Dear all,

IF-UNIKOM-UP-DOC-2009

DOKUMEN PEMBANGUNAN PERANGKAT LUNAK

<SYSTEM INFORMASI RENTAL PLAYSTATION 3>

Versi: <Draft-tgl/Final>

untuk:

Universitas Komputer Indonesia

Dipersiapkan oleh:

Bambang Imam Hermawan ( 10110511 )

Aidil Pratama Putra ( 10110517 )

Giri Muria Shaleh ( 10110531 )

Chandra Agustian ( 10108100 )

Asep Anwar Musadad ( 10108109 )

Jurusan Teknik Informatika – Universitas Komputer Indonesia

Jl. Dipati Ukur Nomor 112-114, 40132

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| unikom | **Program Studi**  **Teknik Informatika UNIKOM** | Nomor Dokumen | | Halaman |
| *PP IF-UNIKOM-UP-DOC-2009* | | *1/58* |
| Revisi | *0* | *28-07-2013* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A |  |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX  TGL | - | A | B | C | D | E | F | G |
| Ditulis oleh |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

Daftar Halaman Perubahan

|  |  |  |  |
| --- | --- | --- | --- |
| Halaman | Revisi | Halaman | Revisi |
|  |  |  |  |

Daftar Isi

1 Pendahuluan 6

1.1 Tujuan Penulisan Dokumen 6

1.2 Lingkup Masalah 6

1.3 Aturan Penomoran 6

1.4 Referensi 6

1.5 Deskripsi Umum Dokumen (Ikhtisar) 7

2 Kebutuhan Perangkat Lunak 7

2.1 Deskripsi Umum Sistem 7

2.2 Fitur Utama Perangkat Lunak 8

2.2.1 Kebutuhan Fungsional 8

2.2.2 Kebutuhan Non Fungsional 8

2.3 Model Use Case 9

2.3.1 Diagram Use Case 9

2.3.2 Definisi Actor 9

2.3.3 Definisi Use Case 9

2.3.4 Skenario Use Case 10

2.4 Spesifikasi Tambahan 13

2.5 Glossary 13

3 Model Analisis 14

3.1 Realisasi Use Case Tahap Analisis 14

3.2 Diagram Kelas Keseluruhan 14

3.3 Kelas Analisis 15

3.4 Paket Analisis 16

3.4.1 Identifikasi Paket Analisis 16

3.4.2 Identifikasi Kelas Analisis tiap Paket 16

3.5 Deskripsi Arsitektur 16

3.6 Pedoman Perancangan 16

4 Model Perancangan 18

4.1 Realisasi Use Case Tahap Perancangan 18

4.1.1 Use Case <nama use case 1> 18

4.1.1.1 Identifikasi Elemen WAE - Logical View 18

4.1.1.2 Identifikasi Kelas Perancangan 18

4.1.1.3 Sequence Diagram 19

4.1.1.4 Diagram Kelas Perancangan 19

4.1.2 Use Case <nama use case 2> 19

4.1.2.1 Identifikasi Elemen WAE - Logical View 19

4.1.2.2 Identifikasi Kelas Perancangan 19

4.1.2.3 Sequence Diagram 20

4.1.2.4 Diagram Kelas Perancangan 20

4.1.3 Use Case <nama use case 3> 20

4.1.3.1 Identifikasi Elemen WAE - Logical View 20

4.1.3.2 Identifikasi Kelas Perancangan 21

4.1.3.3 Sequence Diagram 21

4.1.3.4 Diagram Kelas Perancangan 21

4.2 Perancangan Detil Elemen Logical View 23

4.2.1 Stereotyped Class <nama elemen logical view> 23

4.2.2 Stereotyped Class <nama elemen logical view> 24

4.3 Perancangan Detil Kelas 24

4.3.1 Kelas <nama kelas> 24

4.3.2 Kelas <nama kelas> 25

4.4 Diagram Kelas Keseluruhan 27

4.5 Algoritma/Query 27

4.6 Diagram Statechart 29

4.7 Perancangan Antarmuka 32

4.8 Perancangan Representasi Persistensi Kelas 42

4.9 Coding Standard dan Naming Convention 42

4.10 Deployment Diagram 47

5 Implementasi 52

5.1 Lingkungan Implementasi 52

5.2 Implementasi Kelas 52

5.3 Implementasi Elemen WAE (Component View) 53

5.4 Implementasi Antarmuka 54

5.5 File Lain 54

6 Pengujian 54

6.1 Rencana dan Prosedur Pengujian 54

6.1.1 Rencana Pengujian 54

6.1.2 Prosedur Pengujian 55

6.2 Kasus Uji 55

6.2.1 Pengujian Use Case <nama use case> 55

6.3 Defect dan Status Perbaikan 55

6.4 Evaluasi Pengujian 57

7 Lampiran 58

# 

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini bertujuan sebagai acuan atau panduan baik bagi pengembang dan pengguna perangkat lunak selama dalam pengembangan perangkat lunak yang akan dibangun. Bagi pengembang, SKPL ini dapat digunakan sebagai acuan dalam setiap tahapan pengembangan, agar sesuai dengan kebutuhan yang diinginkan oleh pengguna dan tujuan perangkat lunak itu sendiri.Sedangkan bagi pihak pengguna, SKPL ini digunakan untuk mencatat semua spesifikasi kebutuhan dari perangkat lunak yang dikembangkan dan harapan yang diinginkan. Dokumen SKPL ini berisi spesifikasi kebutuhan dari perangkat lunak rental PS yang akan dikembangkan.

## Lingkup Masalah

Berdasarkan permasalahan yang terjadi di Rental PlayStation, maka diidentifikasi sebagai berikut:

1. Rekap laporan transaksi rental masih tertulis kemungkinan besar dapat terjadi kerusakan dan kehilangan data, sehingga laporan transaksi dengan keuangan tidak sesuai.

Pengelolaan data member masih kurang terorganisir.

## Aturan Penomoran

Bagian ini diisi dengan aturan penomoran yang digunakan dalam dokumen.

## Referensi

Dokumen-dokumen yang digunakan sebagai referensi dalam pembuatan SKPL ini adalah sebagai berikut :

1. IEEE Std 830-1993, IEEE Recommended Practice for Software Requirement Specifications.
2. GL01A : Panduan Pengisian Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL).
3. GL01AT : Template Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL).
4. Presman, Roger. 2001. Software Engineering : A Practitioer’s Approach, 5th ed. McGraw-Hill, Singapore.

## Deskripsi Umum Dokumen (Ikhtisar)

Dokumen SKPL ini memberikan spesifikasi dari produk yang akan dibangun dan sesuai dengan kebutuhan yang diberikan oleh pihak pengguna dan telah disepakati bersama dengan pengembang. Gambaran umum dan khusus dari perangkat lunak, yang meliputi:

* 1. Tujuan dan fungsi perangkat lunak

Menjelaskan tujuan dari pembangunan perangkat lunak pengelolaan Rental *PlayStation*, secara detail fungsi dibangunya perangkat lunak pengelolaan Rental *PlayStation* ini adalah untuk mempermudah Petugas dalam mengelola data member dan data penyewa .

* 1. Kemampuan perangkat lunak

Mendeskripsikan kemampuan yang dimiliki oleh perangkat lunak yang dibangun untuk pengolahan data member dan penyewa.

* 1. Antarmuka pemakai, perangkat keras, perangkat lunak, dan komunikasi.

Menguraikan secara rinci kebutuhan dalam perancangan perangkat lunak yang akan dibangun meliputi pemakai, perangkat keras, perangkat lunak dan komunikasi.

* 1. Batasan-batasan perangkat Lunak

Membahas secara detail batasan yang dapat dilakukan dan diatasi oleh perangkat lunak.

* 1. Karakteristik pengguna

Menjelaskan karateristik pengguna yang akan menggunakan perangkat lunak yang berhubungan untuk kegiatan maintanance perangkat lunak ini sendiri.

* 1. Ketergantungan perangkat lunak

Menjelaskan ketergantungan perangkat lunak terhadap lingkungan implementasi.

# Kebutuhan Perangkat Lunak

## Deskripsi Umum Sistem

Sistem yang dibuat merupakan suatu perangkat lunak yang khusus digunakan dalam sebuah rental playstation. Aplikasi ini dapat memudahkan pengguna khususnya petugas untuk mengolah informasi data di rental playstation tersebut. Penyewa yang datang akan di layani oleh petugas. Penyewa bisa menyewa playstation yang tersedia di dalam database petugas dengan status tidak sedang digunakan. Playstation yang ada dalam database merupakan playstation yang telah memiliki status sedang atau tidak sedang disewakan. Bila database menunjukan status playstation tidak sedang digunakan dan penyewa hendak melakukan penyewaan maka dapat langsung dilakukan perentalan.

## Fitur Utama Perangkat Lunak

SRS-FU-001 Free Software : Perangkat lunak yang dilengkapi dengan kode sumber, dapat digunakan oleh siapa saja dan bebas untuk dikembangkan sendiri oleh pemakai. Untuk mendapatkannya bisa saja gratis ataupun membayar dengan harga yang relatif murah.

SRS-FU-002 Perangkat lunak transaksi penyewaan / rental

SRS-FU-003 Perangkat lunak produktivitas kerja

### Kebutuhan Fungsional

Contoh:

SRS-F-001 Sistem menyediakan fasilitas bagi pengguna untuk memilih tema tampilan.

SRS-F-002 Sistem menyediakan fasilitas bagi pengguna untuk mengelola database.

SRS-F-003 penyimpanan data pelanggan ke database

SRS-F-004 system menyediakan fitur bagi setiap actor untuk keluar masuk aplikasi sesuai dengan hak akses yang dimilikinya

SRS-F-005 system menyediakan fitur untuk mengelola data pembayaran member dan user

SRS-F-006 system menyediakan fitur untuk mengelola data pendaftaran member dan user

### Kebutuhan Non Fungsional

Performansi :

SRS-NF-001 Spesifikasi komputer untuk aplikasi ini minimal intel pentium IV, memori 256 MB dan Hard Disk yang cukup untuk menampung database.

SRS-NF-002 Dalam pencarian informasi atau pemasukan data dapat dilakukan dengan kecepatan < 15 detik.

SRS-NF-003 Sistem harus dapat melakukan pencetakan nota atau struk tanda pembayaran dengan cepat dan jelas.

Antarmuka :

SRS-NF-004 Tampilan program harus sederhana dan mudah digunakan.

SRS-NF-005 Menu perintah harus jelas atau menunjukan fungsi-fungsinya sehingga dapat dimengerti oleh penggunanya.

Keamanan :

SRS-NF-006 Password setiap user sudah terenkripsi secara otomatis di database.

Batasan Data :

SRS-NF-007 Tipe data yang digunakan harus merepresentasikan jumlah benda dan jumlah uang.

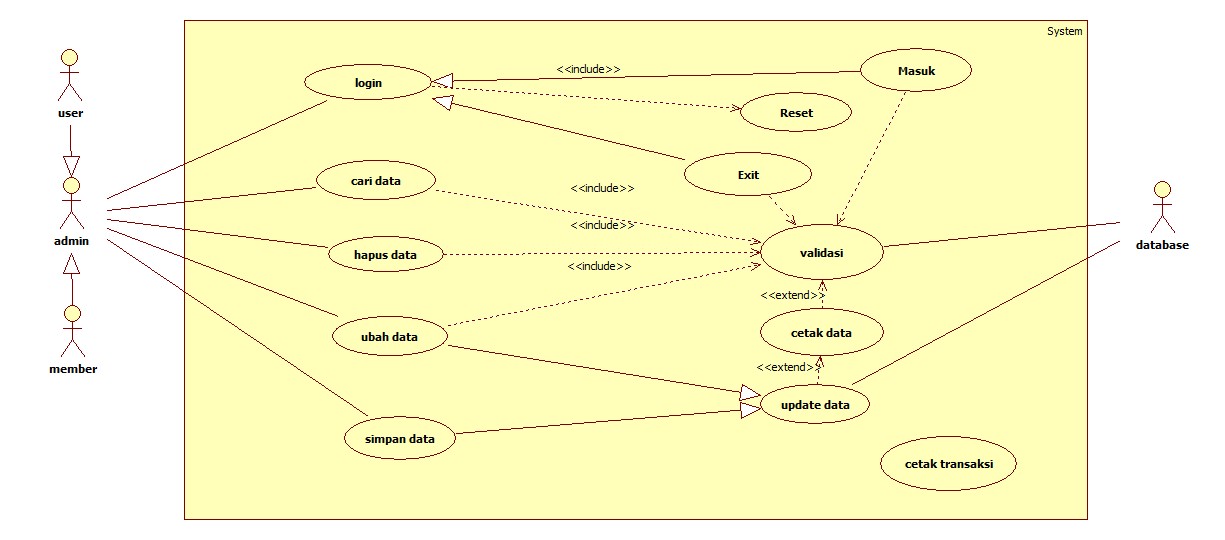
Modus Operasi :

SRS-NF-008 Setiap user memiliki password masing-masing untuk login.

