROM [marissa_db].[sys].[Tables] WHERE
name = 'userpass')
BEGIN
      CREATE TABLE [marissa_db].[dbo].[bus]
            ([no_polisi] [nvarchar](8) NOT NULL,
            [kode_bus] [smallint] NULL,
            [status_ketersediaan] [nvarchar](25) NULL,
            CONSTRAINT [bus$PrimaryKey] PRIMARY KEY CLUSTERED
      ([no_polisi] ASC));

      CREATE TABLE [marissa_db].[dbo].[detail_akomodasi]
            ([kode_akomodasi] [smallint] NOT NULL,
            [tujuan] [nvarchar](40) NULL,
            [konsumsi] [money] NULL DEFAULT ((0)),
            [asuransi] [money] NULL DEFAULT ((0)),
            [tiket_wisata] [money] NULL DEFAULT ((0)),
            [akomodasi_plus] [money] NULL DEFAULT ((0)),
            [diskon] [float] NULL DEFAULT ((0)),
            [max_sewa] [smallint] NULL DEFAULT ((0)),
            [ketersediaan_paket] [bit] NULL DEFAULT ((0)),
            [rincian] [nvarchar](max) NULL,
            CONSTRAINT [detail_akomodasi$PrimaryKey] PRIMARY KEY
      CLUSTERED ([kode_akomodasi] ASC));

      CREATE TABLE [marissa_db].[dbo].[detail_bus]
            ([kode_bus] [smallint] NOT NULL,
            [jumlah_kursi] [smallint] NULL,
            [sewa_pokok] [money] NULL,
            CONSTRAINT [detail_bus$PrimaryKey] PRIMARY KEY
      CLUSTERED ([kode_bus] ASC));

      CREATE TABLE [marissa_db].[dbo].[detail_transaksi]
            ([no_transaksi] [nvarchar](6) NOT NULL,
            [no_polisi] [nvarchar](8) NULL,
            [kapasitas_bus] [smallint] NULL,
            [kode_akomodasi] [smallint] NULL,
            [tujuan_perjalanan] [nvarchar](225) NULL,
            [lama_sewa] [smallint] NULL,
            [paket] [bit] NULL DEFAULT ((0)),
            [tot_biaya_akomodasi] [money] NULL,
            [tot_biaya_sewa_bus] [money] NULL,
            [tot_biaya_perjalanan] [money] NULL);

      CREATE TABLE [marissa_db].[dbo].[jabatan]
            ([kode_jabatan] [nvarchar](2) NOT NULL,
            [nama_jabatan] [nvarchar](25) NULL,
            [gaji_pokok] [money] NULL,
            CONSTRAINT [jabatan$PrimaryKey] PRIMARY KEY CLUSTERED
      ([kode_jabatan] ASC));

      CREATE TABLE [marissa_db].[dbo].[pegawai]
            ([id_pegawai] [nvarchar](9) NOT NULL,
```

```
      [nama] [nvarchar](25) NULL,
      [alamat] [nvarchar](50) NULL,
      [no_telepon] [nvarchar](12) NULL,
      [kode_jabatan] [nvarchar](2) NULL,
      [hari_libur] [nvarchar](25) NULL,
      [status_kerja] [nvarchar](25) NULL,
      CONSTRAINT [pegawai$PrimaryKey] PRIMARY KEY CLUSTERED
([id_pegawai] ASC));

CREATE TABLE [marissa_db].[dbo].[perawatan]
      ([kode_transaksi_perawatan] [int] NOT NULL,
      [no_polisi] [nvarchar](8) NULL,
      [tgl_perawatan] [datetime] NULL,
      [biaya_perawatan] [money] NULL,
      [detail_kerusakan] [nvarchar](25) NULL,
      CONSTRAINT [perawatan$PrimaryKey] PRIMARY KEY
CLUSTERED ([kode_transaksi_perawatan] ASC));

CREATE TABLE [marissa_db].[dbo].[surat_jalan]
      ([no_surat_jalan] [nvarchar](5) NOT NULL,
      [id_pegawai] [nvarchar](9) NULL,
      [no_transaksi] [nvarchar](6) NULL,
      [selesai] [bit] NULL DEFAULT ((0)),
      CONSTRAINT [surat_jalan$PrimaryKey] PRIMARY KEY
CLUSTERED ([no_surat_jalan] ASC));

CREATE TABLE [marissa_db].[dbo].[transaksi]
      ([no_transaksi] [nvarchar](6) NOT NULL,
      [nama_pemesan] [nvarchar](25) NULL,
      [alamat_pemesan] [nvarchar](50) NULL,
      [no_tlp_pemesan] [nvarchar](12) NULL,
      [tgl_berangkat] [datetime] NULL,
      [tgl_kembali] [datetime] NULL,
      [jam_berangkat] [nvarchar](5) NULL,
      [uang_bayar] [money] NULL,
      [uang_batal] [money] NULL DEFAULT ((0)),
      CONSTRAINT [transaksi$PrimaryKey] PRIMARY KEY
CLUSTERED ([no_transaksi] ASC));

CREATE TABLE [marissa_db].[dbo].[userpass]
      ([user_name] [varchar](50) NOT NULL,
      [password] [varchar](50) NOT NULL,
      [id_pegawai] [nvarchar](9) NULL,
      [tipe_user] [nvarchar](25) NULL,
      [status_aktif] [bit] NULL DEFAULT ((0)),
      CONSTRAINT [PK_user] PRIMARY KEY CLUSTERED
([user_name] ASC));

ALTER TABLE [marissa_db].[dbo].[bus]
      WITH CHECK ADD CONSTRAINT [bus$detail_busbus] FOREIGN
KEY([kode_bus])
      REFERENCES [marissa_db].[dbo].[detail_bus]
([kode_bus]);

ALTER TABLE [marissa_db].[dbo].[bus] CHECK CONSTRAINT
[bus$detail_busbus];
```

```
ALTER TABLE [marissa_db].[dbo].[detail_transaksi]
      WITH CHECK ADD CONSTRAINT
[detail_transaksi$no_transaksidetail_transaksi] FOREIGN
KEY([no_transaksi])
      REFERENCES [marissa_db].[dbo].[transaksi]
([no_transaksi]);

ALTER TABLE [marissa_db].[dbo].[detail_transaksi] CHECK
CONSTRAINT [detail_transaksi$no_transaksidetail_transaksi];

ALTER TABLE [marissa_db].[dbo].[detail_transaksi]
      WITH CHECK ADD CONSTRAINT
[detail_transaksi$busdetail_transaksi] FOREIGN
KEY([no_polisi])
      REFERENCES [marissa_db].[dbo].[bus] ([no_polisi]);

ALTER TABLE [marissa_db].[dbo].[detail_transaksi] CHECK
CONSTRAINT [detail_transaksi$busdetail_transaksi];

ALTER TABLE [marissa_db].[dbo].[detail_transaksi]
      WITH CHECK ADD CONSTRAINT
[detail_transaksi$detail_akomodasidetail_transaksi] FOREIGN
KEY([kode_akomodasi])
      REFERENCES [marissa_db].[dbo].[detail_akomodasi]
([kode_akomodasi]);

ALTER TABLE [marissa_db].[dbo].[detail_transaksi] CHECK
CONSTRAINT
[detail_transaksi$detail_akomodasidetail_transaksi];

ALTER TABLE [marissa_db].[dbo].[pegawai]
      WITH CHECK ADD CONSTRAINT [pegawai$jabatanpegawai]
FOREIGN KEY([kode_jabatan])
      REFERENCES [marissa_db].[dbo].[jabatan]
([kode_jabatan]);

ALTER TABLE [marissa_db].[dbo].[pegawai] CHECK CONSTRAINT
[pegawai$jabatanpegawai];

ALTER TABLE [marissa_db].[dbo].[perawatan]
      WITH CHECK ADD CONSTRAINT [perawatan$busperawatan]
FOREIGN KEY([no_polisi])
      REFERENCES [marissa_db].[dbo].[bus] ([no_polisi]);

ALTER TABLE [marissa_db].[dbo].[perawatan] CHECK CONSTRAINT
[perawatan$busperawatan];

ALTER TABLE [marissa_db].[dbo].[surat_jalan]
      WITH CHECK ADD CONSTRAINT [surat_jalan$pegawaisurat
jalan] FOREIGN KEY([id_pegawai])
      REFERENCES [marissa_db].[dbo].[pegawai]
([id_pegawai]);

ALTER TABLE [marissa_db].[dbo].[surat_jalan] CHECK
CONSTRAINT [surat_jalan$pegawaisurat jalan];
```

```
      ALTER TABLE [marissa_db].[dbo].[surat_jalan]
            WITH CHECK ADD CONSTRAINT [surat_jalan$transaksisurat
      jalan] FOREIGN KEY([no_transaksi])
            REFERENCES [marissa_db].[dbo].[transaksi]
      ([no_transaksi]);

      ALTER TABLE [marissa_db].[dbo].[surat_jalan] CHECK
      CONSTRAINT [surat_jalan$transaksisurat jalan];
      SELECT name FROM [marissa_db].[sys].[Tables] WHERE name =
      'userpass';
END;
ELSE
      SELECT name FROM [marissa_db].[sys].[Tables];
```

CREATE TRIGGER [SQL]

```
CREATE TRIGGER updateStatusPegawai1 ON
[marissa_db].[dbo].[surat_jalan]
AFTER INSERT AS
BEGIN
      UPDATE [marissa_db].[dbo].[pegawai]
      SET [status_kerja] = 'Aktif'
      WHERE [id_pegawai]= (SELECT [id_pegawai] FROM Inserted);
 END;

CREATE TRIGGER updateStatusPegawai2 ON
[marissa_db].[dbo].[surat_jalan]
FOR DELETE AS
BEGIN
      UPDATE [marissa_db].[dbo].[pegawai]
      SET [status_kerja] = 'Non-Aktif'
      WHERE [id_pegawai]= (SELECT [id_pegawai] FROM Deleted);
END;

