Dear all,

IF-UNIKOM-UP-DOC-2009

DOKUMEN PEMBANGUNAN PERANGKAT LUNAK

<Info Wisata Bandung>

Versi: <Draft-tgl/Final>

untuk:

Universitas Komputer Indonesia

Dipersiapkan oleh:

<Kelompok 5>

10112747 – Rahmat Syaparudin

10112046 – Fuji Nuriany Agustina

10112729 – Rudi Kurnia

10111203 – Aprial Rajana

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* | | *1/29* |
| Revisi | *0* | *Tgl: <tgl>* |

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) 6

2 Kebutuhan Perangkat Lunak 7

2.1 Deskripsi Umum Sistem 7

2.2 Fitur Utama Perangkat Lunak 7

2.2.1 Kebutuhan Fungsional 7

2.2.2 Kebutuhan Non Fungsional 7

2.3 Model Use Case 7

2.3.1 Diagram Use Case 7

2.3.2 Definisi Actor 7

2.3.3 Definisi Use Case 8

2.3.4 Skenario Use Case 8

2.4 Spesifikasi Tambahan 8

2.5 Glossary 8

3 Model Analisis 9

3.1 Realisasi Use Case Tahap Analisis 9

3.2 Diagram Kelas Keseluruhan 9

3.3 Kelas Analisis 9

3.4 Paket Analisis 9

3.4.1 Identifikasi Paket Analisis 9

3.4.2 Identifikasi Kelas Analisis tiap Paket 10

3.5 Deskripsi Arsitektur 10

3.6 Pedoman Perancangan 10

4 Model Perancangan 11

4.1 Realisasi Use Case Tahap Perancangan 11

4.1.1 Use Case <nama use case 1> 11

4.1.1.1 Identifikasi Elemen WAE - Logical View 11

4.1.1.2 Identifikasi Kelas Perancangan 11

4.1.1.3 Sequence Diagram 11

4.1.1.4 Diagram Kelas Perancangan 11

4.1.2 Use Case <nama use case 2> 11

4.1.2.1 Identifikasi Elemen WAE - Logical View 11

4.1.2.2 Identifikasi Kelas Perancangan 12

4.1.2.3 Sequence Diagram 12

4.1.2.4 Diagram Kelas Perancangan 12

4.1.3 Use Case <nama use case 3> 12

4.1.3.1 Identifikasi Elemen WAE - Logical View 12

4.1.3.2 Identifikasi Kelas Perancangan 13

4.1.3.3 Sequence Diagram 13

4.1.3.4 Diagram Kelas Perancangan 13

4.2 Perancangan Detil Elemen Logical View 13

4.2.1 Stereotyped Class <nama elemen logical view> 13

4.2.2 Stereotyped Class <nama elemen logical view> 14

4.3 Perancangan Detil Kelas 14

4.3.1 Kelas <nama kelas> 14

4.3.2 Kelas <nama kelas> 15

4.4 Diagram Kelas Keseluruhan 15

4.5 Algoritma/Query 15

4.6 Diagram Statechart 15

4.7 Perancangan Antarmuka 16

4.8 Perancangan Representasi Persistensi Kelas 16

4.9 Coding Standard dan Naming Convention 16

4.10 Deployment Diagram 16

5 Implementasi 17

5.1 Lingkungan Implementasi 17

5.2 Implementasi Kelas 17

5.3 Implementasi Elemen WAE (Component View) 17

5.4 Implementasi Antarmuka 17

5.5 File Lain 17

6 Pengujian 18

6.1 Rencana dan Prosedur Pengujian 18

6.1.1 Rencana Pengujian 18

6.1.2 Prosedur Pengujian 18

6.2 Kasus Uji 18

6.2.1 Pengujian Use Case <nama use case> 18

6.3 Defect dan Status Perbaikan 18

6.4 Evaluasi Pengujian 18

7 Lampiran 19

# 

# Pendahuluan

## Tujuan Penulisan Dokumen

Tujuan penulisan dokumen ini yaitu untuk memberi informasi mengenai wisata alam di Bandung, Jawa Barat, serta untuk memenuhi tugas mata kuliah Rekayasa Perangkat Lunak 2 di Universitas Komputer Indonesia (UNIKOM). Dokumen ini bertujuan untuk dijadikan bahan acuan bagi dua pihak yang terkait, antara pengembang dan pengguna. Untuk pengembang dokumen ini digunakan sebagai acuan dalam pembuatan perangkat lunak, sedangkan bagi pengguna dokumen ini digunakan untuk mencatat semua spesifikasi kebutuhan yang akan dibutuhkan nanti oleh pengguna. Perangkat lunak sistem informasi yang dibuat ini berfungsi untuk memberikan informasi wisata Bandung.