SRS-NF-008 Setiap mamberikan informasi akan terjadi pengecekan terlebih dahulu tentang playstation yang akan dipakai.

## Model Use Case

### Diagram Use Case

.

### Definisi Actor

Bagian ini diisi dengan daftar actor dan deskripsi role untuk actor tersebut. Deskripsi role harus menjelaskan wewenang pada role tersebut dalam perangkat lunak. Bisa dibuat dalam bentuk tabel berikut:

|  |  |  |
| --- | --- | --- |
| *No* | *Actor* | *Deskripsi* |
| *1* | *Admin* | *Admin mengelola seluruh pengregistrasian kepada user dan member dan mengatur seluruh info dari database* |
| *2* | *User* | *User melakukan registrasi, pendaftaran sewa ps, pembayaran sewa dan sewa kepada admin* |
| *3* | *Member* | *Member Melakukan registrasi, pendaftaran member, pendaftaran sewa ps, pembayaran member, pembayaran sewa, dan sewa kepada admin* |

### Definisi Use Case

|  |  |  |
| --- | --- | --- |
| *No* | *Use Case* | *Deskripsi* |
| *1* | *Login* | *Sistem menampilkan suatu tampilan untuk keamanan pengguna.* |
| *2* | *Cari Data* | *System menampilkan untuk mencari data dari hasil registrasi user dan member* |
| *3* | *Hapus Data* | *System untuk menghapus data registrasi member dan user oleh admin* |
| *4* | *Ubah Data* | *System untuk mengubah data registrasi member dan user oleh admin* |
| *5* | *Simpan Data* | *System untuk menyimpan data registrasi member dan user oleh admin* |
| *6* | *Masuk Halaman* | *System menampilkan untuk masuk kehalaman utama setelah melakukan login* |
| *7* | *Reset* | *System men-reset seluruh text apabila admin tidak jadi melakukan penulisan atau menghandle text penulisan.* |
| *8* | *Exit* | *Keluar system* |
| *9* | *Validasi* | *System mengecek kesesuaian data sehingga dapat di akses, apabila valid data masuk ke system dan apabila tidak valid data error dan melakukan pengecekan kembali.* |
| *10* | *Cetak Data* | *Mencetak data member dan user* |
| *11* | *Update Data* | *Mengupdate data registrasi dan user* |
| *12* | *Cetak Transaksi* | *Mencetak data transaksi atau pembayaran member dan user* |

### Skenario Use Case

*Contoh:*

*Nama Use Case: Login*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman login* |  |
|  | *2. Menampilkan halaman login* |
| *3. Masukkan username dan password* |  |
| *4. menekan tombol login* |  |
|  | *5. Validasi login* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

*Nama Use Case: Masuk*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman login* |  |
|  | *2. Menampilkan halaman login* |
| *3. Masukkan username dan password* |  |
| *4. menekan tombol login* |  |
|  | *5. Validasi login* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

*Nama Use Case: Reset*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman login* |  |
|  | *2. Menampilkan halaman login* |
| *3. Masukkan username dan password* |  |
| *4. menekan tombol reset* |  |
|  | *5. System membackup ulang username dan password* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

*Nama Use Case: Exit*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman login* |  |
|  | *2. Menampilkan halaman login* |
| *3. menekan tombol keluar* |  |
|  | *4. System keluar* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

*Nama Use Case: Cari Data*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman Database, daftar user, daftar member, dan main member* |  |
|  | *2. Menampilkan halaman Database, daftar user, daftar member, dan main member* |
| *3. Masukkan nama data yang dicari* |  |
| *4. menekan tombol Cari* |  |
|  | *5. System Melakukan Pencarian* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

*Nama Use Case: Hapus Data*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman daftar user, daftar member, dan main member* |  |
|  | *2. Menampilkan halaman daftar user, daftar member, dan main member* |
|  | *3.System menampilkan Database* |
| *4.Memilih data pelanggan di database* |  |
| *5. Menekan tombol hapus* |  |
|  | *6.System menghapus database di gridnya* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

*Nama Use Case: Ubah Data*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman daftar user, daftar member, dan main member* |  |
|  | *2. Menampilkan halaman daftar user, daftar member, dan main member* |
|  | *3.System menampilkan Database* |
| *4.Memilih data pelanggan di database* |  |
| *5. Mengubah database* |  |
| *6. Menekan tombol ubah* |  |
|  | *7. System Menyimpan hasil ubah database* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

*Nama Use Case: Simpan Data*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman Daftar personal, Daftar member,Pembayaran Personal, Pembayaran member* |  |
|  | *2. Menampilkan halaman Daftar personal, Daftar member,Pembayaran Personal, Pembayaran member* |
| *3. Masukkan data-data di form* |  |
| *4. menekan tombol submit* |  |
|  | *5. System Melakukan Penyimpanan* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

*Nama Use Case: Validasi*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman login dan halaman database / keterangan* |  |
|  | *2. Menampilkan halaman login dan halaman database / keterangan* |
| *3. Masukkan username dan password* |  |
| *4. menekan tombol login, ubah , cari dan hapus* |  |
|  | *5. Validasi login, ubah data, cari data, dan hapus data* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

*Nama Use Case: Cetak Data*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman report transaksi personal, dan halaman report transaksi member* |  |
|  | *2. Menampilkan halaman report transaksi personal, dan halaman report transaksi member* |
|  | *3. System menampilkan database di halaman report* |
| *4. menekan tombol tombol print* |  |
|  | *5. Mencetak data transaksi* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

*Nama Use Case: Update Data*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman Database / Keterangan* |  |
|  | *2. Menampilkan halaman lKeterangan* |
| *3. Menekan tombol lihat* |  |
|  | *4.System menampilkan Database* |
| *5. Mengubah database* |  |
| *6. Menekan tombol ubah* |  |
|  | *7. System Menyimpan hasil ubah database* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

*Nama Use Case: Cetak Transaksi*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Masuk ke halaman report transaksi personal, dan halaman report transaksi member* |  |
|  | *2. Menampilkan halaman report transaksi personal, dan halaman report transaksi member* |
|  | *3. System menampilkan database di halaman report* |
| *4. menekan tombol tombol print* |  |
|  | *5. Mencetak data transaksi* |
| *Skenario Alternatif* |  |
|  |  |
|  |  |

## Spesifikasi Tambahan

Bagian ini diisi dengan informasi tambahan mengenai setiap atau seluruh use case utama, terutama mengenai kebutuhan non fungsional.

## Glossary

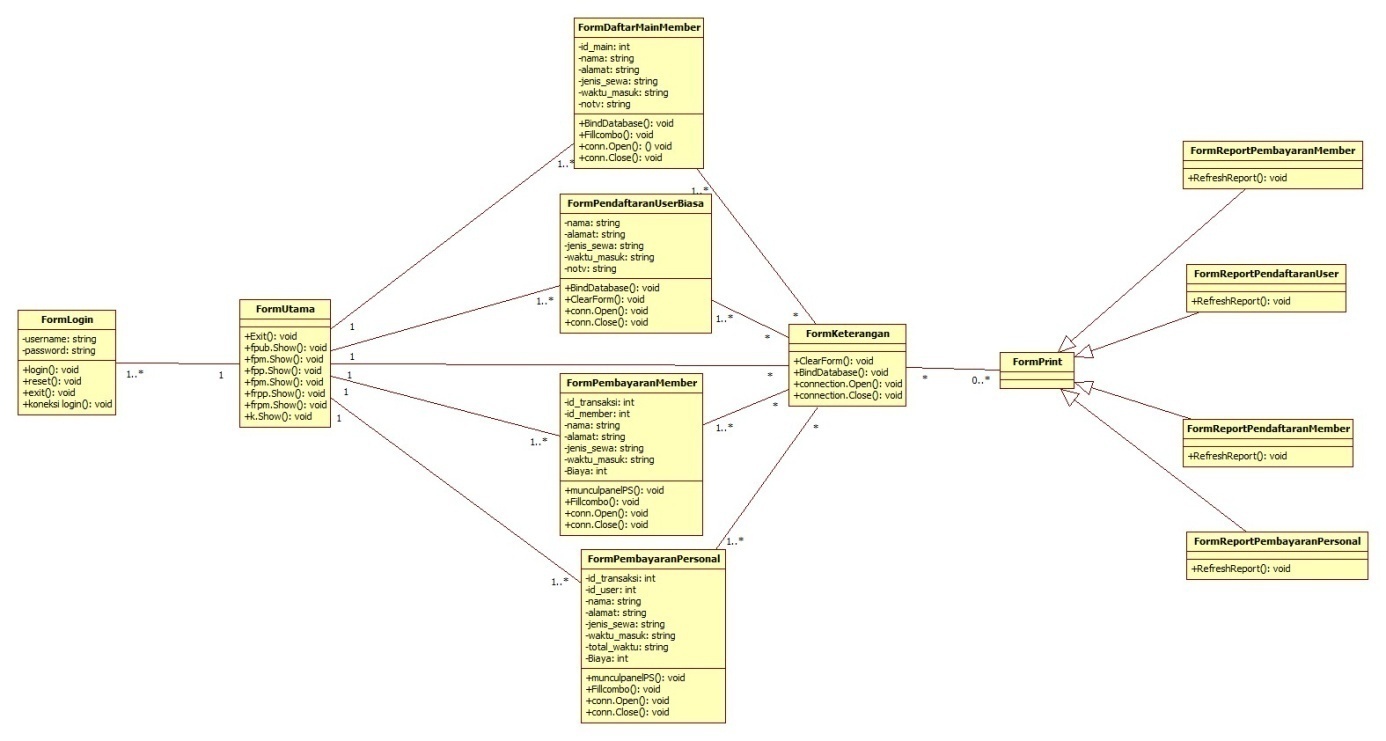
Bagian ini diisi dengan daftar istilah yang digunakan, terutama istilah yang spesifik terhadap domain problem.

# Model Analisis

## Realisasi Use Case Tahap Analisis

Bagian ini dibuat subbab per use case. Setiap subbab diisi dengan diagram kelas analisis dan sequence diagram yang menggambarkan interaksi setiap objek dari kelas analisis yang terlibat di dalam use case tersebut. Lengkapi dengan kalimat pengantar (jangan hanya diisi diagram saja)

## Diagram Kelas Keseluruhan



* FormLogin : kelas ini berisi data untuk halaman login.
* FormUtama : kelas ini berisi menu-menu untuk akses ke halaman lain.
* FormDaftarMainMember : kelas ini berisi untuk daftar main member bila mendaftar.
* FormPendaftaranUserBiasa : kelas ini berisi untuk daftar main pelanggan biasa.
* FormPembayaranMember : kelas ini berisi halaman untuk bayar member.
* FormPembayaranPersonal : kelas ini berisi halaman untuk bayar pelanggan biasa.
* FormKeterangan : kelas ini berisi halaman untuk keterangan pelanggan biasa dan member.
* FormPrint : kelas ini laporan kosong untuk mencetak data transaksi pelanggan dan member.
* FormReportPendaftaranUser : kelas ini untuk mencetak data pendaftaran pelanggan biasa.
* FormReportPendaftaranMember : kelas ini untuk mencetak data pendaftaran member.
* FormReportPembayaranPersonal : kelas ini untuk mencetak data pembayaran pelanggan biasa.
* FormReportPembayaranMember : kelas ini untuk mencetak data pembayaran member.