CREATE TRIGGER updateStatusPegawai3 ON
[marissa_db].[dbo].[surat_jalan]
FOR UPDATE AS
BEGIN
      UPDATE [marissa_db].[dbo].[pegawai]
      SET [status_kerja] = 'Non-Aktif'
      WHERE [id_pegawai]= (SELECT [id_pegawai] FROM Inserted);
END;

CREATE TRIGGER updateStatusBus ON [marissa_db].[dbo].[perawatan]
FOR UPDATE AS
BEGIN
      UPDATE [marissa_db].[dbo].[bus]
      SET [status_ketersediaan] = 'Perawatan'
      WHERE [no_polisi]= (SELECT [no_polisi] FROM Inserted);
END;
```

CREATE USER [SQL]

```
IF NOT EXISTS (SELECT name FROM [marissa_db].[dbo].[sysusers]
WHERE name = 'Marissa')
BEGIN
```

```
      CREATE USER [Marissa] FOR LOGIN [Marissa] WITH
      DEFAULT_SCHEMA=[marissa_db.dbo];
      SELECT name FROM [marissa_db].[dbo].[sysusers] WHERE name =
      'Marissa';
END;
ELSE
      SELECT name FROM marissa_db.dbo.sysusers;
```

5.1.5. Implementasi Antar Muka

Implementasi antarmuka merupakan tahapan dalam memenuhi kebutuhan pengguna (*user*) dalam berinteraksi dengan komputer. Fasilitas antar muka yang baik sangat membantu pemakai dalam memahami proses yang sedang dilakukan oleh sistem tersebut dan dapat meningkatkan kinerja sistem.

1. Implementasi Menu Utama

Berikut implementasi menu utama dari sistem informasi penyewaan bus, dapat dilihat pada tabel dibawah ini :

Tabel 5.1 Implementasi Menu Utama

Menu	Deskripsi
File	Menu file untuk menangani pengaturan user, logout, dan *exit*.
Transaksi	Menu yang berfungsi untuk menginput proses transaksi penyewaan bus dan menghasilkan output yaitu struk penyewaan dan surat jalan bagi sopir dan pemandu.dan mengupdate jadwal sopir dan pemandu ketika mereka selesai atau pulang bertugas.
Master	Menu yang berfungsi untuk mengelola data bus, karyawan,paket wisata dan pengguna /*user control*
Cetak	Menu cetak berfungsi untuk menampilkan dan mencetak laporan penyewaan bus, batal pesan,laporan jadwal sopir dan pemandu, dan laporan perawatan bus
bantuan	Menu pbantuan berfungsi untuk menampilkan informasi paket wisata info jadwal sopir dan bus,info bus dan tentang perusahaan yaitu marissa holiday tour and travel cianjur

2. Implementasi Menu File

Berikut implementasi menu file dari sistem informasi penyewaan bus dapat dilihat pada tabel dibawah ini :

Tabel 5.2 Implementasi Menu File

Menu	Deskripsi
Ubah kata sandi	File program untuk menangani pengaturan kata sandi sebagai akses masuk ke aplikasi
Logout	File program untuk menangani user yang akan keluar dari sistem tanpa menutup aplikasi
keluar	File program untuk menangani user yang akan keluar dari sistem dan menutup aplikasi.

3. Implementasi Menu Transaksi

Berikut implementasi menu transaksi dari sistem informasi informasi penyewaan bus, dapat dilihat pada tabel dibawah ini :

Tabel 5.3 Implementasi Menu Transaksi

Menu	Deskripsi
Transaksi	File program untuk mengiput data yang didapat dari pemesan sebagai informasi untuk melakukan penyewaan dan timbal balik ke pemesan
Konfirmasi perjalanan	File konfirmasi perjalanan ini untuk menginput no surat jalan yang didapat dari sopir dan pemandu untuk mengupdate jadwal mereka yang berguna sebagai informasi ketersedian sopir dan pemandu.

4. Implementasi Menu Master

Berikut implementasi menu Master dari sistem informasi informasi penyewaan bus, dapat dilihat pada tabel dibawah ini :

Tabel 5.4 Implementasi Menu Master

Menu	Deskripsi
Paket wisata	File program untuk mengelola data daftar paket wisata yang tersedia di marissa holday tour and travel seperti tujuan wisata harga paket wisata dan lain-lainnya.
Bus	File program untuk mengelola data bus yang tersedia di marissa holday tour and travel seperti no polisi bus,kode bus,sewa pokok bus,dan lain-lainnya.