## Lingkup Masalah

Perangkat lunak Sistem Informasi Wisata Bandung dikembangkan dengan tujuan untuk:

1. Memberikan informasi tempat wisata yang ada di Bandung.
2. Memberikan informasi lokasi tempat wisata yang ada di Bandung.
3. Memberikan gambaran tempat wisata beruapa galeri foto

## Aturan Penomoran

Bagian ini diisi dengan aturan penomoran yang digunakan dalam dokumen.

## Referensi

Bagian ini diisi dengan daftar Buku, Panduan, atau Dokumentasi lain yang digunakan sebagai referensi dalam pembuatan dokumen dan dalam pengembangan perangkat lunak..

## Deskripsi Umum Dokumen (Ikhtisar)

|  |  |
| --- | --- |
| BAB I | Pendahuluan |
|  | Pendahuluan terdiri dari tujuan penulisan dokumen, lingkup masalah, aturan penomoran, referensi dan deskripsi umum dokumen. |
| BAB II | Kebutuhan Perangkat Lunak |
|  | Kebutuhan perangkat lunak terdiri dari deskripsi umum sistem, fitur utama perangkat lunak, model use case, spesifikasi tambahan dan glossary. Fitur utama perangkat lunak mencakup kebutuhan fungsional dan kebutuhan non fungsional. |
| BAB III | Model Analisis |
|  | Model analisis terdiri dari realisasi use case tahap analisis, diagram kelas keseluruhan, kelas analisis, paket analisis, deskripsi arsitektur, dan pedoman perancangan. Paket analisis mencakup identifikasi paket analisis dan identifikasi kelas analisis tiap paket. |
| BAB IV | Model Perancangan |
|  | Model perancangan terdiri dari realisasi use case tahap perancangan, perancangan detil elemen logical view, perancangan detil class, diagram kelas keseluruhan, algoritma query, diagram statechart, perancangan antarmuka, perancangan representasi persistensi class, koding standar dan naming convention, dan development diagram. |
| BAB V | Implementasi |
|  | Implementasi terdiri dari lingkungan implemnetasi, implementasi kelas, implementasi elemen WAE, implementasi antarmuka, dan file lain. |
| BAB VI | Pengujian |
|  | Pengujian terdiri dari rencana dan prosedur pengujian, dan kasus uji. |
| BAB VII | Lampiran |
|  |  |

# Kebutuhan Perangkat Lunak

## Deskripsi Umum Sistem

Aplikasi info wisata bandung berbasis android ini membutuhkan jaringan koneksi internet untuk mengakses database sehingga akan menampilkan konten yang tersedia di database. Aplikasi ini berguna untuk memberikan informasi tentang tempat-tempat wisata yang ada di bandung.

## Fitur Utama Perangkat Lunak

Beberapa fitur yang tersedia di aplikasi info wisata bandung berbasis android ini adalah:

* Pencarian informasi tempat wisata di bandung
* Pencarian rute menuju tempat wisata dengan menggunakan gps
* Pencarian lokasi wisata berdasarkan koordinat lokasi di maps
* Pencarian tempat wisata berdasarkan kategori wilayah

### Kebutuhan Fungsional

|  |  |  |
| --- | --- | --- |
| Kode | Kebutuhan Fungsional | Keterangan |
| SRS-F-001 | Sistem menampilkan menu utama | Merupakan tampilan awal sistem ketika user masuk atau membuka aplikasi |
| SRS-F-002 | Sistem menampilkan semua daftar tempat wisata di bandung | Merupakan tampilan |
| SRS-F-003 | Sistem menampilkan daftar kategori tempat wisata yag ada di bandung |  |
| SRS-F-004 | Sistem menampilkan daftar wisata berdasarkan kategori yang dipilih |  |
| SRS-F-005 | Sistem menampilkan tentang informasi aplikasi |  |
| SRS-F-006 | Sistem menampilkan informasi tempat wisata yang dipilih |  |
| SRS-F-007 | Sistem menampilkan galeri foto tempat wisata yang dipilih |  |
| SRS-F-008 | Sistem menampilkan lokasi koordinat tempat wisata yang dipilih melalui maps |  |
| SRS-F-009 | Sistem menampilkan rute perjalanan ke tempat wisata menggunakan layanan gps |  |
| SRS-F-010 | Sistem menampilkan hasil pencarian tempat wisata |  |

### Kebutuhan Non Fungsional

SRS-NF-001 Sistem memberikan respon terhadap aksi pengguna dalam waktu kurang dari 5 detik.