## Kelas Analisis

Bagian ini diisi dengan daftar seluruh kelas kelas analisis dalam tabel berikut:

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas* | *Jenis* |
| *1* | *FormLogin* | *Boundary* |
| *2* | *FormUtama* | *Entity* |
| *3* | *FormDaftarMainMember* | *Entity* |
| *4* | *FormPendaftaranUserBiasa* | *Entity* |
| *5* | *FormPembayaranMember* | *Entity* |
| *6* | *FormPembayaranPersonal* | *Entity* |
| *7* | *FormKeterangan* | *Control* |
| *8* | *FormPrint* | *Control* |
| *9* | *FormReportPendaftaranUser* | *Entity* |
| *10* | *FormReportPendaftaranMember* | *Entity* |
| *11* | *FormReportPembayaranPersonal* | *Entity* |
| *12* | *FormReportPembayaranMember* | *Entity* |

Untuk setiap kelas analisis, lakukan (dengan melengkapi subbab-subbab berikutnya):

* identifikasi tanggung-jawab (responsibility)
* identifikasi atribut

|  |  |  |
| --- | --- | --- |
| *Nama Kelas* | *Daftar Tanggung-Jawab* | *Daftar Atribut* |
| *FormLogin* | *1.login*  *2.Reset*  *3.Exit* | *1.username*  *2.password* |
| *FormUtama* | *1.Exit* |  |
| *FormDaftarMainMember* | *1.koneksi*  *2.Tambah*  *3.Hapus*  *4.Ubah*  *5.Cari* | *1.id\_main*  *2.nama*  *3.alamat*  *4.jenis\_sewa*  *5.waktu\_masuk*  *6.notv* |
| *FormPendaftaranUserBiasa* | *1.koneksi*  *2.Tambah*  *3.Hapus*  *4.Ubah*  *5.Cari* | *1.nama*  *2.alamat*  *3.jenis\_sewa*  *4.waktu\_masuk*  *5.notv* |
| *FormPembayaranMember* | *1.koneksi*  *2.Tambah* | *1.id\_transaksi*  *2.id\_member*  *3.nama*  *4.alamat*  *5.jenis\_sewa*  *6.waktu\_masuk*  *7.biaya* |
| *FormPembayaranPersonal* | *1.koneksi*  *2.Tambah* | *1.id\_transaksi*  *2.id\_member*  *3.nama*  *4.alamat*  *5.jenis\_sewa*  *6.waktu\_masuk*  *7.biaya* |
| *FormKeterangan* | *1.koneksi* |  |
| *FormPrint* |  |  |
| *FormReportPendaftaranUser* | *1.cetak data* |  |
| *FormReportPendaftaranMember* | *1.cetak data* |  |
| *FormReportPembayaranPersonal* | *1.cetak data* |  |
| *FormReportPembayaranMember* | *1.cetak data* |  |

## Paket Analisis

### Identifikasi Paket Analisis

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Paket* | *Use Case Terkait* |
| *1.* | *Paket Pengelolaan Informasi Rental ps 3* | *1. Pengelolaan Informasi Data User*  *2. Pengelolaan Informasi Data Member*  *3. Pengelolaan Informasi pendaftaran User*  *4.Pengelolaan Informasi Pendaftaran Member*  *5.Pengelolaan Informasi Pembayaran User*  *6. Pengelolaan Informasi Pembayaran Member* |
| *2* | *Paket Pengelolaan Data rental PS 3* |
| *1. Pengelolaan Data User*  *2. Pengelolaan Data Member*  *3. Pengelolaan Data Pendaftaran User*  *4. Pengelolaan Data Pendaftaran Member*  *5. Pengelolaan Data Pembayaran User*  *6. Pengelolaan Data Pembayaran Member* |

Gambarkan pula **diagram package**, serta berikan uraian singkat mengenai diagram tersebut. Diagram package menggambarkan ketergantungan antar package. Lengkapi daftar perubahan jika terjadi perubahan.

### Identifikasi Kelas Analisis tiap Paket

Bagian ini diisi dengan hasil identifikasi kelas analisis untuk setiap paket analisis dengan mengacu pada skenario setiap use case. Sebuah kelas seharusnya tidak muncul di lebih dari satu paket. Jika sebuah kelas terlibat di dua use case yang berbeda paket, alokasikan kelas di salah satu paket. Hal ini akan menggambarkan ketergantungan antar paket.

*Contoh:*

|  |  |  |  |
| --- | --- | --- | --- |
| *No* | *Nama Paket* | *Nama Kelas Analisis* | *Jenis Kelas*  *(Boundary, Control, Entity)* |
| *1* | *Paket xxx* | *1.*  *2.*  *3.* |  |
|  |  |  |  |

## Deskripsi Arsitektur

Bagian ini diisi dengan gambaran umum arsitektur perangkat lunak, mis. arsitektur client-server atau arsitektur aplikasi berbasis web.

## Pedoman Perancangan

Framework .NET didisain untuk dapat memenuhi beberapa tujuan berikut ini :

• Untuk menyediakan lingkungan kerja yang konsisten bagi bahasa pemrograman yang

berorientasi objek ( *object-oriented programming - OOP* )

• Untuk menyediakan lingkungan kerja di dalam mengeksekusi kode yang dapat

meminimaliasi proses *software deployment* dan menghindari konflik

penggunaan versi software yang dibuat.

• Untuk menyediakan lingkungan kerja yang aman dalam hal pengeksekusian kode,

termasuk kode yang dibuat oleh pihak ketiga ( *third party).*

• Untuk menyediakan lingkungan kerja yang dapat mengurangi masalah pada persoalan

performa dari kode atau dari lingkungan *interpreter* nya.

# Model Perancangan

## Realisasi Use Case Tahap Perancangan

### Use Case <login>

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### Identifikasi Elemen WAE - Logical View

Bagian ini diisi dengan hasil identifikasi elemen WAE-Logical View yang merealisasikan use case:

* *server page*
* *client page*
* *html form*

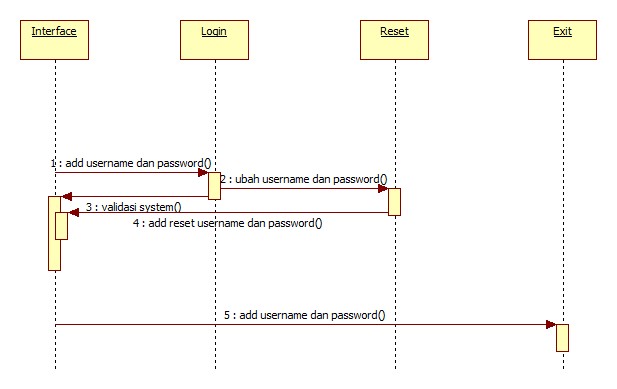
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Elemen Lojik WAE* | *Stereotype* |
|  |  | *Server Page* |
|  |  | *Server Page* |
|  |  | *Client Page* |

#### Identifikasi Kelas Perancangan

Buat daftar kelas analisis yang merealisasikan use case dalam tabel berikut:

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



#### Diagram Kelas Perancangan

Buatlah diagram kelas perancangan lengkap dengan Elemen WAE-Logical View (jika berbasis web).

### Use Case <cari data>

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### Identifikasi Elemen WAE - Logical View

Bagian ini diisi dengan hasil identifikasi elemen WAE-Logical View yang merealisasikan use case:

* *server page*
* *client page*
* *html form*

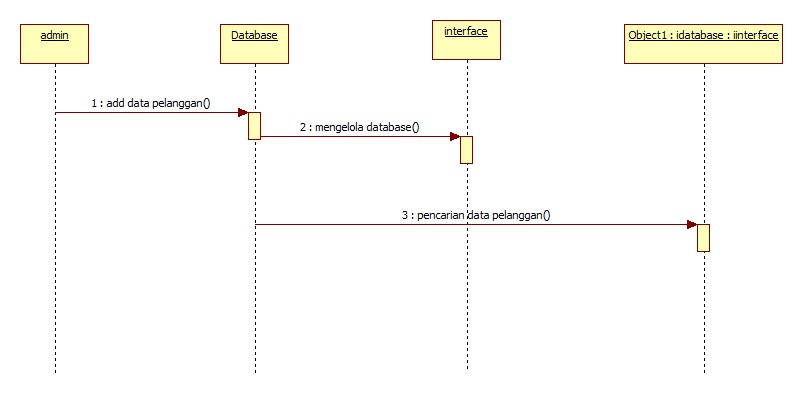
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Elemen Lojik WAE* | *Stereotype* |
|  |  | *Server Page* |
|  |  | *Server Page* |
|  |  | *Client Page* |

#### Identifikasi Kelas Perancangan

Buat daftar kelas analisis yang merealisasikan use case dalam tabel berikut:

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



#### Diagram Kelas Perancangan

Buatlah diagram kelas perancangan lengkap dengan Elemen WAE-Logical View (jika berbasis web).

### Use Case <hapus data>

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### Identifikasi Elemen WAE - Logical View

Bagian ini diisi dengan hasil identifikasi elemen WAE-Logical View yang merealisasikan use case:

* *server page*
* *client page*
* *html form*

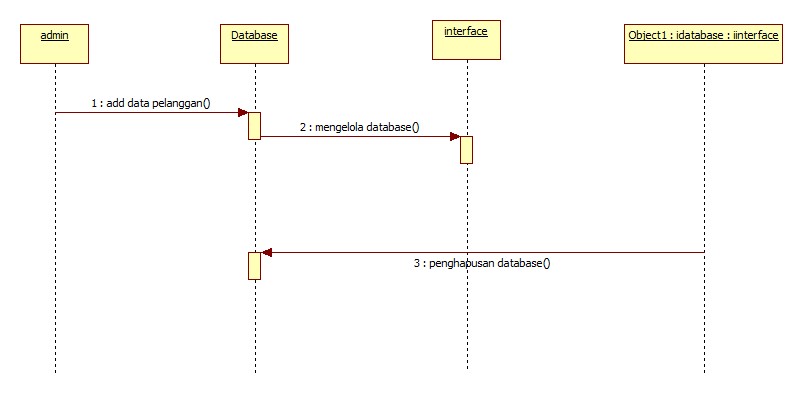
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Elemen Lojik WAE* | *Stereotype* |
|  |  | *Server Page* |
|  |  | *Server Page* |
|  |  | *Client Page* |

#### Identifikasi Kelas Perancangan

Buat daftar kelas analisis yang merealisasikan use case dalam tabel berikut:

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



#### Diagram Kelas Perancangan

Buatlah diagram kelas perancangan lengkap dengan Elemen WAE-Logical View (jika berbasis web).

### Use Case <ubah data>

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### Identifikasi Elemen WAE - Logical View

Bagian ini diisi dengan hasil identifikasi elemen WAE-Logical View yang merealisasikan use case:

* *server page*
* *client page*
* *html form*

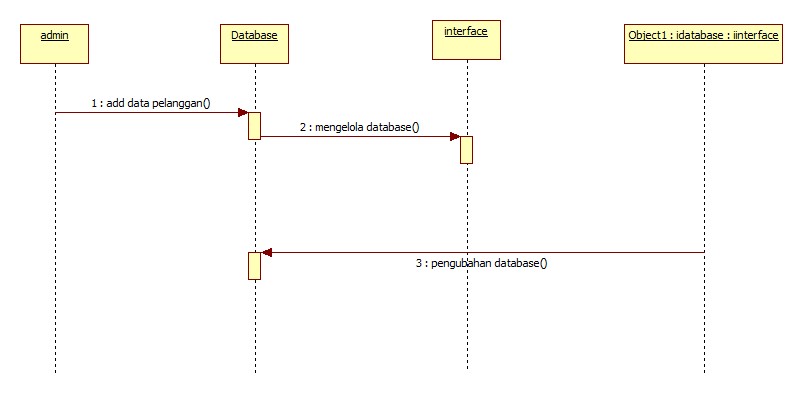
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Elemen Lojik WAE* | *Stereotype* |
|  |  | *Server Page* |
|  |  | *Server Page* |
|  |  | *Client Page* |

#### Identifikasi Kelas Perancangan

Buat daftar kelas analisis yang merealisasikan use case dalam tabel berikut:

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



#### Diagram Kelas Perancangan

Buatlah diagram kelas perancangan lengkap dengan Elemen WAE-Logical View (jika berbasis web).

### Use Case <simpan data>

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### Identifikasi Elemen WAE - Logical View

Bagian ini diisi dengan hasil identifikasi elemen WAE-Logical View yang merealisasikan use case:

* *server page*
* *client page*
* *html form*

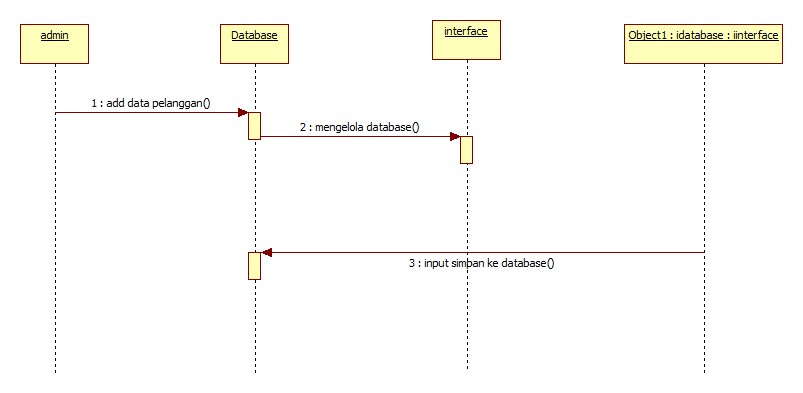
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Elemen Lojik WAE* | *Stereotype* |
|  |  | *Server Page* |
|  |  | *Server Page* |
|  |  | *Client Page* |

#### Identifikasi Kelas Perancangan

Buat daftar kelas analisis yang merealisasikan use case dalam tabel berikut:

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram



#### Diagram Kelas Perancangan

Buatlah diagram kelas perancangan lengkap dengan Elemen WAE-Logical View (jika berbasis web).