Pegawai	File program untuk mengelola data pegawai yang tersedia di marissa holday tour and travel seperti no id pegawai dan identitas pegawai,mengisi data baru pegawai,mengedit informasi pegawai dan lain-lainnya.
pengguna	File program untuk mengelola pengguna aplikasi sistem yang dibangun.

5. Implementasi Menu Cetak

Berikut implementasi menu Cetak dari sistem informasi informasi penyewaan bus, dapat dilihat pada tabel dibawah ini :

Tabel 5.5 Implementasi Menu Cetak

Menu	Deskripsi
Laporan penyewaan	File program untuk mencetak laporan penyewaan bus yang sudah tersimpan di database dan prose cetak laporan penyewaan dilakukan secara berkala yaitu setiap satu bulan sekali

Laporan batal pesan	File program untuk mencetak laporan batal penyewaan bus yang sudah tersimpan di database dan proses cetak laporan batal penyewaan dilakukan secara berkala yaitu setiap satu bulan sekali
Laporan perawatan bus	File program untuk mencetak laporan perawatan bus yang sudah tersimpan di database dan proses cetak laporan perawatan bus dilakukan secara berkala yaitu setiap satu bulan sekali
Laporan jadwal sopir dan pemandu	File program untuk mencetak laporan jadwal sopir dan pemandu yang sudah tersimpan di database dan proses cetak laporan jadwal sopir dan pemandu dilakukan secara berkala yaitu setiap satu bulan sekali

6. Implementasi Menu bantuan

Berikut implementasi menu bantuan dari sistem informasi informasi penyewaan bus, dapat dilihat pada tabel dibawah ini :

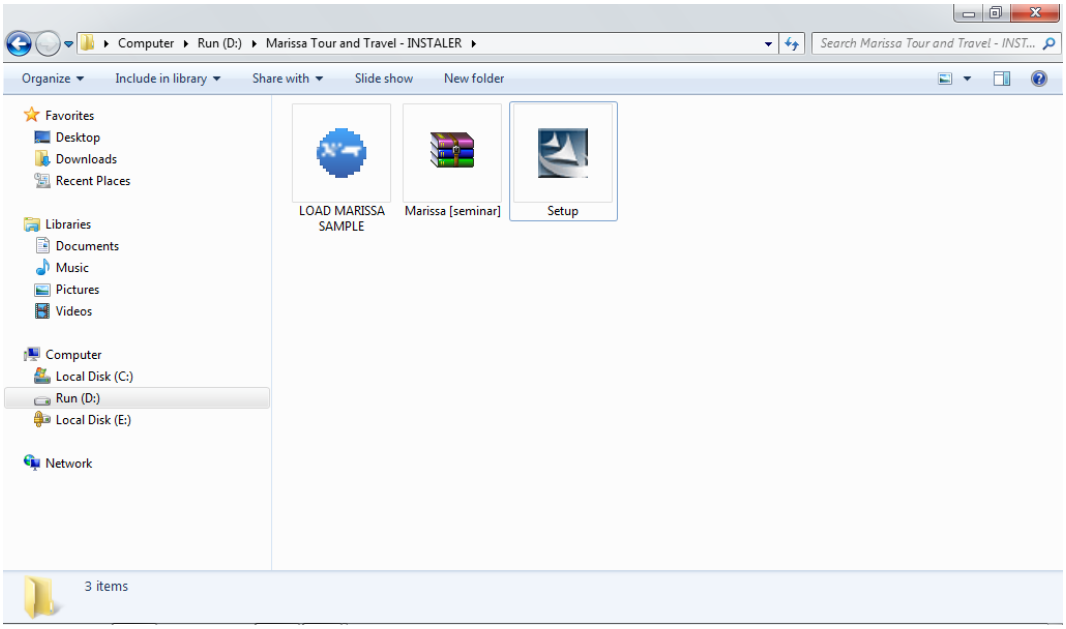
Tabel 5.6 Implementasi Menu bantuan

Menu	Deskripsi
Info paket wisata	File program untuk melihat info paket wisata yang menjadi favorite pemesan dan lain-lainnya.
Info jadwal sopir dan pemandu	File program untuk melihat info sopir dan pemandumulai dari jadwal mereka dan status kerja mereka beserta hari libur kerja mereka.
Info bus	File program untuk melihat data bus status dan kondisi bus sebagai informasi ketersediaan bus
tentang	File program ini sebagia informasi perusahaan berupa visi misi marissa holiday tour and travel

5.1.6. Implementasi Instalasi Program

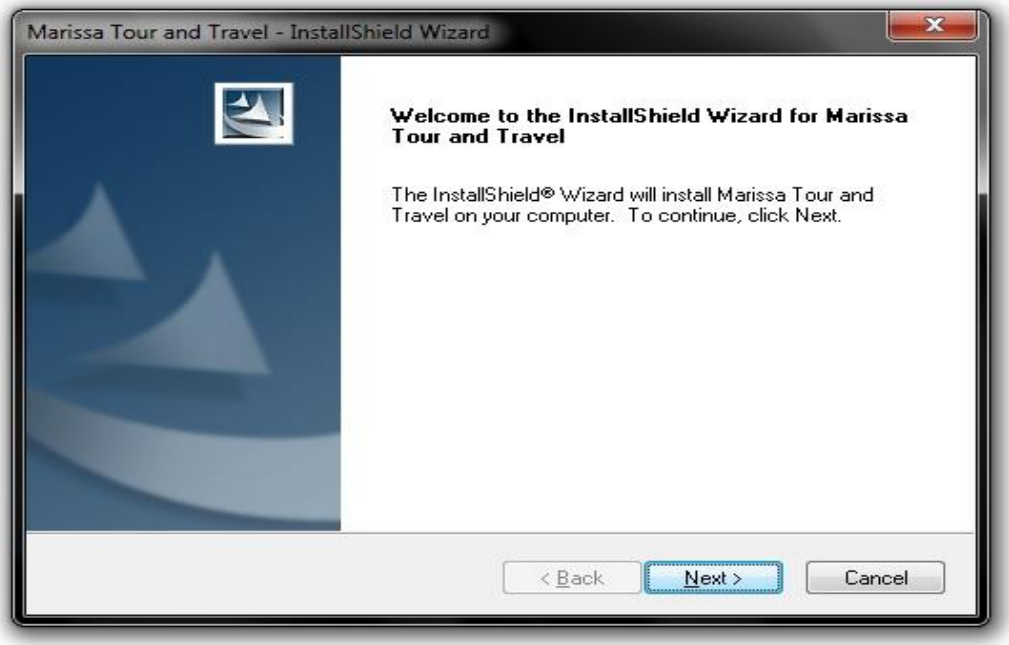
Program aplikasi penyewaan bus yang akan digunakan harus terlebih dahulu di instal pada komputer, berikut akan dijelaskan mengenai cara penginstalan *software* pada komputer.

Cara penginstalan program aplikasi penyewaan bus pada umumnya sama seperti penginstalan *software-software* yang lain. Pertama, *double* klik file setup.exe program aplikasi penyewaan bus seperti pada gambar berikut :



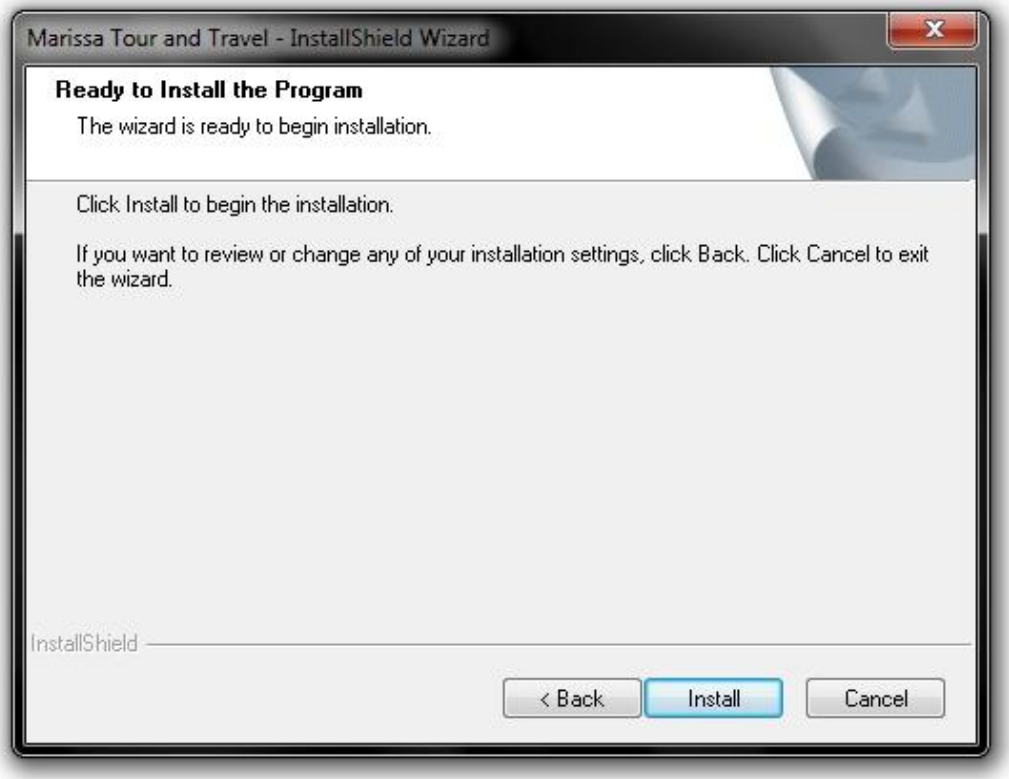
Gambar 5.1 Pemilihan Setup Pada Explorer

Maka beberapa saat kemudian akan muncul tampilan seperti pada gambar berikut :



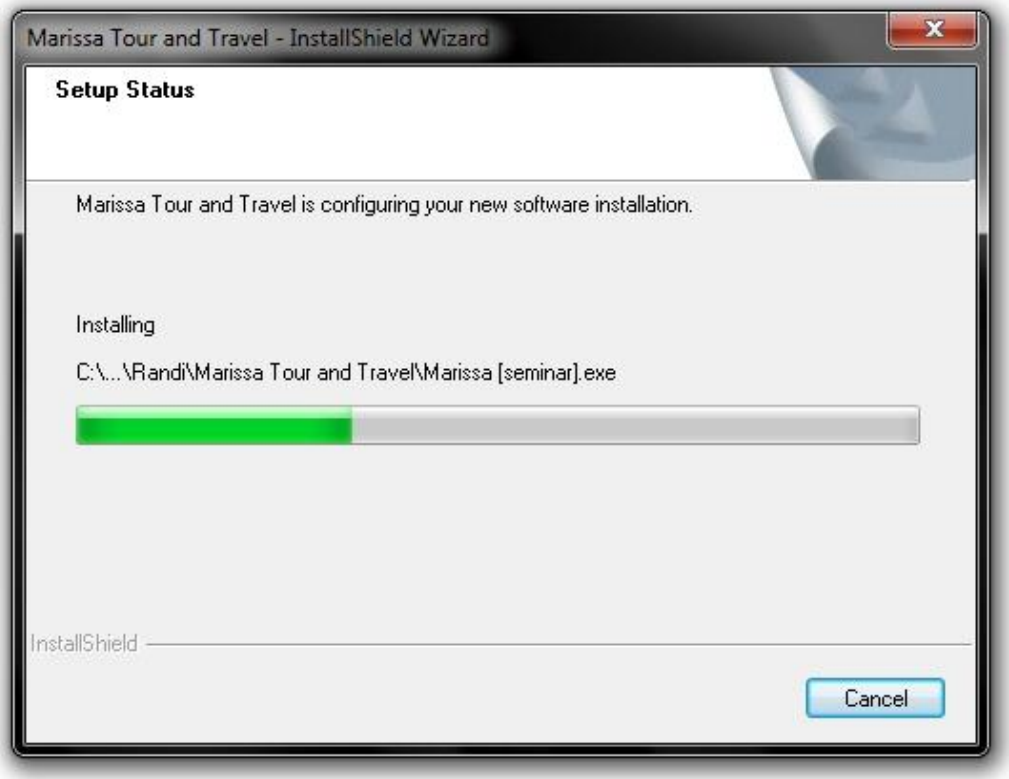
Gambar 5.2 Dialog Awal Instalasi Program

Pada saat proses instalasi program muncul dialog seperti gambar diatas, maka klik tombol *next* untuk melanjutkan ke proses selanjutnya, apabila ingin membatalkan proses instalasi maka klik tombol *cancel*, apabila memilih tombol next maka proses berikutnya akan terlihat seperti gambar berikut:



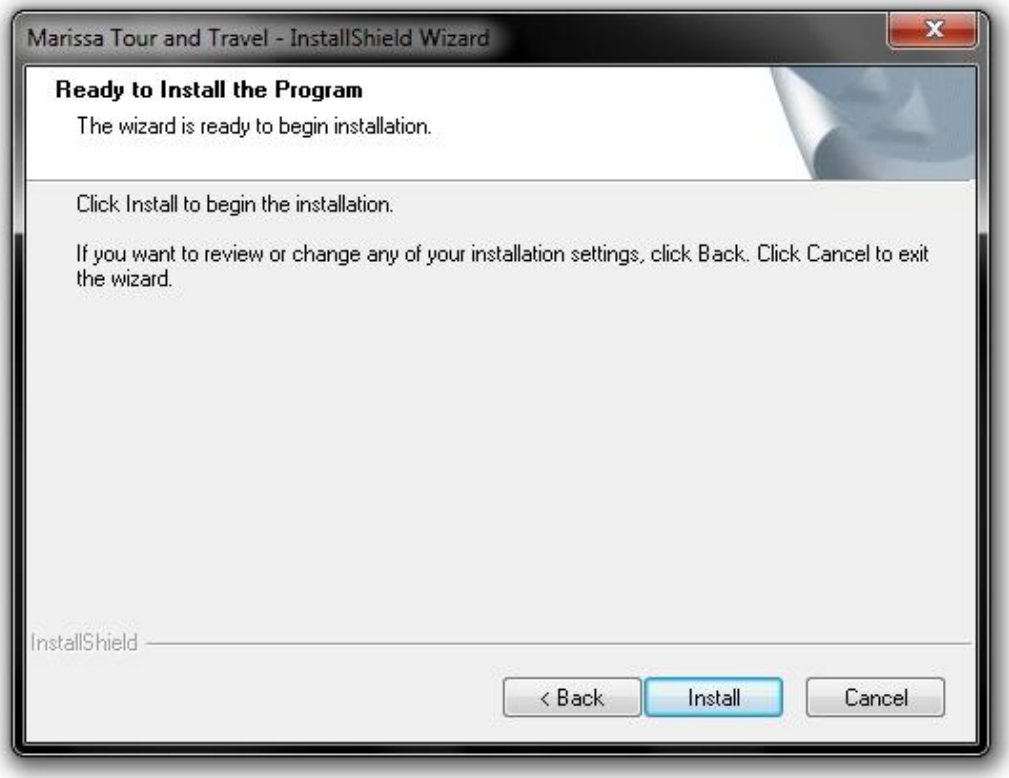
Gambar 5.3 Memilih Drive Dimana Program akan Di Instal

Gambar diatas menunjukan dimana program akan di instal, pilih browse untuk memilih drive, setelah selesai menentukan dimana program akan di install maka proses selanjutnya adalah menekan tombol *next*, maka proses berikutnya akan terlihat seperti gambar berikut ini:



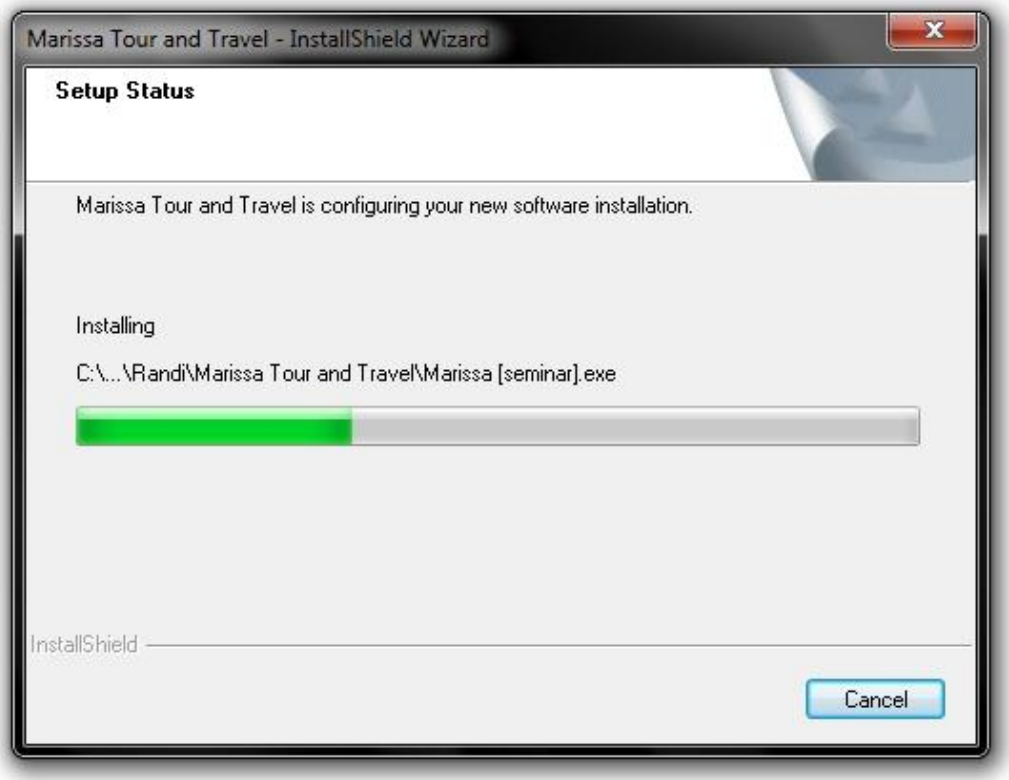
Gambar 5.4 Dialog Menentukan Nama *Folder* Pada *Start Menu*

Gambar Dialog diatas menunjukan pemberian nama folder program yang nantinya akan bernama marissa ketika menjalankan program dari Start menu, jika ingin kembali tekan tombol *back*, jika ingin membatalkan proses instalasi maka tekan *cancel* dan jika ingin melanjutkan proses instalasi maka tekan tombol *instal.* Setelah menekan tombol *instal* maka proses berikutnya akan terlihat seperti gambar berikut ini:



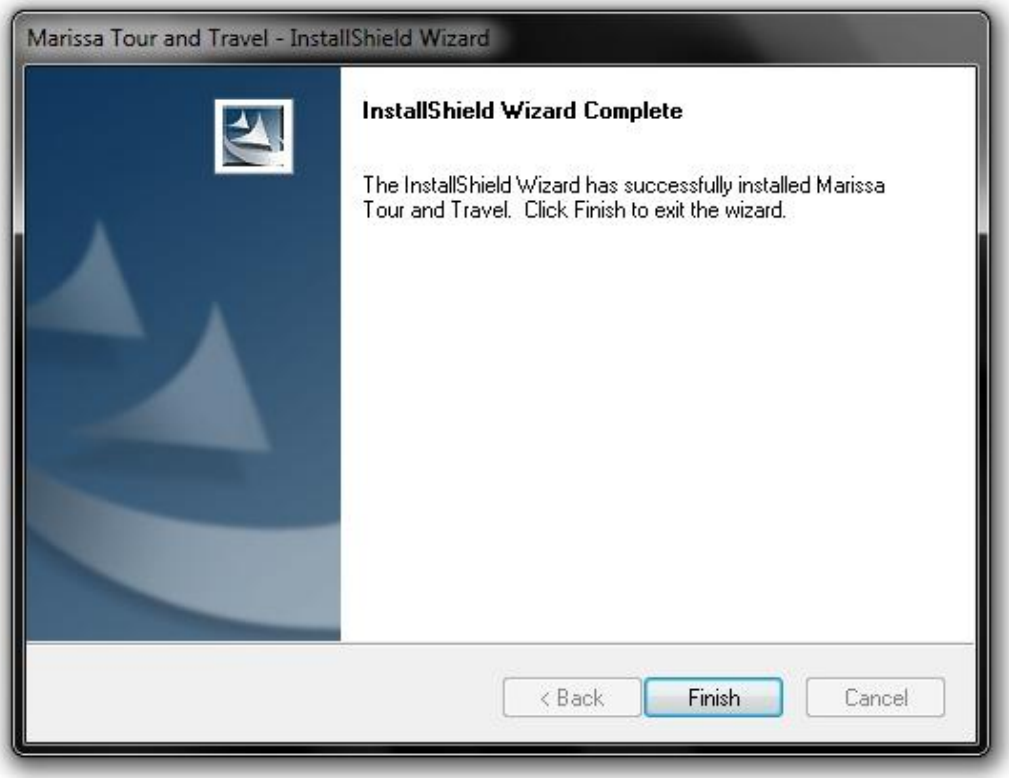
Gambar 5.5 Persiapan *Install Program*

Pada dialog gambar diatas menunjukan bahwa program akan di install pada *drive* C dengan nama *folder* pada *start menu* marissa dan membuat *icon short cut* pada *desktop*, jika semuanya telah sesuai maka tekan tombol *install* untuk melakukan proses instalasi pada komputer. Proses berikutnya akan terlihat seperti gambar dibawah ini:



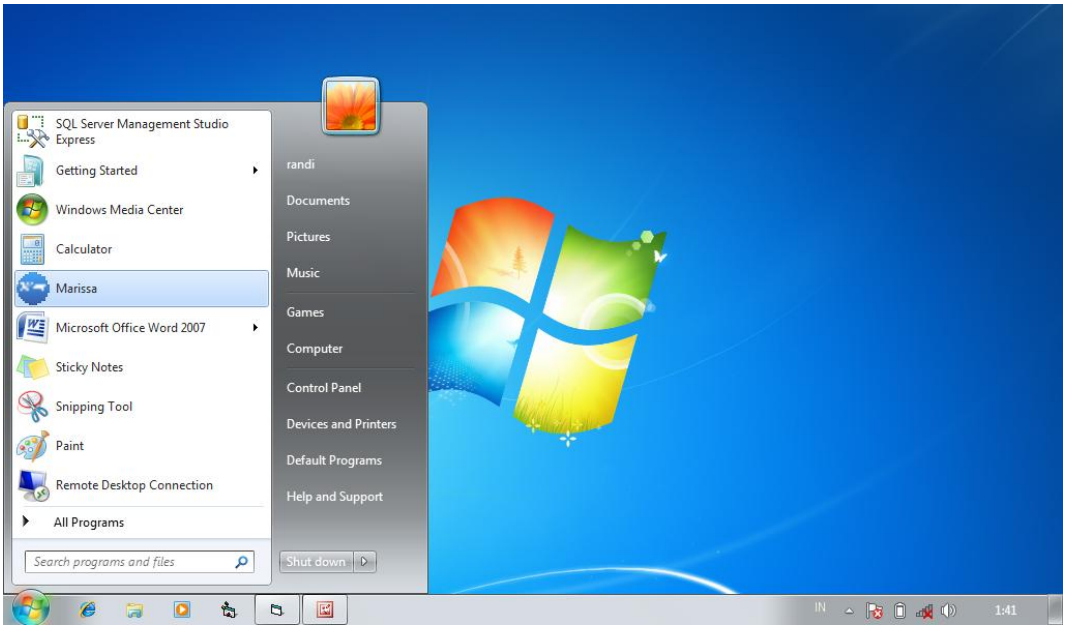
Gambar 5.6 Proses Instalasi Program

Pada dialog gambar diatas menunjukan bahwa program sedang di install pada *drive* C. Proses berikutnya akan terlihat seperti gambar dibawah ini:



Gambar 5.7 Proses Instalasi Program Selesai

Gambar diatas menunjukan bahwa proses instalasi program sudah selesai di install, tekan tombol finish untuk mengakhiri proses instalasi program. Setelah proses instalasi telah selesai, maka secara otomatis program aplikasi penyewaan bus akan tampil pada *start menu window*. Kemudian, untuk menjalankan program aplikasi penyewaan bus klik Launch Aplikasi Perangkat Lunak.exe seperti pada gambar berikut :



Gambar 5.8 Menjalankan Program Aplikasi

5.1.7. Penggunaan Program

Kegiatan ini bertujuan untuk menerangkan secara singkat penggunaan program aplikasi penyewaan bus. Cara penggunaanya adalah sebagai berikut :

1. *Network Config* (Konfigurasi Jaringan)

Form ini berfungsi untuk pengaturan nama server dan nama database yang digunakan. Berikut adalah tampilan *Network Config* (Konfigurasi Jaringan) dari program aplikasi penyewaan bus:



Gambar 5.9 Form pengaturan koneksi SQL

2. Login

Jika ingin masuk ke menu lain maka user harus melakukan login terlebih dahulu dengan menginputkan user name, password Tampilan login sistem ini dimaksudkan untuk mengidentifikasi apakah user berhak atau tidak untuk masuk ke sistem. Untuk lebih jelasnya, implementasi untuk tampilan login pada sistem informasi penyewaan bus ditunjukan pada gambar di bawah ini :



Gambar 5.10. Form login

Jika Password atau nama user yang dimasukan salah, maka akan tampil peringatan seperti pada gambar di bawah ini :



Gambar 5. 11 Pesan Proses Login Gagal

Dan jika data yang di masukan cocok atau benar maka sistem akan langsung masuk kepada menu utama sesuai dengan bagian yang dipilih pengguna dan

didalam aplikasi ini disediakan fasilitas ganti kata sandi berikut ini adalah tampilan kata sandi



Gambar 5. 12 ubah kata sandi

3. Menu Utama Bag. Administrator

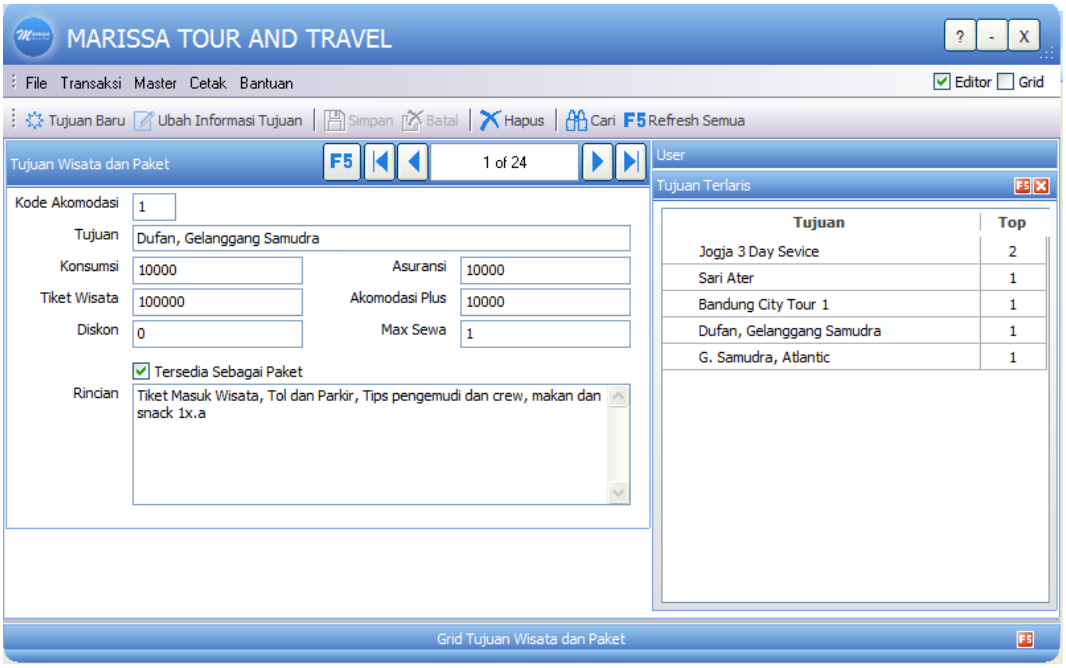
Menu utama bagian administrator merupakan tampilan setelah proses login *user* berhasil dijalankan. Dimana menu utama bagian administrator memiliki fasilitas hak akses penuh terhadap sistem. Untuk keluar dari bagian menu ini cukup mengklik menu file lalu pilih logout. Berikut adalah tampilan menu utama bagian administrator dari program aplikasi penyewaan bus:



Gambar 5.13 Menu Utama Bagian Administrator

4. *Form Master* Paket wisata

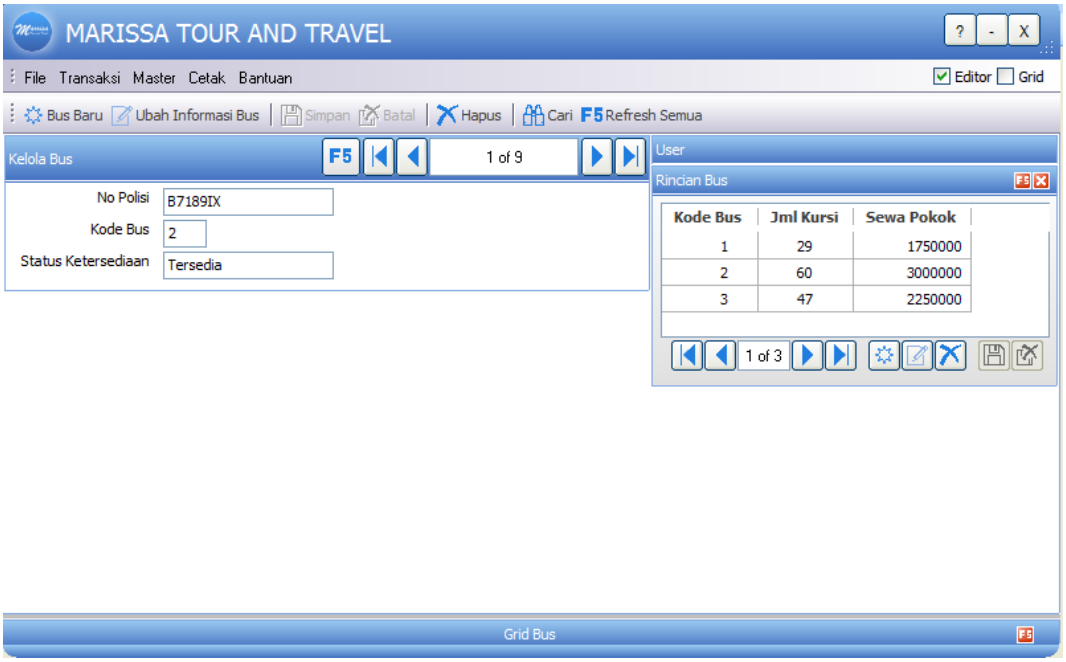