## Perancangan Detil Elemen Logical View

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Elemen Lojik WAE* | *Stereotype* |
|  |  | *Server Page* |
|  |  | *Server Page* |
|  |  | *Client Page* |

Untuk setiap elemen:

* identifikasi atribut
* identifikasi operasi

### Stereotyped Class <nama elemen logical view>

*Nama Kelas : ……..*

*Jenis : Server Page / Client Page / HTML Form*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Diisi dengan signature operasi* |  |  |
|  |  |  |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Diisi dengan nama atribut* |  | *Tuliskan tipenya sesuai dengan yang dikenal pada bahasa pemrograman yang digunakan* |
|  |  |  |
|  |  |  |

### Stereotyped Class <nama elemen logical view>

*Nama Kelas : ……..*

*Jenis : Server Page / Client Page / HTML Form*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Diisi dengan signature operasi* |  |  |
|  |  |  |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Diisi dengan nama atribut* |  | *Tuliskan tipenya sesuai dengan yang dikenal pada bahasa pemrograman yang digunakan* |
|  |  |  |
|  |  |  |

## Perancangan Detil Kelas

Bagian ini diisi dengan daftar kelas perancangan dalam tabel berikut:

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

Untuk setiap kelas:

* identifikasi operasi (mengacu pada tanggung-jawab kelas), termasuk visibility-nya
* identifikasi atribut, termasuk visibility-nya

### Kelas <Form Login>

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

*Nama Kelas : FormLogin*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Login* | *Public void* | *Login system* |
| *Reset* | *Public void* | *Reset text login* |
| *Exit* | *Public void* | *Keluar system* |
| *Koneksi login* | *Public void* | *Validasi login* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *username* | *private* | *string* |
| *password* | *private* | *string* |
|  |  |  |

### Kelas <FormUtama>

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

*Nama Kelas: FormUtama*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Exit()* | *Public void* | *Keluar system* |
| *Fpub.show()* | *Public void* | *Form untuk pendaftaran user* |
| *Fpm.show()* | *Public void* | *Form untuk pendaftaran member* |
| *Fpp.show()* | *Public void* | *Form untuk pembayaran user* |
| *Fpm.show()* | *Public void* | *Form untuk pembayaran member* |
| *Frpm.show()* | *Public void* | *Form untuk report pembayaran member* |
| *Frpp.show()* | *Public void* | *Form untuk report pembayaran user* |
| *k.show()* | *Public void* | *Form database* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |

### Kelas <FormDaftarMainMember>

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

*Nama Kelas :* FormDaftarMainMember

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *BindDatabase()* | *Public void* | *Panggil untuk mencetak database ke system* |
| *Fillcombo()* | *Public void* | *Memanggil combo text di database* |
| *Conn.Open()* | *Public void* | *Buka koneksi* |
| *Conn.Close()* | *Public void* | *Tutp koneksi* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Id\_main* | *private* | *int* |
| *Nama* | *private* | *string* |
| *Alamat* | *Private* | *string* |
| *Jenis\_sewa* | *Private* | *string* |
| *Waktu\_masuk* | *Private* | *string* |
| *Notv* | *Private* | *string* |

### Kelas <FormPendaftaranUserBiasa>

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

*Nama Kelas :* FormPendaftaranUserBiasa

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *BindDatabase()* | *Public void* | *Panggil untuk mencetak database ke system* |
| *ClearForm()* | *Public void* | *Membersihkan form database* |
| *Conn.Open()* | *Public void* | *Buka koneksi* |
| *Conn.Close()* | *Public void* | *Tutp koneksi* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Nama* | *private* | *string* |
| *Alamat* | *Private* | *string* |
| *Jenis\_sewa* | *Private* | *string* |
| *Waktu\_masuk* | *Private* | *string* |
| *Notv* | *Private* | *string* |

### Kelas <FormPembayaranMember>

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

*Nama Kelas :* FormPembayaranMember

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *munculpanelPS()* | *Public void* | *Panggil dari panel database ke system* |
| *Fillcombo()* | *Public void* | *Memanggil combo text di database* |
| *Conn.Open()* | *Public void* | *Buka koneksi* |
| *Conn.Close()* | *Public void* | *Tutp koneksi* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Id\_transaksi* | *private* | *int* |
| *Id\_member* | *private* | *int* |
| *Nama* | *Private* | *string* |
| *Alamat* | *Private* | *string* |
| *Jenis\_sewa* | *Private* | *string* |
| *Waktu\_masuk* | *Private* | *string* |
| *biaya* | *Private* | *int* |

### Kelas <FormPembayaranPersonal>

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

*Nama Kelas : FormLogin*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *munculpanelPS()* | *Public void* | *Panggil dari panel database ke system* |
| *Fillcombo()* | *Public void* | *Memanggil combo text di database* |
| *Conn.Open()* | *Public void* | *Buka koneksi* |
| *Conn.Close()* | *Public void* | *Tutp koneksi* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Id\_transaksi* | *private* | *int* |
| *Id\_user* | *private* | *int* |
| *Nama* | *Private* | *string* |
| *Alamat* | *Private* | *string* |
| *Jenis\_sewa* | *Private* | *string* |
| *Waktu\_masuk* | *Private* | *string* |
| *biaya* | *Private* | *int* |

### Kelas <FormKeterangan>

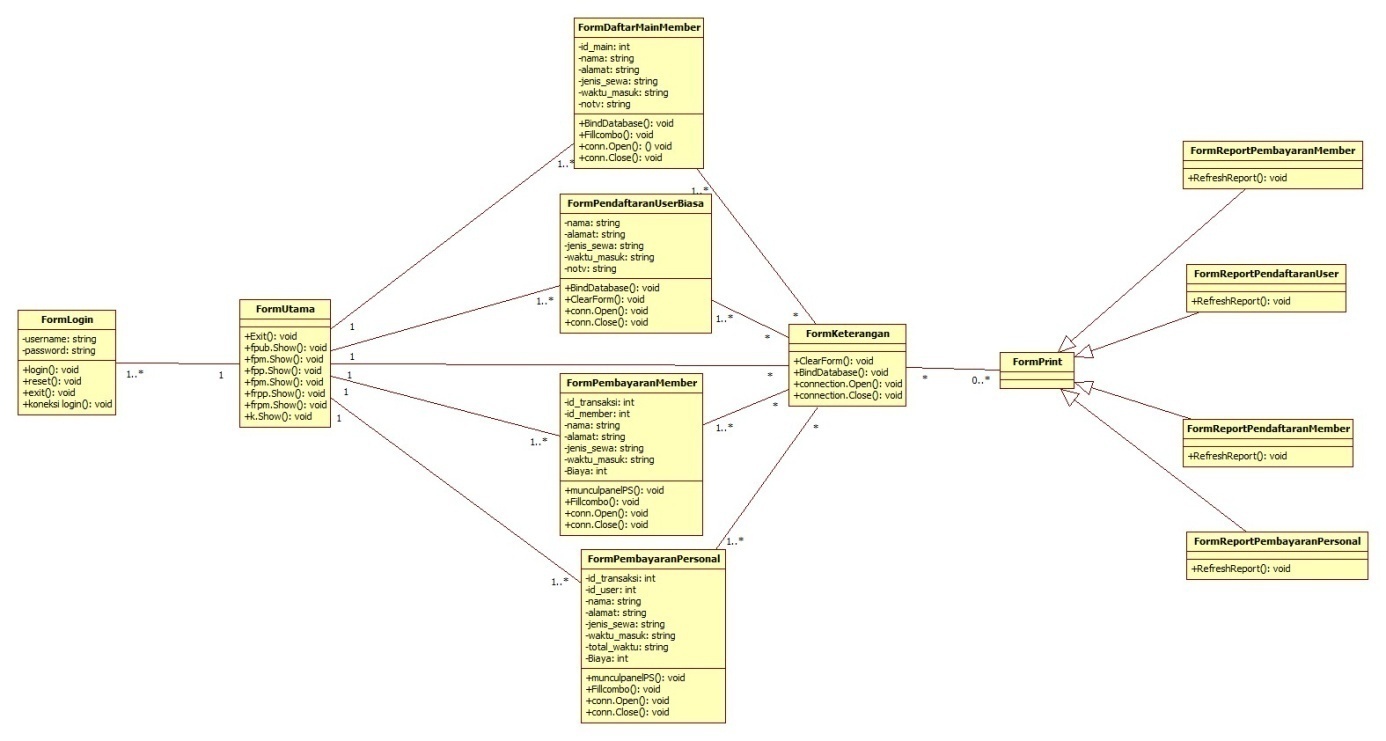
Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

*Nama Kelas :* FormKeterangan

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *BindDatabase()* | *Public void* | *Panggil untuk mencetak database ke system* |
| *ClearForm()* | *Public void* | *Membersihkan form database* |
| *Connection.Open()* | *Public void* | *Buka koneksi* |
| *Connection.Close()* | *Public void* | *Tutp koneksi* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |

## Diagram Kelas Keseluruhan

Bagian ini diisi dengan diagram kelas keseluruhan.



## Algoritma/Query

Bagian ini hanya diisi untuk kerangka algoritma untuk proses-proses yang dianggap cukup penting. Implementasi skeleton code juga sudah dapat dilakukan untuk kelas-kelas yang terdefinisi pada bahasa pemrograman tertentu. Boleh dibuat subbab per kelas.

Contoh:

*Nama Kelas : Form Login*

*Nama Operasi : Login*

*Algoritma : (Algo-xxx)*

*"SELECT \* FROM TAdmin where username ='" + username + "' and password ='" + password + "'"*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form Keterangan*

*Nama Operasi : Select*

*Algoritma : (Algo-xxx)*

*"SELECT \* FROM TBayarmember JOIN TBayarpersonal ON TBayarmember.nama=TBayarpersonal.nama" +*

*"JOIN TDaftarmember JOIN TDaftaruser ON TDaftarmember.alamat = TDaftaruser.alamat";*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form Keterangan*

*Nama Operasi : Delete*

*Algoritma : (Algo-xxx)*

*"DELETE \* FROM TBayarmember JOIN TBayarpersonal ON TBayarmember.nama=TBayarpersonal.nama" +*

*"JOIN TDaftarmember JOIN TDaftaruser ON TDaftarmember.alamat = TDaftaruser.alamat WHERE kode = id\_transaksi"*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form Keterangan*

*Nama Operasi : Update*

*Algoritma : (Algo-xxx)*

*"UPDATE \* FROM TBayarmember JOIN TBayarpersonal ON TBayarmember.nama=TBayarpersonal.nama" +*

*"JOIN TDaftarmember JOIN TDaftaruser ON TDaftarmember.alamat = TDaftaruser.alamat WHERE kode = id\_transaksi"*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form Keterangan*

*Nama Operasi : Pencarian*

*Algoritma : (Algo-xxx)*

*"SELECT \* FROM TBayarmember JOIN TBayarpersonal ON TBayarmember.nama=TBayarpersonal.nama" +*

*"JOIN TDaftarmember JOIN TDaftaruser ON TDaftarmember.alamat = TDaftaruser.alamat" +*

*" WHERE nama LIKE @merk"*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form Daftar Main Member*

*Nama Operasi : Select Data*

*Algoritma : (Algo-xxx)*

*"SELECT \* FROM TMainmember"*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form Daftar Main Member*

*Nama Operasi : Select id*

*Algoritma : (Algo-xxx)*

*"SELECT nama,alamat FROM TDaftarmember WHERE id\_member = " + comboIdMember.Text + ""*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |
|  |  |  |

*Nama Kelas : Form Daftar Main Member*

*Nama Operasi : Submit*

*Algoritma : (Algo-xxx)*

*"INSERT INTO TMainmember (nama, alamat, jenis\_sewa, waktu\_masuk, notv) values (@nama, @alamat, @jenis, @waktu, @noTV)"*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form Pembayaran Personal*