Pada *Form* Paket wisata, digunakan untuk melihat semua data paket wisata, menambah data paket wisata, mengubah data paket wisata dan paket wisata. Untuk menginputkan paket wisata baru klik tombol paket wisata baru dan lengkapi seluruh data yang telah disediakan selanjutnya klik tombol simpan, Untuk mengubah data paket wisata terlebih dahulu pilih data paket wisata yang akan dirubah, setelah itu klik tombol ubah, lalu isi *textbox* sesuai dengan data paket wisata yang akan dirubah. Untuk menghapus data paket wisata, pilih terlebih dahulu data paket wisata mana yang akan dihapus lalu klik tombol hapus. Dan untuk keluar dari form master paket wisata klik tombol silang di kanan atas form. Berikut adalah contoh tampilan form master paket wisata :



Gambar 5.14 Menu Master Paket wisata

5. *Form Master* Bus

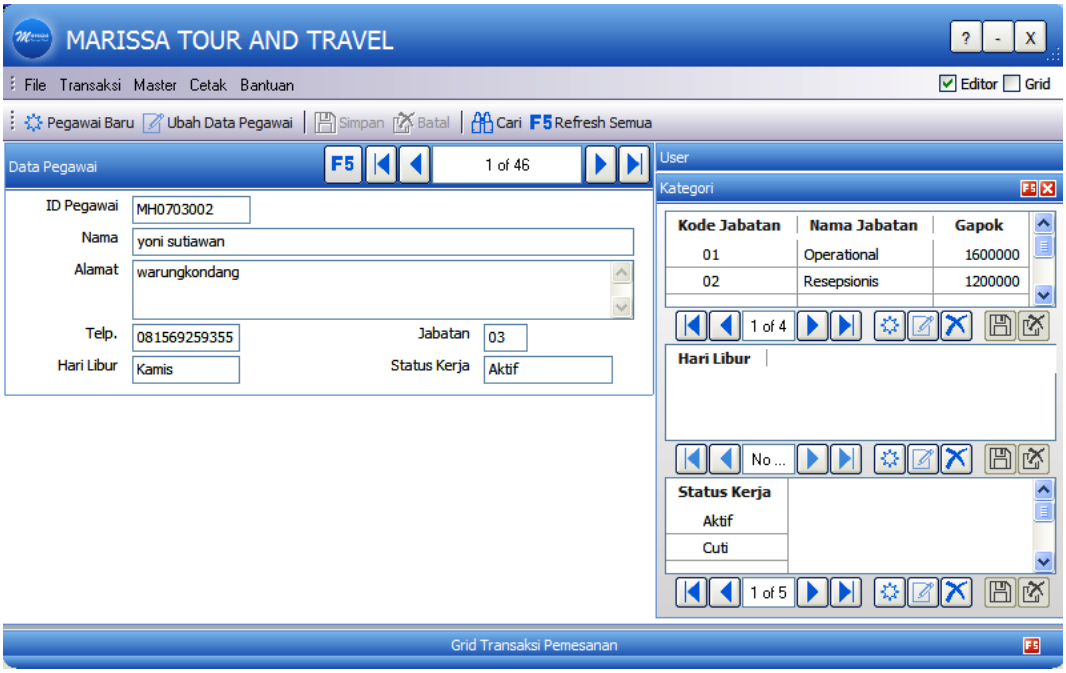
Pada *Form* Bus, digunakan untuk melihat semua data Bus, menambah data Bus, mengubah data Bus, dan Untuk menginputkan Bus baru klik tombol Bus baru dan lengkapi seluruh data yang telah disediakan selanjutnya klik tombol simpan, Untuk mengubah data Bus terlebih dahulu pilih Bus yang akan dirubah, setelah itu klik tombol ubah, lalu isi *textbox* sesuai dengan data Bus yang akan dirubah. Untuk menghapus Bus, pilih terlebih dahulu data Bus mana yang akan dihapus lalu klik tombol hapus. Dan untuk keluar dari form master Bus klik tombol silang di kanan atas form. Berikut adalah contoh tampilan form master Bus:



Gambar 5.15 Menu Master Bus

6. *Form Master* pegawai

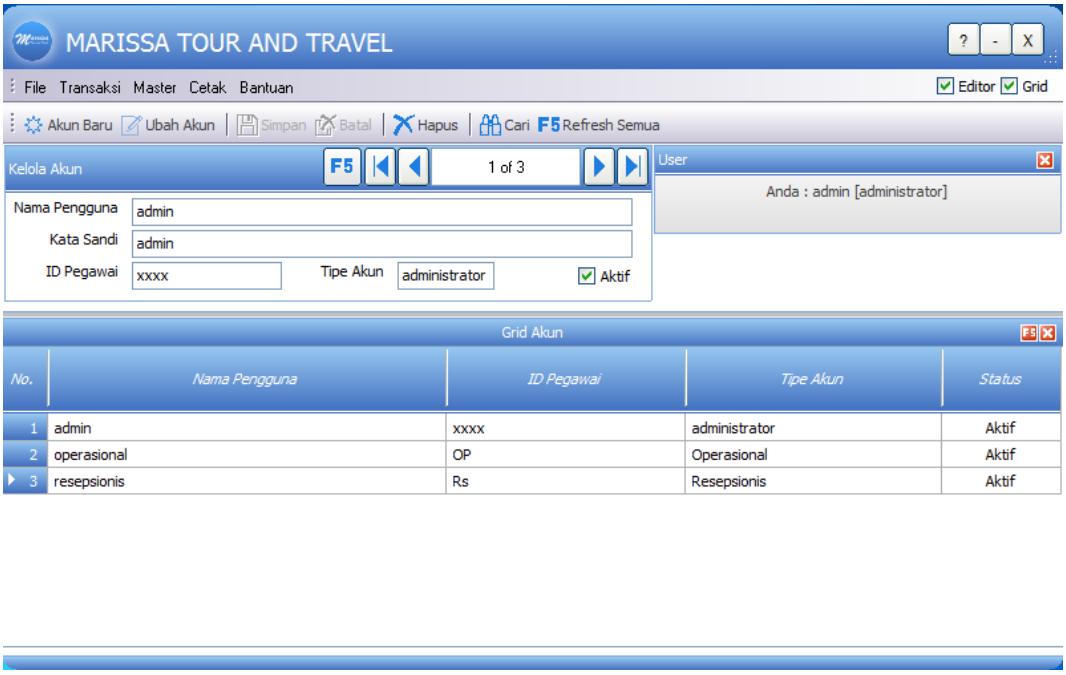
Pada *Form* pegawai, digunakan untuk melihat semua data pegawai, menambah data pegawai, mengubah data pegawai dan Untuk menginputkan pegawai baru klik tombol pegawai baru dan lengkapi seluruh data yang telah disediakan selanjutnya klik tombol simpan, Untuk mengubah data pegawai terlebih dahulu pilih pegawai yang akan dirubah, setelah itu klik tombol ubah, lalu isi *textbox* sesuai dengan data pegawai yang akan dirubah. Untuk menghapus pegawai, pilih terlebih dahulu data pegawai mana yang akan dihapus lalu klik tombol hapus. Dan untuk keluar dari form master pegawai klik tombol silang di kanan atas form. Berikut adalah contoh tampilan form master pegawai:



Gambar 5.16 Menu Master pegawai

7. *Form Master* Pengguna

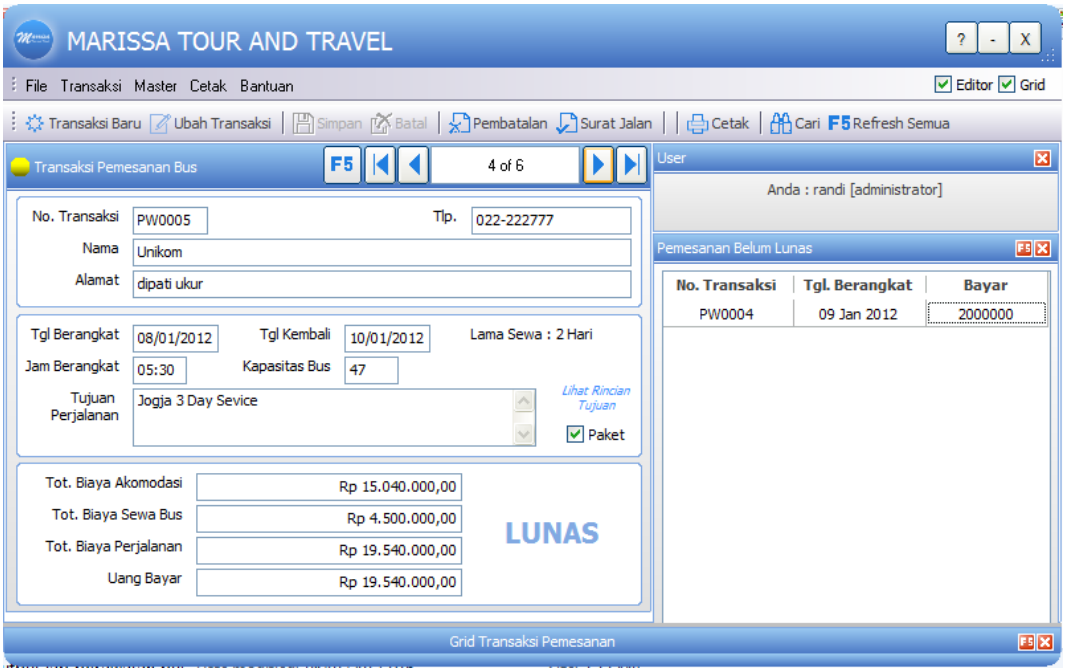
Pada *Form* Pengguna, digunakan untuk melihat semua data Pengguna,yang melakukan aktifitas kedalam aplikasi dan hanya administrator yang bisa mengakses menu master pengguna ini dan Untuk menginputkan Pengguna baru klik tombol akun baru dan lengkapi seluruh data yang telah disediakan selanjutnya klik tombol simpan, Untuk mengubah data Pengguna terlebih dahulu pilih Pengguna yang akan dirubah, setelah itu klik tombol ubah, lalu isi *textbox* sesuai dengan data Pengguna yang akan dirubah. Untuk menghapus Pengguna, pilih terlebih dahulu data Pengguna mana yang akan dihapus lalu klik tombol hapus. Dan untuk keluar dari form master Pengguna klik tombol silang di kanan atas form. Berikut adalah contoh tampilan form master Pengguna:



Gambar 5.17 Menu Master Pengguna

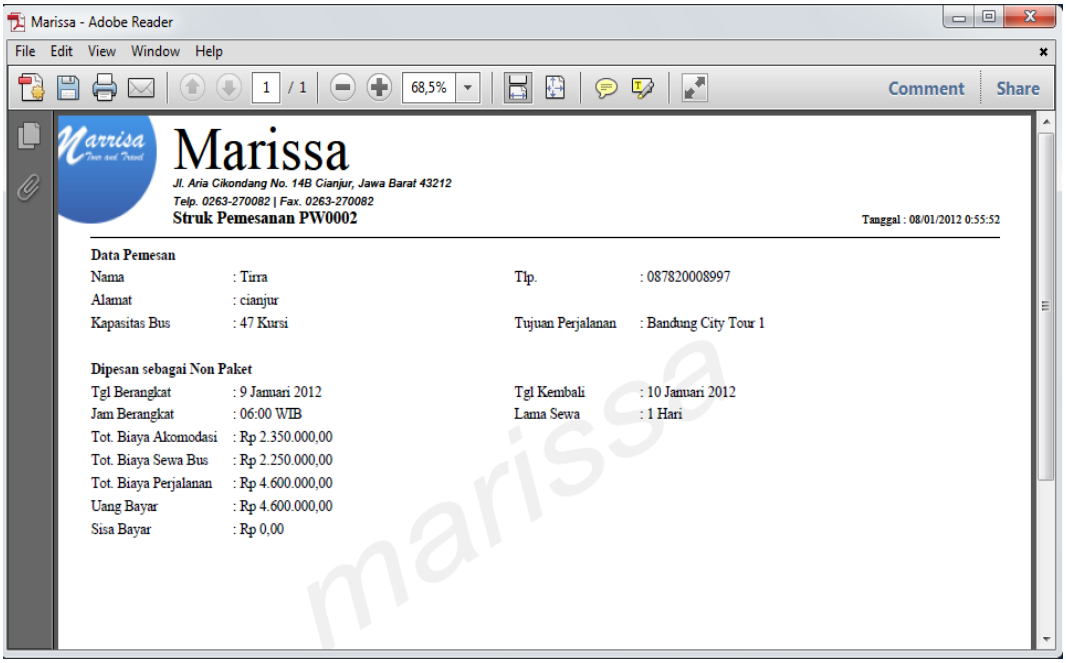
8. *Form* Transaksi penyewaan Bus

Pada form Transaksi, digunakan untuk melakukan transaksi pada saat terjadi penyewaan bus dari pemesan Untuk menginputkan transaksi penyewaan bus, klik tombol transaksi baru dan lengkapi seluruh data yang telah disediakan selanjutnya klik tombol simpan, Untuk mengubah data Pengguna terlebih dahulu pilih Pengguna yang akan dirubah, setelah itu klik tombol ubah transaksi, lalu isi *textbox* sesuai dengan data transaksi yang akan dirubah. Untuk menghapus transaksi penyewaan bus, pilih terlebih dahulu data transaksi mana yang akan dihapus lalu klik tombol hapus. Berikut ini adalah form transaksi penyewaan bus :



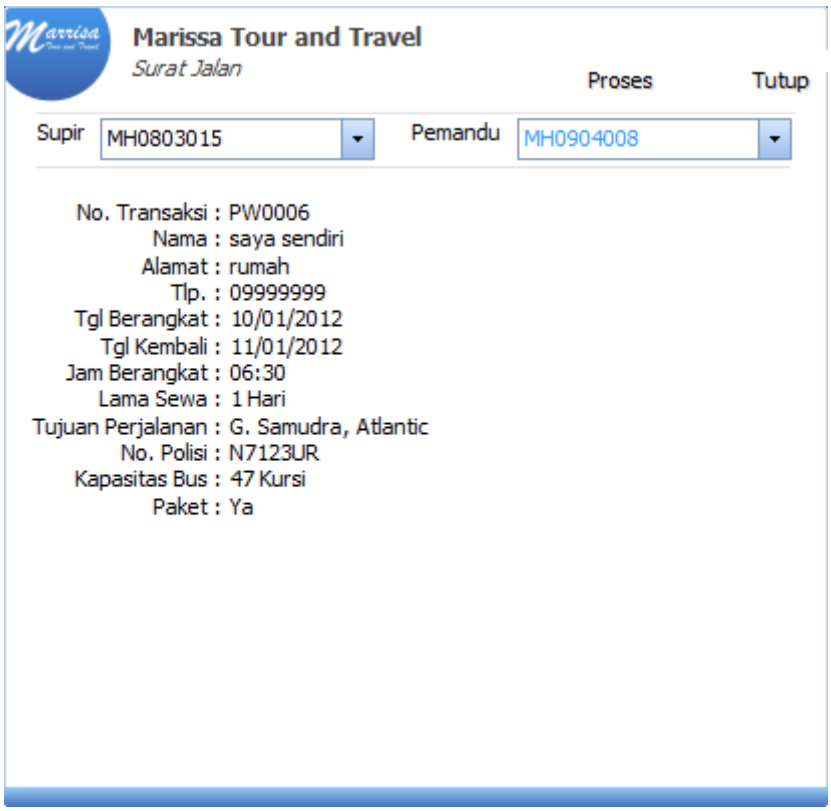
Gambar 5. 18 penyewaan bus

Dan dalam form transaksi penyewaan bus terdapat menu untuk mencetak nota transaksi yang akan di berikan kepada pemesan sebagai tanda bukti penyewaan berikut ini adalah gambar struk transaksi penyewaan :



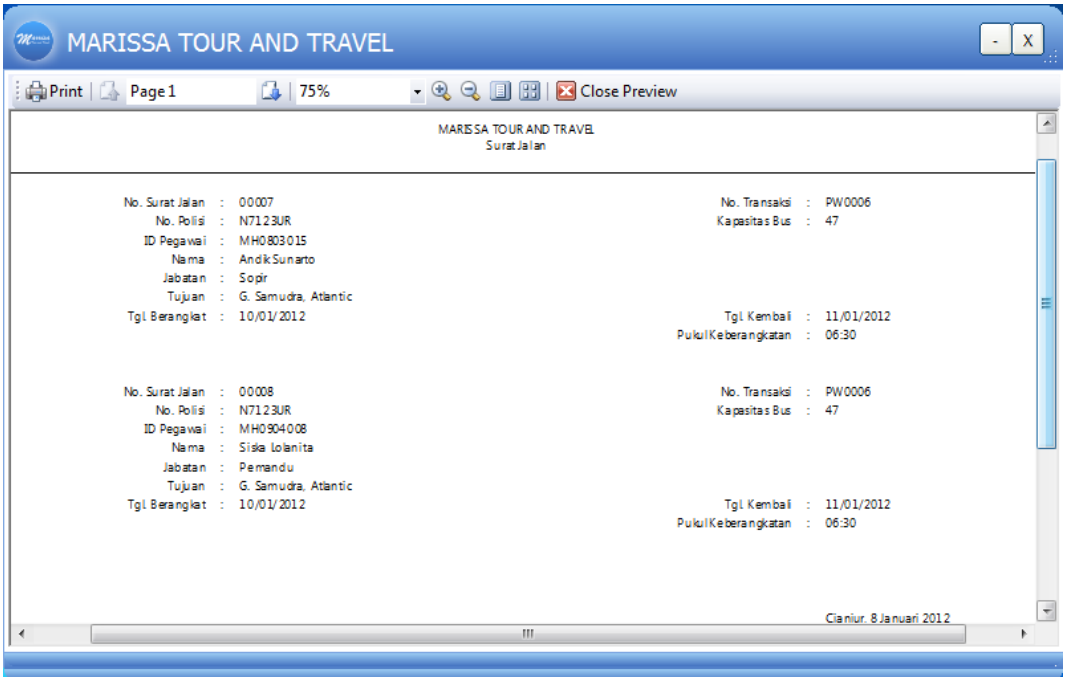
Gambar 5.19 Cetak nota transaksi

Dan dalam form transaksi penyewaan bus terdapat menu untuk mencetak surat jalan yang akan di berikan kepada sopir dan pemandu sebagai tanda bukti surat tugas perjalanan dan surat jalan ini akan bisa di cetak apabila pada saat proses transaksi pemesan sudah melakukan pembayaran Lunas berikut ini adalah gambar surat jalan :



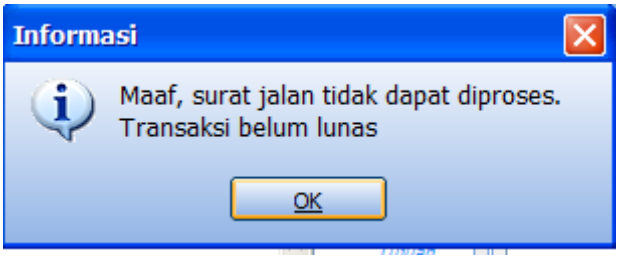
Gambar 5. 20 Cetak surat jalan

Setelah memilih sopi dan pemandu wisata maka klik tombol proses untuk mencetak surat jalan dibawah ini adalah gambar output surat jalan yang diberikan kepada sopir dan pemandu pada saat akan melaksanakan tugas



Gambar 5.21 .*Output* Surat Jalan

Proses cetak surat jalan hanya bisa dilakukan apabila penyewaan bus berstatus lunas dan apabila statusnya belum lunas akan muncul peringatan seperti pada gambar dibawah ini :



Gambar 5. 22 Peringatan transaksi belum lunas

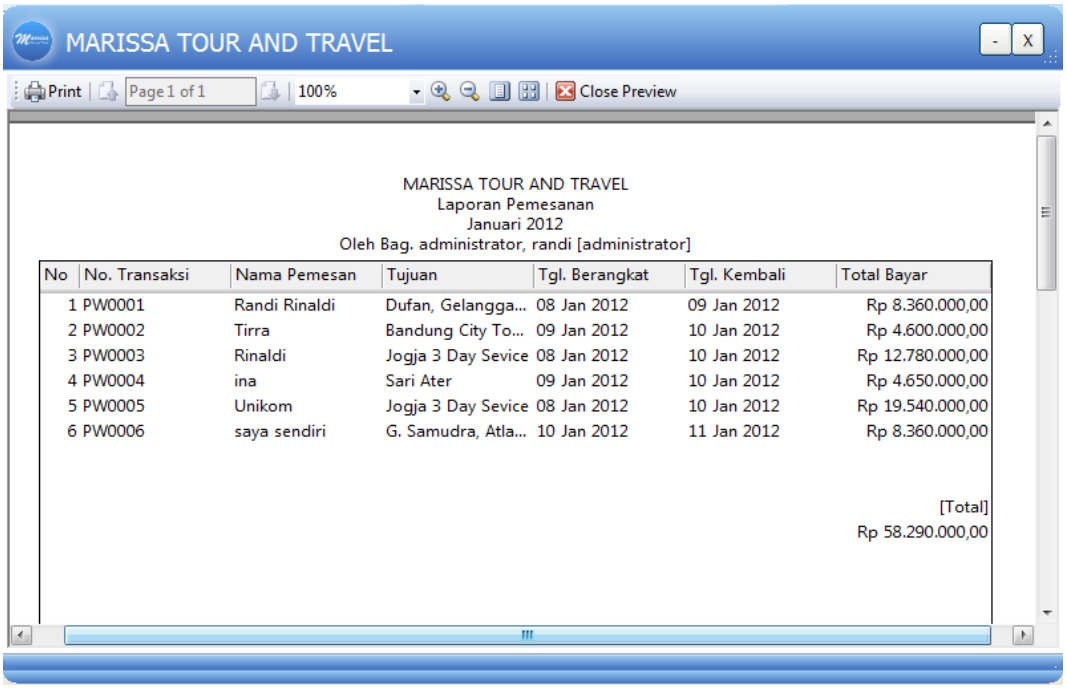
9. *Form* cetak

Pada form cetak, digunakan untuk melakukan laporan penyewaan bus,laporan batal pesan,laporan perawatan bus,dan laporan jadwal sopir dan pemandu.laporan-laporan tersebut dapat diproses hanya dalam periode satu bulan sekali.gambar dibawah ini contoh salah satu form cetak laporan :



Gambar 5.23 Cetak laporan penyewaan Buss

Dan untuk melihat hasil otput dari proses cetak laporan penyewaan klik tombol proses setelah itu akan muncul laporan penyewaan bus seperti yang terlihat pada gambar di bawah ini :



Gambar 5.24 *Output* laporan penyewaan bus

Dalam menu cetak proses laporan batal pesan,perawatan bus,laporan jadwal sopir dan pemandu prosedurnya sama dengan pada saat melakukan proses cetak laporan penyewaan bus.

10. *Form* Bantuan

Pada menu form Bantuan terdapat beberapa sub menu form yaitu info paket wisata,info jadwal sopir,info bus form ini berfungsi untuk informasi melihat info paket wisata,info status bus dan pegawai.sebagai salah satu contoh dibawah ini terdapat gambar mengenai info bus :



No.	No Pilisi	Jumlah Kursi	Sewa Pokok	Status Ketersediaan
1	B7165ZX	60	Rp 3.000.000,00	Terpesan
2	B7189IX	60	Rp 3.000.000,00	Tersedia
3	B7258MH	29	Rp 1.750.000,00	Tersedia
4	B7723ZY	47	Rp 2.250.000,00	Terpesan
5	B7789IZ	60	Rp 3.000.000,00	Tersedia
6	F7347WF	29	Rp 1.750.000,00	Terpesan
7	F7362WU	29	Rp 1.750.000,00	Terpesan
8	F7515F	60	Rp 3.000.000,00	Tersedia
9	F7563F	60	Rp 3.000.000,00	Tersedia
10	F7581SC	47	Rp 2.250.000,00	Terpesan
11	F7585ID	60	Rp 3.000.000,00	Tersedia
12	F7590SC	47	Rp 2.250.000,00	Terpesan
13	F7621F	29	Rp 1.750.000,00	Terpesan
14	F7639F	47	Rp 2.250.000,00	Terpesan

Gambar 5.25 Info rincian bus

Dan info tentang ialah informasi singkat perusahaan Marissa holiday tour and travel gambar nya bisa dilihat seperti dibawah ini :



Gambar 5.26 Tentang (profil perusahaan)

5.2. Pengujian

Pengujian dilakukan untuk mengetahui sejauh mana kinerja sistem. Pengujian yang digunakan untuk menguji Sistem Informasi penyewaan bus ini adalah dengan menggunkan metode pengujian *Black box*. Pengujian *Black box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak.

Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *Black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak, kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.