*Nama Operasi : Submit*

*Algoritma : (Algo-xxx)*

*"INSERT INTO TBayarpersonal (nama, alamat, jenis\_sewa, total\_waktu, biaya) values (@nama, @alamat, @jenis\_sewa, @totwak, @totbiaya)"*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form Pembayaran Personal*

*Nama Operasi : Select id*

*Algoritma : (Algo-xxx)*

*"SELECT nama,alamat,jenis\_sewa,waktu\_masuk,notv FROM TDaftaruser WHERE id\_user = " + comboId.Text + ""*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form pendaftaran member*

*Nama Operasi : Select id*

*Algoritma : (Algo-xxx)*

*"SELECT nama,alamat,jenis\_sewa,waktu\_masuk,notv FROM TDaftaruser WHERE id\_user = " + comboId.Text + ""*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form pendaftaran member*

*Nama Operasi : Daftar*

*Algoritma : (Algo-xxx)*

*"INSERT INTO TDaftarmember (nama, alamat, nope, tanggal\_daftar, jenis\_member) values (@nama, @alamat, @nope, @tanggal, @jenis)"*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form pendaftaran user biasa*

*Nama Operasi : Select*

*Algoritma : (Algo-xxx)*

*"SELECT \* FROM TDaftaruser"*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

*Nama Kelas : Form Pendaftaran user biasa*

*Nama Operasi : Submit*

*Algoritma : (Algo-xxx)*

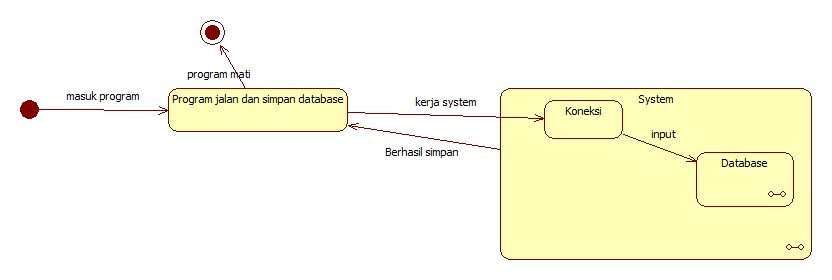
*"INSERT INTO TDaftaruser (nama, alamat, jenis\_sewa, notv, waktu\_masuk) values (@nama, @alamat, @jenis, @notv, @waktu)"*

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Untuk* |
|  |  |  |
|  |  |  |

## Diagram Statechart

.

## Perancangan Antarmuka

Pada fase Inception

1. Perancangan Antarmuka Login



Gambar 4. 1 Form Login

1. Perancangan Antarmuka Pesan Gagal Login



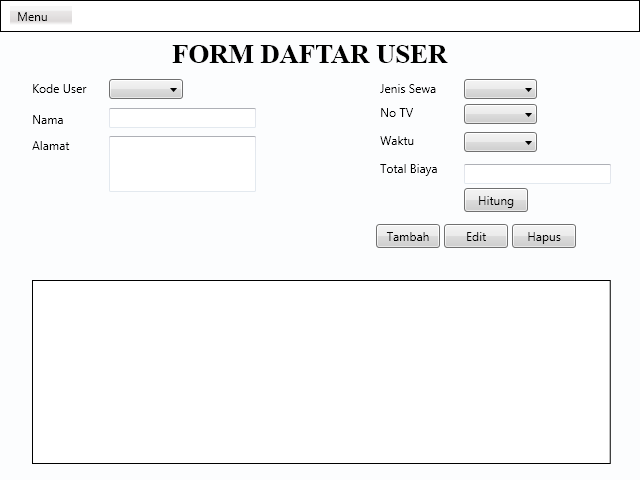
Gambar 4. 2 Pesan Gagal Login

1. Perancangan Antarmuka Pendaftaran Member



Gambar 4. 3 Form Pendaftaran Member

1. Perancangan Antarmuka Pendaftaran User



Gambar 4. 4 Form Daftar User

1. Perancangan Antarmuka Pesan Input Data User



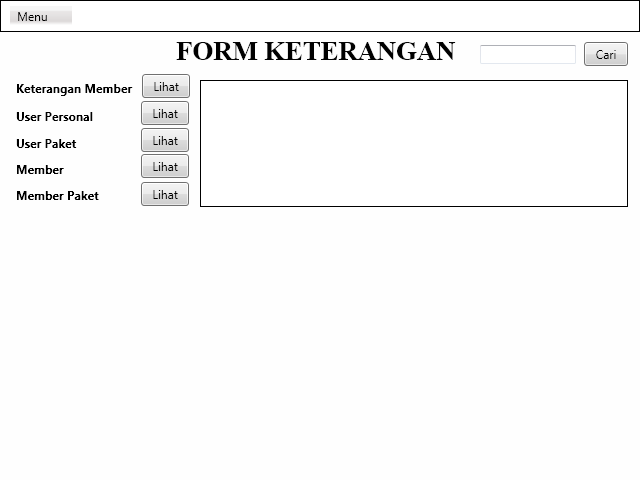
Gambar 4. 5 Pesan Input Data User Berhasil.

1. Perancangan Antarmuka Pesan Hapus Data User



Gambar 4. 6 Pesan Data User Berhasil dihapus

1. Perancangan Antarmuka Form Keterangan



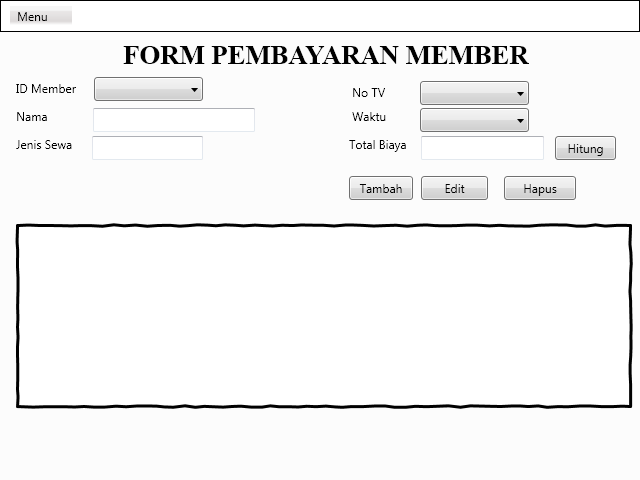
Gambar 4. 7 Form Keterangan

1. Perancangan Antarmuka Halaman Menu Utama



Gambar 4. 8 Form Menu Utama

1. Perancangan Antarmuka Pembayaran Member



Gambar 4. 9 Form Pembayaran

Bagian ini diisi dengan versi awal prototipe antarmuka.

Pada fase Elaboration:

*Antarmuka : Gambar 4..10 Form Login*

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Button1* | *Button* | *LOGIN* | Button Login berfungsi untuk masuk ke menu utama. |
| *Button2* | Button | RESET | Button Reset berfungsi untuk mengembalikan field ke keadaan semula atau awal |
| *Button 3* | *Button* | *EXIT* | Button Exit berfungsi untuk keluar dari program atau aplikasi |
| *TexsBox1* | *TexsBox* | *Text User* | Berfungsi untuk memasukan username untuk login |
| *TexsBox2* | *TexsBox* | *TextPass* | Berfungsi untuk memasukan password |

*Antarmuka : Gambar 4..11 Form Menu Utama*

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *MenuStrip1* | *MenuStrip* | *Menu Item* | Berfungsi untuk masuk ke database, log out dari aplikasi, dan keluar dari aplikasi |
| *MenuStrip2* | MenuStrip | Daftar | Untuk mengisi pendataran personal atau member |
| *MenuStrip3* | *MenuStrip* | *Pembayaran* | Untuk mengetahui total pembayaran personal atau member |
| *MenuStrip4* | *MenuStrip* | *Report* | Untuk mencatatkan report pembayaran personal atau member |

*Antarmuka : Gambar 4..12 Form Keterangan*

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Button1* | *Button* | *Daftar Member* | Untar melihat daftar member |
| *Button2* | Button | Daftar User Personal | Untuk melihat daftar user personal |
| *Button3* | *Button* | *Pembayaran personal* | Untuk melihat pembayaran personal |
| *Button4* | *Button* | *Pembayaran Member* | Untuk melihat pembayaran member |

*Antarmuka : Gambar 4..13 Form Pendaftaran User Personal*

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Button1* | *Button* | *Set* | Untuk mengeset waktu mulai |
| *Button2* | Button | Submit | Untuk mengirimkan data yang telah di input |
| *Data gridview1* | *datagridview* | *Datagridview member* | Untuk melihat pembayaran personal |
| *Button3* | *Button* | *Batal* | Untuk membatallkan proses |

*Antarmuka : Gambar 4..14 Form Pendaftaran Member*

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Button1* | *Button* | *Set* | Untuk mengeset tanggal pendaftaran |
| *Button2* | Button | Daftar | Untuk mengirimkan data yang telah di input |
| *Data gridview1* | *datagridview* | *Datagridview member* | Untuk melihat data pendaftaran |
| *Button3* | *Button* | *Batal* | Untuk membatallkan proses |

*Antarmuka : Gambar 4..15 Form Pembayaran Personal*

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Button1* | *Button* | *Set* | Untuk mengeset waktu selesai |
| *Button2* | Button | Hitung | Untuk menghitung total waktu dan biaya personal |
| *Button3* | *Button* | *submit* | Untuk mengirim biaya yang telah di input |
| *Button3* | *Button* | *Batal* | Untuk membatallkan proses |

*Antarmuka : Gambar 4..16 Form Pembayaran Member*

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Button1* | *Button* | *Set* | Untuk mengeset waktu selesai |
| *Button2* | Button | Hitung | Untuk menghitung total waktu dan biaya member |
| *Button3* | *Button* | *submit* | Untuk mengirim biaya yang telah di input |
| *Button3* | *Button* | *Batal* | Untuk membatallkan proses |

*Antarmuka : Gambar 4..17 Form Report Pembayaran Personal*

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Reportviewer1* | *Reportviewer* | *Report pembayaran personal* | Untuk menampilkan report pembayaran seluruh personal |

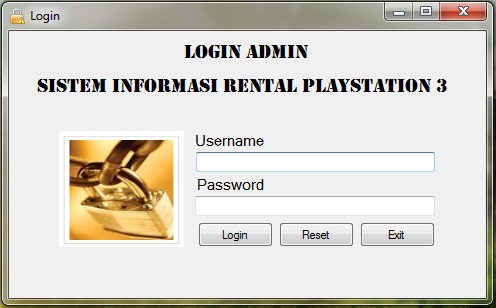
*Antarmuka : Gambar 4..18 Form Report Pembayaran Member*

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Reportviewer1* | *Reportviewer* | *Report pembayaran member* | Untuk menampilkan report pembayaran seluruh member |

Jika objek dikaitkan ke File lain (misalnya file gambar, file teks), berikan nama file terkait dan deskripsi ringkas dalam kolom keterangan

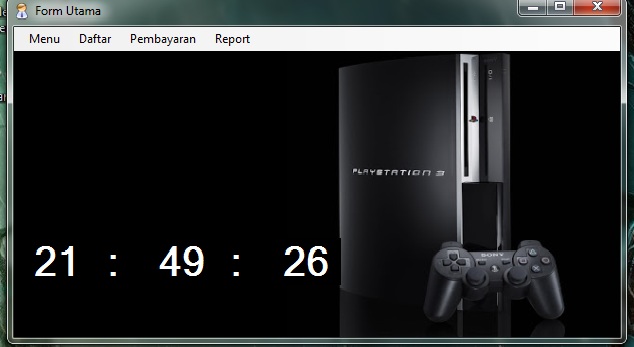
Pada fase Construction:

1. Perancangan Antarmuka Login



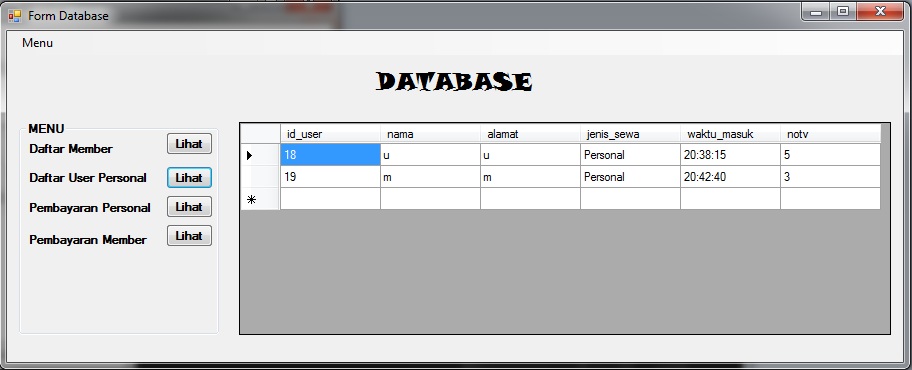
Gambar 4. 10 Form Login

1. Perancangan Antarmuka Menu Utama



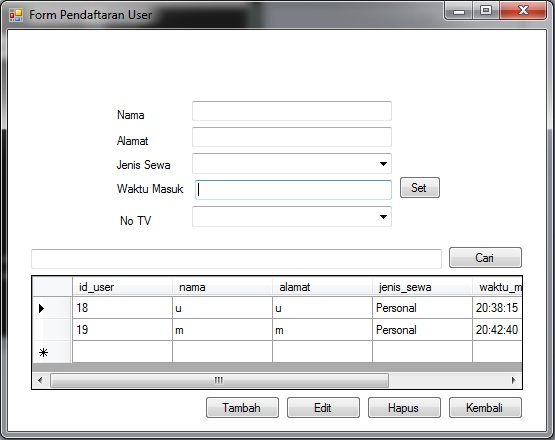
Gambar 4. 11 Form Menu Utama

1. Perancangan Antarmuka Keterangan



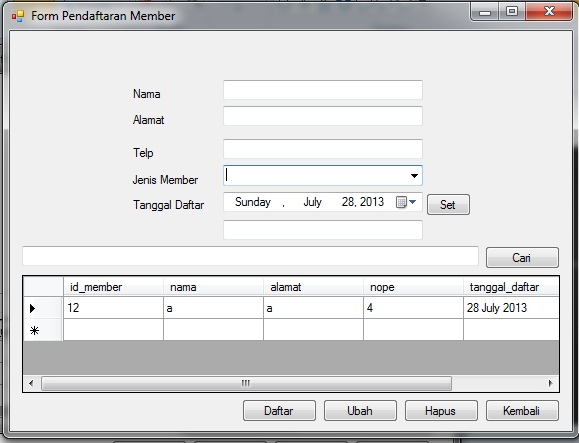
Gambar 4. 12 Form Keterangan

1. Perancangan Antarmuka Pendaftaran User



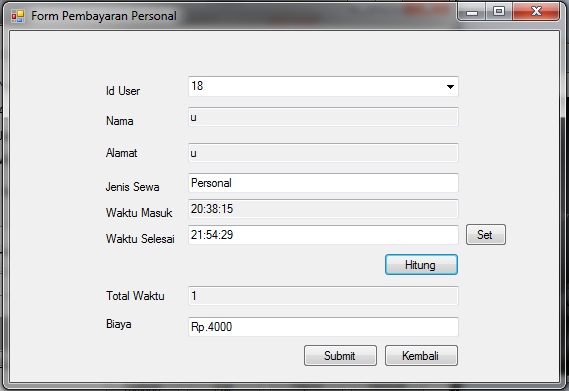
Gambar 4. 13 Form Pendaftaran User Personal

1. Perancangan Antarmuka Pendaftaran Member



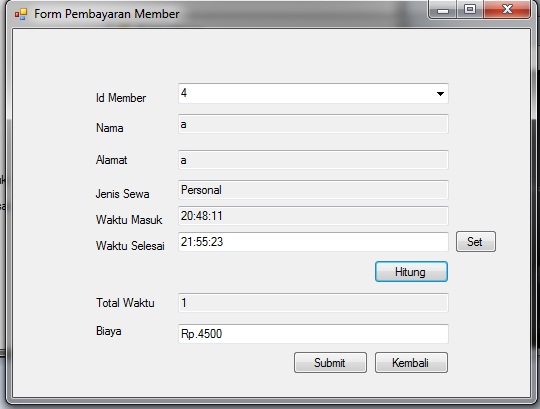
Gambar 4. 14 Form Pendaftaran Member

1. Perancangan Antarmuka Form Pembayaran Personal



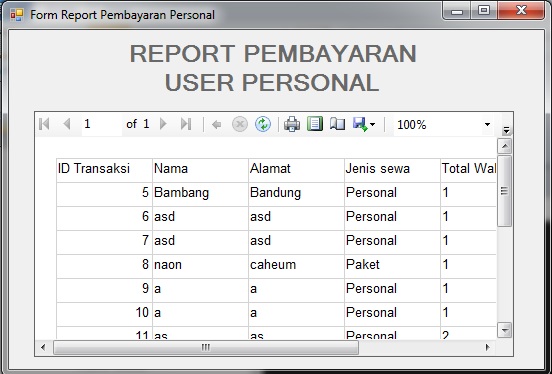
Gambar 4. 15 Form Pembayaran Personal

1. Perancangan Antarmuka Pembayaran Member



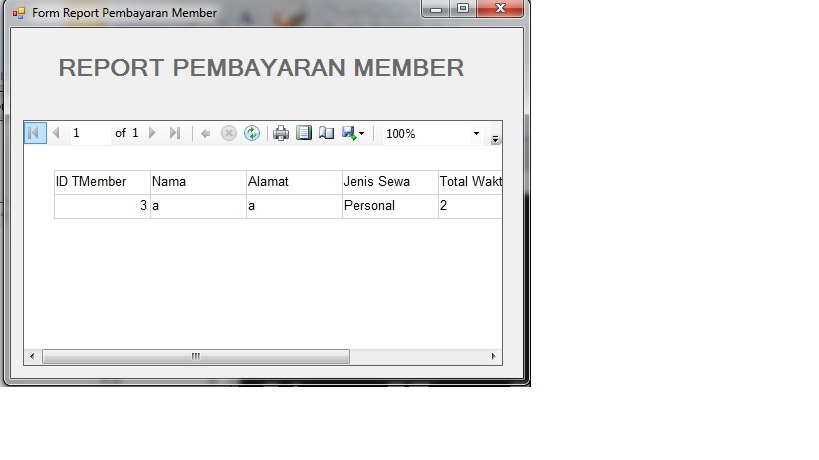
Gambar 4. 16 Form Pembayaran Member

1. Perancangan Antarmuka Report Pembayaran User Personal



Gambar 4. 17 Form Report Pembayaran Personal

1. Perancangan Antarmuka Report Pembayaran Member



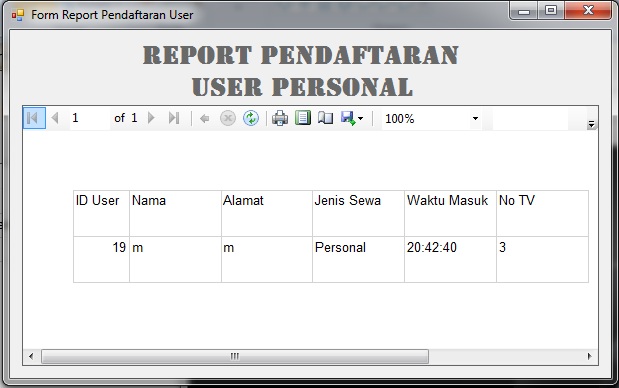
Gambar 4. 18 Form Report Pembayaran Member

1. Perancangan Antarmuka Report Pendaftaran Member



Gambar 4. 19 Form Report Pembayaran Member

1. Perancangan Antarmuka Report Pendaftaran User



Gambar 4. 20 Form Report Pembayaran Member

## Perancangan Representasi Persistensi Kelas

Bagian ini diisi dengan rancangan skema basisdata dan traceability-nya terhadap kelas entity.

## Coding Standard dan Naming Convention

Pada fase Inception:

Bagian ini belum diisi.

Pada fase Elaboration:

*Nama Form Koding : Daftar Member*

*Algoritma : (Algo-xxx)*

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Windows.Forms;*

*using System.Configuration;*

*using System.Data.SqlClient;*

*namespace ProgramRentalPS*

*{*

*public partial class FormMember : Form*

*{*

*private string strConnection = ConfigurationManager.ConnectionStrings["DbRentalConnectionString"].ConnectionString;*

*private Member member;*

*private string tanggal\_daftar;*

*private string tanggal\_berakhir;*

*private string id\_member;*

*public FormMember()*

*{*

*InitializeComponent();*

*member = new Member();*

*}*

*private void ClearForm()*

*{*

*textPass.Clear();*

*textNama.Clear();*

*textTelp.Clear();*

*textDaftar.Clear();*

*textAkhir.Clear();*

*textmasa.Clear();*

*richTextAlamat.Clear();*

*}*

*private void BindDatabase()*

*{*

*SqlConnection conn = new SqlConnection(strConnection);*

*using (conn)*

*{*

*conn.Open();*

*string select = "SELECT \* FROM tmember";*

*SqlCommand cmd = new SqlCommand(select, conn);*

*SqlDataReader reader = cmd.ExecuteReader();*

*DataTable table = new DataTable();*

*table.Load(reader);*

*dataGridMember.DataSource = table;*

*conn.Close();*

*}*

*}*

*private void backToolStripMenuItem\_Click(object sender, EventArgs e)*

*{*

*this.Close();*

*}*

*private void exitToolStripMenuItem\_Click(object sender, EventArgs e)*

*{*

*Application.Exit();*

*}*

*private void button3\_Click(object sender, EventArgs e)*

*{*

*DateTime dt = dateTimeAwal.Value;*

*string strDate = dt.ToString("dd MMMM yyyy");*

*textDaftar.Text = strDate;*

*}*

*private void button4\_Click(object sender, EventArgs e)*

*{*

*DateTime dt = dateTimeAkhir.Value;*

*string strDate = dt.ToString("dd MMMM yyyy");*

*textAkhir.Text = strDate;*

*}*

*private void button5\_Click(object sender, EventArgs e)*

*{*

*DateTime dt1 = dateTimeAwal.Value;*

*DateTime dt2 = dateTimeAkhir.Value;*

*TimeSpan ts = dt2 - dt1;*

*string days = ts.Days.ToString();*

*textmasa.Text = days;*

*}*

*private void FormMember\_Load(object sender, EventArgs e)*

*{*

*BindDatabase();*

*}*

*private void radioButton1\_CheckedChanged(object sender, EventArgs e)*

*{*

*member.jenis\_kelamin = "Pria";*

*}*

*private void button7\_Click(object sender, EventArgs e)*

*{*

*string jenis\_kelamin = radioButton1.Text;*

*radioButton1.Text = member.GetSummary();*

*}*

*private void radioButton2\_CheckedChanged(object sender, EventArgs e)*

*{*

*member.jenis\_kelamin = "Wanita";*

*}*

*private void button1\_Click(object sender, EventArgs e)*

*{*

*string jenis\_kelamin = radioButton2.Text;*

*radioButton2.Text = member.GetSummary();*

*string password = textPass.Text;*

*string nama = textNama.Text;*

*string jenis\_Kelamin = radioButton2.Text;*

*string nope = textTelp.Text;*

*tanggal\_daftar = textDaftar.Text;*

*tanggal\_berakhir = textAkhir.Text;*

*string masa\_aktif = textmasa.Text;*

*string alamat = richTextAlamat.Text;*

*SqlConnection conn = new SqlConnection(strConnection);*

*using (conn)*

*{*

*conn.Open();*

*string insert = "INSERT INTO TMember (password, nama, jenis\_kelamin, nope, tanggal\_daftar, tanggal\_berakhir, masa\_aktif, alamat ) values (@password, @nama, @jenis\_kelamin, @nope, @tanggal\_daftar, @tanggal\_berakhir, @masa\_aktif, @alamat)";*

*SqlCommand cmd = new SqlCommand(insert, conn);*

*cmd.Parameters.AddWithValue("@password", password);*

*cmd.Parameters.AddWithValue("@nama", nama);*

*cmd.Parameters.AddWithValue("@jenis\_kelamin", jenis\_Kelamin);*

*cmd.Parameters.AddWithValue("@nope", nope);*

*cmd.Parameters.AddWithValue("@tanggal\_daftar", tanggal\_daftar);*

*cmd.Parameters.AddWithValue("@tanggal\_berakhir", tanggal\_berakhir);*

*cmd.Parameters.AddWithValue("@masa\_aktif", masa\_aktif);*

*cmd.Parameters.AddWithValue("@alamat", alamat);*

*int insertResult = cmd.ExecuteNonQuery();*

*if (insertResult == 1)*

*{*

*MessageBox.Show("Sukses Menambah Data User");*

*}*

*else*

*{*

*MessageBox.Show("Gagal Menambah Data User");*

*}*

*conn.Close();*

*BindDatabase();*

*}*

*}*

*private void button2\_Click(object sender, EventArgs e)*

*{*

*textPass.ResetText();*

*textNama.ResetText();*

*radioButton1.ResetText();*

*textTelp.ResetText();*

*textDaftar.ResetText();*

*textAkhir.ResetText();*

*textmasa.ResetText();*

*richTextAlamat.ResetText();*

*}*

*private void button6\_Click(object sender, EventArgs e)*

*{*

*string nama = textNama.Text;*

*string title = "Pesan Konfirmasi";*

*string message = "Apakah Anda Yakin " + nama.Trim() + " Akan Dihapus?";*

*DialogResult result = MessageBox.Show(message, title, MessageBoxButtons.OKCancel, MessageBoxIcon.Warning);*