5.2.1. Rencana Pengujian

Pengujian untuk sistem informasi penyewaan bus yang penulis bangun menggunakan data uji berupa sebuah data dan masukan dari *user*.

Tabel 5.7 Rencana Pengujian

Kelas Uji	Butir Uji	Tingkat Pengujian	Jenis Pngujian
Setting Configuration	Pengaturan Nama Jaringan	Sistem	*Black Box*
Login User	Pengecekan User yang telah terdaftar	Sistem	*Black Box*
Input Data Master	Pengisian Data paket wisata	Modul	*Black Box*
	Pengisian Data bus	Modul	*Black Box*
Input Data Transaksi	Pengisian Data Transaksi penyewaan bus	Modul	*Black Box*
	Konfirmasi perjalanan	Modul	*Black Box*

5.2.2. Kasus dan Hasil Pengujian

Berikut adalah hasil pengujian yang telah penulis lakukan :

1. Pengujian Konfigurasi Jaringan

   Pengujian konfigurasi dilakukan untuk mengatur nama jaringan yang dipakai dan nama database yang digunakan, hasil pengujiannya dapat dilihat pada tabel dibawah ini :

Tabel 5.8 Pengujian Konfigurasi

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	1. Server Name : Randi-pc 2. Database Name : marissa
Yang diharapkan	Dapat masuk ke form selanjutnya/menu Login.
Pengamatan	Tombol *next* dapat berfungsi sesuai dengan yang diharapkan.
Kesimpulan	[ ] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data Masukan	1. Server Name : Database 2. Database Name : Sistem
Yang diharapkan	*User* tidak dapat masuk kedalam menu utama sesuai dengan hak aksesnya
Pengamatan	Tidak dapat masuk kedalam menú utama dan menampilkan pesan "Nama *Database* dan nama *server* tidak ditemukan harap *setting* koneksi terlebih dahulu"
Kesimpulan	[ ] Diterima [ ] Ditolak

2. Pengujian Login Bagian operasional

Pengujian login bag.operasional dilakukan hanya untuk pengecekan *user* yang akan menggunakan sistem, hasil pengujiannya dapat dilihat pada tabel dibawah ini :

Tabel 5.9 Pengujian Login Bagian operasional

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	1 Bagian : operasional 2 Nama Pengguna : operasional 3 Kata Sandi : operasional
Yang diharapkan	Dapat masuk ke form selanjutnya/menu utama sesuai dengan hak aksesnya
Pengamatan	Tombol login dapat berfungsi sesuai dengan yang diharapkan.
Kesimpulan	[ ] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data Masukan	3. Bagian : operasional 4. Nama Pengguna : user 5. Kata Sandi : user
Yang diharapkan	*User* tidak dapat masuk kedalam menu utama Bag. operasional
Pengamatan	Tidak dapat login dan menampilkan pesan di *box* kata sandi berwarna merah
Kesimpulan	[ ] Diterima [ ] Ditolak

3. Pengujian Input Data Master paket wisata

Pengujian input data master paket wisata dilakukan untuk mengecek data yang diinput sesuai dengan yang diharapkan, adapun pengujian yang di lakukan adalah sebagai berikut :

Tabel 5.10 Pengujian Pengisian Data paket wisata

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	1. Klik tombol paket wisata baru 2. Mengisi data paket wisata dengan lengkap 3. Klik tombol simpan
Yang diharapkan	1. Kode akomodasi akan muncul secara automatic 2. Mengisi data paket wisata dengan lengkap 3. Data paket wisata dapat tersimpan dalam database
Pengamatan	1. Dapat mengisi *text* box pada *form* pendaftaran paket wisata baru dan menambah paket wisata anggota sesuai dengan yang diharapkan 2. Mengisi data paket wisata dengan lengkap 3. Data paket wisata dapat disimpan dalam database sesuai dengan yang diharapkan
Kesimpulan	[ ] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data Masukan	Tidak mengisi data paket wisata dengan lengkap
Yang diharapkan	Tidak dapat menyimpan data paket wisata dalam database