*if (result != DialogResult.OK)*

*{*

*return;*

*}*

*SqlConnection conn = new SqlConnection(strConnection);*

*using (conn)*

*{*

*conn.Open();*

*string insert = "DELETE FROM tmember WHERE id\_member = @id";*

*SqlCommand cmd = new SqlCommand(insert, conn);*

*cmd.Parameters.AddWithValue("@id", id\_member);*

*int insertResult = cmd.ExecuteNonQuery();*

*if (insertResult == 1)*

*{*

*MessageBox.Show("Sukses Menghapus Data User");*

*}*

*else*

*{*

*MessageBox.Show("Gagal Menghapus Data User");*

*}*

*conn.Close();*

*BindDatabase();*

*}*

*}*

*private void dataGridMember\_SelectionChanged(object sender, EventArgs e)*

*{*

*foreach (DataGridViewRow row in dataGridMember.SelectedRows)*

*{*

*id\_member = row.Cells[0].Value.ToString();*

*string password = row.Cells[1].Value.ToString();*

*string nama = row.Cells[2].Value.ToString();*

*string jenis\_kelamin = row.Cells[3].Value.ToString();*

*string nope = row.Cells[4].Value.ToString();*

*string tanggal\_daftar = row.Cells[5].Value.ToString();*

*string tanggal\_berakhir = row.Cells[6].Value.ToString();*

*string massa\_aktif = row.Cells[7].Value.ToString();*

*string alamat = row.Cells[8].Value.ToString();*

*textPass.Text = password;*

*textNama.Text = nama;*

*radioButton1.Text = jenis\_kelamin;*

*textTelp.Text = nope;*

*textDaftar.Text = tanggal\_daftar;*

*textAkhir.Text = tanggal\_berakhir;*

*textmasa.Text = massa\_aktif;*

*richTextAlamat.Text = alamat;*

*}*

*}*

*}*

*}*

Pada fase Construction:

*Nama Form Koding : Daftar Member*

*Algoritma : (Algo-xxx)*

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Windows.Forms;*

*using System.Configuration;*

*using System.Data.SqlClient;*

*namespace ProgramRentalPS*

*{*

*public partial class FormPendaftaranMember : Form*

*{*

*SqlConnection conn = koneksi.getKoneksi();*

*private string id;*

*public FormPendaftaranMember()*

*{*

*InitializeComponent();*

*}*

*private void ClearForm()*

*{*

*textNama.Clear();*

*textAlamat.Clear();*

*}*

*private void BindDatabase()*

*{*

*SqlConnection conn = koneksi.getKoneksi();*

*using (conn)*

*{*

*conn.Open();*

*string select = "SELECT \* FROM TDaftarmember";*

*SqlCommand cmd = new SqlCommand(select, conn);*

*SqlDataReader reader = cmd.ExecuteReader();*

*DataTable table = new DataTable();*

*table.Load(reader);*

*dataGridMember.DataSource = table;*

*dataGridMember.ClearSelection();*

*ClearForm();*

*conn.Close();*

*}*

*}*

*private void button1\_Click(object sender, EventArgs e)*

*{*

*string nama = textNama.Text;*

*string nope = textTelp.Text;*

*string alamat = textAlamat.Text;*

*string jenis = comboJenisMember.Text;*

*string tanggal = textTanggalDaftar.Text;*

*using (conn)*

*{*

*conn.Open();*

*string insert = "INSERT INTO TDaftarmember (nama, alamat, nope, tanggal\_daftar, jenis\_member) values (@nama, @alamat, @nope, @tanggal, @jenis)";*

*SqlCommand cmd = new SqlCommand(insert, conn);*

*cmd.Parameters.AddWithValue("@nama", nama);*

*cmd.Parameters.AddWithValue("@alamat", alamat);*

*cmd.Parameters.AddWithValue("@nope", nope);*

*cmd.Parameters.AddWithValue("@tanggal", tanggal);*

*cmd.Parameters.AddWithValue("@jenis", jenis);*

*int insertResult = cmd.ExecuteNonQuery();*

*if (insertResult == 1)*

*{*

*MessageBox.Show("Sukses Menambah Data Member");*

*}*

*else*

*{*

*MessageBox.Show("Gagal Menambah Data Member");*

*}*

*conn.Close();*

*BindDatabase();*

*}*

*}*

*private void FormPendaftaranMember\_Load(object sender, EventArgs e)*

*{*

*BindDatabase();*

*}*

*private void button3\_Click(object sender, EventArgs e)*

*{*

*DateTime dt = dateTanggalDaftar.Value;*

*string strDate = dt.ToString("dd MMMM yyyy");*

*textTanggalDaftar.Text = strDate;*

*}*

*private void button2\_Click(object sender, EventArgs e)*

*{*

*this.Close();*

*}*

*private void button5\_Click(object sender, EventArgs e)*

*{*

*string nama = textNama.Text;*

*string alamat = textAlamat.Text;*

*string nope = textTelp.Text;*

*string tgl = textTanggalDaftar.Text;*

*string jenis = comboJenisMember.Text;*

*SqlConnection connection = koneksi.getKoneksi();*

*using (connection)*

*{*

*connection.Open();*

*string insert = "UPDATE TDaftarmember SET nama = @nama, alamat = @alamat, nope = @nope, tanggal\_daftar = @tanggal\_daftar, jenis\_member = @jenis\_member WHERE id\_member = @idm";*

*SqlCommand cmd = new SqlCommand(insert, connection);*

*cmd.Parameters.AddWithValue("@idm", id);*

*cmd.Parameters.AddWithValue("@nama", nama);*

*cmd.Parameters.AddWithValue("@alamat", alamat);*

*cmd.Parameters.AddWithValue("@nope", nope);*

*cmd.Parameters.AddWithValue("@tanggal\_daftar", tgl);*

*cmd.Parameters.AddWithValue("@jenis\_member", jenis);*

*int insertResult = cmd.ExecuteNonQuery();*

*if (insertResult == 1)*

*{*

*MessageBox.Show("Sukses Mengubah Data");*

*}*

*else*

*{*

*MessageBox.Show("Gagal Mengubah Data");*

*}*

*connection.Close();*

*BindDatabase();*

*}*

*}*

*private void dataGridMember\_SelectionChanged(object sender, EventArgs e)*

*{*

*foreach (DataGridViewRow row in dataGridMember.SelectedRows)*

*{*

*id = row.Cells[0].Value.ToString();*

*string nama = row.Cells[1].Value.ToString();*

*string alamat = row.Cells[2].Value.ToString();*

*string nope = row.Cells[3].Value.ToString();*

*string tgl = row.Cells[4].Value.ToString();*

*string jenis = row.Cells[5].Value.ToString();*

*textNama.Text = nama;*

*textAlamat.Text = alamat;*

*textTelp.Text = nope;*

*textTanggalDaftar.Text = tgl;*

*comboJenisMember.Text = jenis;*

*}*

*}*

*private void button4\_Click(object sender, EventArgs e)*

*{*

*string nama = textNama.Text;*

*string title = "Pesan Konfirmasi";*

*string message = "Anda yakin " + nama.Trim() + " akan dihapus ?";*

*DialogResult result = MessageBox.Show(message, title, MessageBoxButtons.OKCancel, MessageBoxIcon.Warning);*

*if (result != DialogResult.OK)*

*{*

*return;*

*}*

*SqlConnection connection = koneksi.getKoneksi();*

*using (connection)*

*{*

*connection.Open();*

*string insert = "DELETE FROM TDaftarmember WHERE id\_member = @idm";*

*SqlCommand cmd = new SqlCommand(insert, connection);*

*cmd.Parameters.AddWithValue("@idm", id);*

*int insertResult = cmd.ExecuteNonQuery();*

*if (insertResult == 1)*

*{*

*MessageBox.Show("Sukses Menghapus Data Member");*

*}*

*else*

*{*

*MessageBox.Show("Gagal Menghapus Data Member");*

*}*

*connection.Close();*

*BindDatabase();*

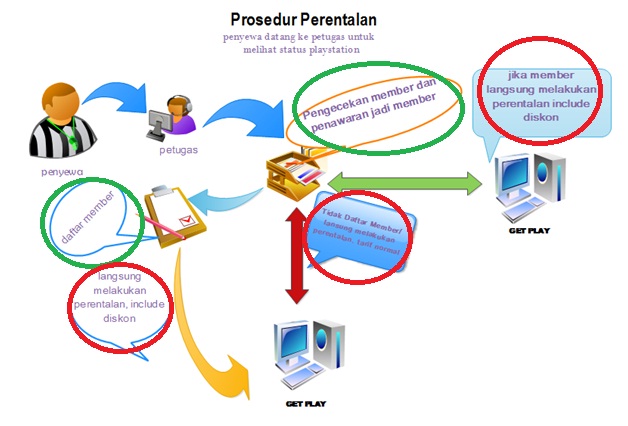
*}*

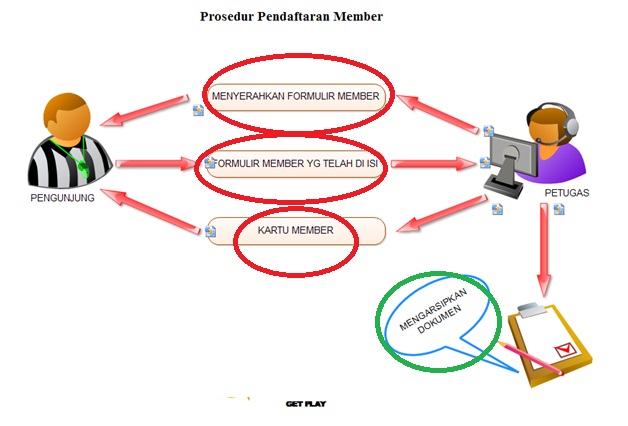
*}*

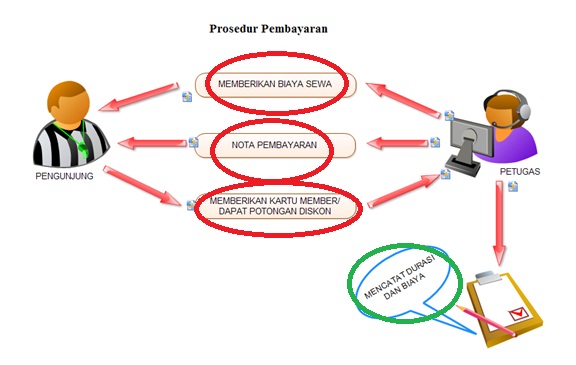
*}*

*}*

## Deployment Diagram







# Implementasi

Hasil yang didapat setelah mengimplementasikan langkah-langkah kerja pada bab sebelumnya adalah hasil analisis kebutuhan, fungsionalitas system informasi Rental PS3, *user* yang menggunakan fungsionalitas tersebut, beserta nilai yang diharapkan dari adanya fungsionalitas yang dapat dilihat pada Tabel 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifikasi | Kebutuhan fungsional | Prosedur Pengujian | Nilai yang di harapkan | Analisis di rental PS dipatiukur |
| *U-1-01* | Pengelolaan database pendaftaran | * Buka Aplikasi rental ps3 oleh admin * Lakukan pengelolaan database pelanggan * Simpan ke database | Tersedia cetak laporan bagi pelanggan yang mendaftar (01) | Baik |
| U-1-02 | Pengelolaan database pembayaran | Tersedia cetak laporan bagi pelanggan yang membayar(02) | Baik |

## Lingkungan Implementasi

Pada pembahasan lingkungan implementasi meliputi pembahasan spesifikasi hardware dan software yang digunakan untuk membuat sistem aplikasi rental plasystation.

Sistem aplikasi rental playstation ini dibuat dengan perangkat keras dengan spesifikasi sebagai berikut :

### Hardware

Sistem aplikasi rental playstation dibuat dengan perangkat keras dengan spesifikasi sebagai berikut :

System Model : K42JY

Bios: Ver 08.00.10

Processor: Intel (R) Pentium (R) CPU

Memory :2048MB RAM

Direct Versi: DirectX 11

Name: AMD Radeon HD 6470M

Chip Type : ATI display adapter (90x6760)

DAC Type : Internal DAC (400mhz)

current Display mode: 1366 x 768 (32 bit)(60Hz).

### Software

Sistem aplikasi rental playstation dibuat dengan perangkat lunak sebagai berikut :

Sistem operasi : Microsoft windows 7 ultimate 32 bit

Perangkat pembangun : Microsoft visual Studio 2010

## Implementasi Kelas

Bagian ini diisi dengan daftar kelas yang TELAH diimplementasikan. Misalnya dalam bentuk tabel berikut:

|  |  |  |  |
| --- | --- | --- | --- |
| *No* | *Nama Kelas* | *Nama File Fisik* | *Nama File Executable* |
| *1* | *Login* | *Login.cs* | *login.class* |
| *2* | *koneksi* | *Koneksi.cs* | *Koneksi.class* |
| *3* | *member* | *Member.cs* | *Member.class* |
| *4* | *Program* | *Program.cs* | *Program.class* |
| *5* | *Report1* | *Report1.rdlc* | *Report 1.class* |
| *6* | *Report2* | *Report2.rdlc* | *Report 3.class* |
| *7* | *Report 3* | *Report 3.rdlc* | *Report 3.class* |
| *8* | *Form utama designer* | *Formutama.designer.cs* | *Formutama.designer.cs* |

## Implementasi Elemen WAE (Component View)

Bagian ini diisi dengan hasil identifikasi elemen WAE-Componen View yang merealisasikan use case:

* *dynamic page*
* *static page*
* *physical root*

Jelaskan juga hubungan realisasinya: elemen component view tersebut merealisasikan elemen logical view yang mana. Gunakan tabel jika perlu.

## Implementasi Antarmuka

Bagian ini diisi dengan daftar implementasi antarmuka. Misalnya dalam bentuk tabel berikut:

|  |  |  |  |
| --- | --- | --- | --- |
| *No* | *Antarmuka* | *Nama File Fisik* | *Nama File Executable* |
| *1* | *login* | *Login.cs* | *Login.class* |
| *2* | *Form utama* | *Formutama.cs* | *Formutama.class* |
| *3* | *Form daftar main member* | *Formdaftarmainmember.cs* | *Formdaftarmainmember.class* |
| *4* | *Form keterangan* | *FormKeterangan.cs* | *FormKeterangan.class* |
| *5* | *Form pembayaran* | *FormPembayaran.cs* | *FormPembayaran.class* |
| *6* | *Form pembayaran member* | *FormPembayaranMember.cs* | *FormPembayaranMember.class* |
| *7* | *Form pembayaran personal* | *FormPembayaranPersonal.cs* | *FormPembayaran.class* |
| *8* | *Form pendaftaran user biasa* | *FormPendaftaranUserBiasa.cs* | *FormPendaftaranUserBiasa.class* |
| *9* | *Form print* | *FormPrint.cs* | *FormPrint.class* |
| *10* | *Form report pembayaran member* | *FormReportPembayaranMember.cs* | *FormReportPembayaranMember.class* |
| *11* | *Form report pembayaran personal* | *Form Report PembayaranPersonal.cs* | *Form Report PembayaranPersonal.class* |
| *12* | *Form report pendaftaran user* | *FormReportPedaftaranUser.cs* | *FormReportPedaftaranUser.class* |

## File Lain

Bagian ini diisi dengan file-file lain yang dibuat, selain file kelas dan antarmuka, misalnya file utility yang digunakan.

|  |  |  |
| --- | --- | --- |
| *No* | *Nama File* | *Keterangan* |
|  |  |  |
|  |  |  |
|  |  |  |

# Pengujian

## Rencana dan Prosedur Pengujian

### Rencana Pengujian

Bagian ini diisi dengan rencana pengujian, misalnya dalam bentuk tabel berikut:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *No* | *Use Case* | *Pengujian* | *Jenis Pengujian* | *Identifikasi* |
| *1* | *login* | *1. Skenario normal* | *1. Black box dan White Box*  *2. Black Box* | *U-1-01*  *U-1-02* |
| *2* | *Cari data* | *1. Skenario normal* | *1. Black box dan White Box*  *2. Black Box* | *U-1-01*  *U-1-02* |
| *3* | *Hapus data* | *1. Skenario normal* | *1. Black box dan White Box*  *2. Black Box* | *U-1-01*  *U-1-02* |
| *4* | *Ubah data* | *1. Skenario normal* | *1. Black box dan White Box*  *2. Black Box* | *U-1-01*  *U-1-02* |
| *5* | *Simpan data* | *1. Skenario normal* | *1. Black box dan White Box*  *2. Black Box* | *U-1-01*  *U-1-02* |

### 

### Prosedur Pengujian

Persiapan pengujian

1. ATK
2. program pengujian (visual studio 2010)
3. anggota team kelompok
4. pengadaan computer, printer dan alat jaringan

prosedur pengujian

1. Menyiapkan ATK untuk membuat perancangan ole anggota team kelompok
2. buat rancangan document seperti erd dfd dan use case
3. menguji hasil perancangan berupa dokumen dengan whitebox dan blackbox
4. setelah selesai dan sesuai dengan pengujian dibuat program
5. di uji kembali dengan whitebox dan blackbox
6. bila sesuai dengan program maka pengujian selesai, bila tidak maka prosedur pengujian di ulang kembali hingga dapat sesuai.

## Kasus Uji

Bagian ini diisi dengan kasus uji untuk setiap use case (dibuat subbab untuk setiap use case). Contohnya adalah sebagai berikut:

### Pengujian Use Case <login>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Identifikasi | Deskripsi | Prosedur Pengujian | Masukan | Keluaran yang Diharapkan | Kriteria Evaluasi Hasil | Hasil yang Didapat | Kesimpulan |
| *U-1-01* | Pengujian pemasukkan data admin oleh operator | * Buka table database * Cari Show table data * Isi table database operator (admin) | Kode  Pemasukan username dan password admin(01) | Validasi login | Valid dan invalid login | Valid dan invalid login | diterima |
| U-1-02 | Pengujian hasil pemasukan data pelanggan oleh pelanggan secara on-line | Kode pemasukan on-line  (02) | Validasi login | 02<tgl\_ lahir> <2> | 02<tgl\_ lahir><no\_ terurut> | diterima |

### Pengujian Use Case <cari data>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Identifikasi | Deskripsi | Prosedur Pengujian | Masukan | Keluaran yang Diharapkan | Kriteria Evaluasi Hasil | Hasil yang Didapat | Kesimpulan |
| *U-1-01* | Pengujian pencarian data pelanggan oleh operator | * Buka Form data keterangan * Lakukan pencarian data yang diinginkan * Tulis nama pelanggan di system * Pencarian dilakukan | Pemasukan nama pelanggan (01) | Data pelanggan yang dicari | Id,nama, alamat, tgl daftar, jenis member | Id,nama, alamat, tgl daftar, jenis member | diterima |
| U-1-02 | Pengujian pencarian data pelanggan oleh operator secara on-line | Pemasukan nama pelanggan on-line  (02) | Data pelanggan yang dicari | Id,nama, alamat, tgl daftar, jenis member | Id,nama, alamat, tgl daftar, jenis member | diterima |

### Pengujian Use Case <hapus data>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Identifikasi | Deskripsi | Prosedur Pengujian | Masukan | Keluaran yang Diharapkan | Kriteria Evaluasi Hasil | Hasil yang Didapat | Kesimpulan |
| *U-1-01* | Pengujian penghapusan data pelanggan oleh operator | * Buka Form data keterangan * Lakukan pencarian data yang diinginkan * Tulis nama pelanggan di system * Pencarian dilakukan * Pilih data yang diinginkan * Lakukan penghapusan data | Pemasukan nama pelanggan (01) | Data pelanggan yang berhasil dihapus | Id,nama, alamat, dst | Id,nama, alamat, tgl dst terhapus didatabase | diterima |
| U-1-02 | Pengujian hasil pemasukan data pelanggan oleh pelanggan secara on-line | Pemasukan nama pelanggan on-line  (02) | Data pelanggan yang berhasil di hapus | Id,nama, alamat, dst | Id,nama, alamat, tgl dst terhapus didatabase | diterima |

### Pengujian Use Case <ubah data>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Identifikasi | Deskripsi | Prosedur Pengujian | Masukan | Keluaran yang Diharapkan | Kriteria Evaluasi Hasil | Hasil yang Didapat | Kesimpulan |
| *U-1-01* | Pengujian pengubahan data pelanggan oleh operator | * Buka Form data keterangan * Lakukan pencarian data yang diinginkan * Tulis nama pelanggan di system * Pencarian dilakukan * Pilih data yang diinginkan * Lakukan pengubahan data | Pemasukan nama pelanggan (01) | Data pelanggan yang berhasil diubah | Id,nama, alamat, dst | Id,nama, alamat, tgl dst terubah didatabase | diterima |
| U-1-02 | Pengujian hasil pengubahan data pelanggan oleh pelanggan secara on-line | Pemasukan nama pelanggan on-line  (02) | Data pelanggan yang berhasil diubah | Id,nama, alamat, dst | Id,nama, alamat, tgl dst terubah didatabase | diterima |

### Pengujian Use Case <simpan data>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Identifikasi | Deskripsi | Prosedur Pengujian | Masukan | Keluaran yang Diharapkan | Kriteria Evaluasi Hasil | Hasil yang Didapat | Kesimpulan |
| *U-1-01* | Pengujian penyimpanan data pelanggan oleh operator | * Buka Form pembayan, form pendaftaran dll * Lakukan pemasukan data informasi pelanggan * Lakukan penyimpanan data | Pemasukan data pelanggan (01) | Data pelanggan yang berhasil disimpan | Id,nama, alamat, dst | Id,nama, alamat, tgl dst tersimpan didatabase | diterima |
| U-1-02 | Pengujian hasil penyimpanan data pelanggan oleh pelanggan secara on-line | Pemasukan nama pelanggan on-line  (02) | Data pelanggan yang berhasil disimpan | Id,nama, alamat, dst | Id,nama, alamat, tgl dst tersimpan didatabase | diterima |

## Defect dan Status Perbaikan

* Pencarian data pelanggan
* Hapus data pelanggan
* Date time picker

## Evaluasi Pengujian

Bagian ini diisi dengan uraian evaluasi hasil pengujian.

# Lampiran

Bagian lampiran diisi dengan

* pembagian tugas,
* Bambang Imam Hermawan ( programmer dan document srs )
* Aidil Pratama Putra ( konsumsi, tempat dan perancangan )
* Giri Muria Shaleh ( programmer dan document srs )
* Chandra Agustian ( document srs dan analisis perancangan )
* Asep Anwar Musadad ( analisis perancangan dan document srs )
* rencana iterasi,
* notulen koordinasi

hasil dari kegiatan kelompok kita telah ditentukan hasil dari koordinasi tentang pembuatan aplikasi pendaftaran rental ps 3

yaitu :

1. agar mempermudah bagi admin utuk mengelola data pembayaran dan pendaftaran

2. mempercepat proses kerja

3. Keamanan data

* user manual,
* contoh brosur promosi produk perangkat lunak.,
* tabel *traceability* (contoh lihat halaman berikutnya).

Tracebility Use Case

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Analisis** | | **Perancangan** | | | **Pengujian** |
| **SRS-ID** | **Use Case ID** | **Skenario** | **Sequence** | **Diagram** | **Sequence** | **Diagram** | **Antarmuka** | **Kasus** |
|  |  |  | **Diagram** | **Kelas** | **Diagram** | **Kelas** |  | **Uji** |
| SRS-F-01 | UC-01 | SC-01-01 | SQ-A-01-01 | CL-A-01 | SQ-D-01-01 | CL-D-01 | INT-01 | U-01 |
|  |  | SC-01-02 | SQ-A-01-02 |  | SQ-D-01-02 |  | INT-02 | U-02 |
|  | UC-02 | SC-02-01 | SQ-A-02-01 | CL-A-02 | SQ-D-02-01 | CL-D-02 | INT-01 | U-03 |
|  |  | SC-02-02 | SQ-A-02-02 |  | SQ-D-02-02 |  | INT-03 | U-04 |
|  |  | SC-02-03 | SQ-A-02-03 |  | SQ-D-02-03 |  | INT-04 | U-05 |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| Catatan: Setiap kolom diisi dengan nomor/ID | | | |  |  |  |  |  |