Pengamatan	Menampilkan pesan “Wajib Mengisi Semua Data”pada *form* pendaftaran paket wisata baru
Kesimpulan	[ ] Diterima [ ] Ditolak

4. Pengujian Input Data master bus

Pengujian input data master bus dilakukan untuk mengecek data yang diinput sesuai dengan yang diharapkan, adapun pengujian yang di lakukan adalah sebagai berikut :

Tabel 5.11 Pengujian Pengisian Data master bus

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	1. No polisi,kode bus,status ketersedian,kode bus jumlah kursi dan sewa pokok 2. Klik tombol simpan
Yang diharapkan	1. Menampilkan data bus yang akan dipinjam 2. Data bus dapat tersimpan dalam database
Pengamatan	1. Dapat menampilkan data bus sesuai yang diharapkan 2. Dapat menampilkan data bus di form transaksi penyewaan bus,perwatan bus 3. Data master bus dapat disimpan sesuai yang diharapkan.
Kesimpulan	[ ] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data Masukan	1. Tidak menentukan kode bus 2. Tidak menentukan harga sewa pokok bus
Yang diharapkan	Tidak dapat menyimpan master bus dalam database
Pengamatan	Menampilkan pesan “Harap Isi semua data” pada form master bus.
Kesimpulan	[ ] Diterima [ ] Ditolak

5 Pengujian Pengisian Transaksi penyewaan bus

Pengujian input data penyewaan bus dilakukan untuk mengecek data yang diinput sesuai dengan yang diharapkan, adapun pengujian yang di lakukan adalah sebagai berikut :

Tabel 5.12 Pengujian Transaksi penyewaan bus

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	1. Input no transaksi,nama,alamat,tlp,tgl berangkat dan kembali,lama sewa,jam berangkat,tujuan, dan uang bayar 2. Menetuka paket wisata yang akan di pilih 3. Menentukan uang bayar 4. Klik tombol simpan
Yang diharapkan	1. Menampilkan penyewaan sesuai yang diharapkan 2. Menampilkan nota pembayaran sesuai yang diharapkan 3. Menampilkan surat jalan sesuai yang diharapkan 4. Data transaksi penyewaan bus dapat tersimpan dalam database
Pengamatan	1. Dapat Menampilkan penyewaan sesuai yang diharapkan 2. Dapat Menampilkan nota pembayaran sesuai yang diharapkan 3. Dapat Menampilkan surat jalan sesuai yang diharapkan 4. Data transaksi penyewaan bus dapat tersimpan dalam database
Kesimpulan	[ ] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data Masukan	1. Tidak menentukan tujuan wisata yang di pesan 2. Tidak menentukan kapasitas bus yang dipesan

Yang diharapkan	Tidak dapat menyimpan data transaksi penyewaan bus dalam database

Pengamatan	Menampilkan pesan “Harap Isi no anggota atau nis” pada form pengembalian siswa dan “Harap Isi Nip” pada form pengembalian guru/tu.
Kesimpulan	[ ] Diterima [ ] Ditolak

6 Pengujian input konfirmasi perjalanan

Pengujian input konfirmasi perjalanan dilakukan hanya untuk pengecekan data sopir pemandu wisata dan bus untuk mengupdate ketersedian sopir,bus dan pemandu, hasil pengujiannya dapat dilihat pada tabel dibawah ini:

Tabel 5.13 Pengujian input konfirmasi perjalanan

Kasus dan Hasil Uji (Data Normal)	
Data Masukan	1. no transaksi penyewaan
Yang diharapkan	Dapat mengupdate ketersedian sopir pemendu beserta bus menjadi status tersedia
Pengamatan	Tombol ok dapat mengupdate jadwal setelah memasukan no transaksi
Kesimpulan	[ ] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)	
Data Masukan	6. Bukan no transaksi penyewaan

Tabel 5.13 Pengujian input konfirmasi perjalanan (lanjutan)

Yang diharapkan	*Tidak dapat mengupdate jadwal dan ketersedian*
Pengamatan	Tidak dapat mengupdate ketersediaan
Kesimpulan	[ ] Diterima [ ] Ditolak

5.2.3. Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus uji sampel diatas, dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.