## 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* | *User* | *Actor dengan role ini mempunyai hak mengakses informasi wisata berupa keterangan tempat wisata, galeri, dan lokasi koordinat dari tempat wisata tersebut.* |
| *2* | *Admin* | *Actor dengan role ini mempunyai wewenang untuk mengakses, mengubah, menambah, dan menghapus konten yang tersedia di database.* |
| *3* | *Webservice* | *Aktor ini merupakan webservice untuk menampilkan database. Aktor ini lebih bersifat pasif. Beberapa use case membutuhkan membutuhkan informasi dari webservice ini secara up-to-date* |

### Definisi Use Case

Bagian ini diisi dengan daftar use case dan deskripsi singkat mengenai use case tersebut. Bisa dibuat dalam bentuk tabel berikut:

|  |  |  |
| --- | --- | --- |
| *No* | *Use Case* | *Deskripsi* |
| *1* | *Lihat Info Wisata* | *Sistem menampilkan daftar wisata yang dapat dipilih pengguna agar bisa melihat detail informasi wisata tersebut* |
| *2* | *Lihat Kategori* | *Sistem menampilkan kategori wisata berdasarkan lokasi yang tersedia di dalam aplikasi.* |
| *3* | *Lihat Info* | *Sistem menampilkan info aplikasi yang dapat dilihat pengguna seperti versi, dan pembuat aplikasi* |
| *4* | *Pilih Wisata* | *Sistem meampilkan pilihan daftar wisata* |
| *5* | *Cari Wisata* | *Sistem menampilkan hasil dari wisata yang dicari oleh user* |
| *6* | *Pilih Kategori* | *Sistem menmapilkan pilihan kategori berdasarkan wilayah* |
| *7* | *Lihat Galleri* | *Sistem menampilkan informasi wisata berupa galeri foto* |
| *8* | *Lihat Maps* | *Sistem menampilkan lokasi koordinat tempatwisata dengan marker di maps* |
| *9* | *Lihat Rute* | *Sistem menampilkan rute perjalanan* |
| *10* | *Mengolah Database* |  |

### Skenario Use Case

Bagian ini diisi dengan skenario (*flow of event*) untuk beberapa use case utama, yang menggambarkan urutan interaksi actor dengan use case tersebut, dari awal sampai akhir.

*Nama Use Case: Lihat Info Wisata*

*Aktor : User*

*Kondisi Awal : User berada di Menu Utama apikasi*

*Kondisi Akhir : Sistem menampilkan info wisata yg dipilih user*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Memilih menu Info Wisata* |  |
|  | *2. Menampilkan daftar wisata dari webservice ke layar* |
| *3. Menekan/memilih salah satu tempat wisata yang ingin dilihat atau mengetik nama wisata dengan pencarian* |  |
|  | *4. Menampilkan info wisata yang dipilih user (berupa deskripsi tempat wisata tersebut)* |
| *Skenario Alternatif 1* |  |
| *1. Memilih menu Info Wisata* |  |
|  | *2. Menampilkan daftar wisata dari webservice ke layar* |
| *3. Mengetik nama wisata di form pencarian* |  |
|  | *4. Menampilkan hasil pencarian* |
| *5. Memilih tempat wisata (hasil Pencarian)* |  |
|  | *6. Menampilkan info wisata yang dipilih user (berupa deskripsi tempat wisata tersebut)* |
| *Skenario Alternatif 2* |  |
| *1. Memilih menu Info Wisata* |  |
|  | *2. Menampilkan layar putih (blank)* |

*Nama Use Case: Lihat kategori*

*Aktor : User*

*Kondisi Awal : User berada di Menu Utama apikasi*

*Kondisi Akhir : Sistem menampilkan info wisata berdasarkan kategori yg dipilih user*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Memilih menu Kategori* |  |
|  | *2. Menampilkan menu kategori berdasarkan wilayah* |
| *3. Menekan/memilih salah satu menu kategori (Misal Bandung Barat)* |  |
|  | *4. Menampilkan daftar wisata dari webservice ke layar berdasarkan kategori yang dipilih* |
| *Skenario Alternatif* |  |
| *1. Menekan/memilih salah satu menu kategori (Misal Bandung Barat)* |  |
|  | *2. Menampilkan layar putih (blank)* |

*Nama Use Case: Lihat Info*

*Aktor : User*

*Kondisi Awal : User berada di Menu Utama apikasi*

*Kondisi Akhir : Sistem menampilkan info aplikasi*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Memilih menu Info* |  |
|  | *2. Menampilkan informasi aplikasi (versi dan pembuat aplikasi)* |

*Nama Use Case: Lihat Galleri*

*Aktor : User*

*Kondisi Awal : Sistem menampilkan deskripsi/info wisata yang dipilih*

*Kondisi Akhir : Sistem menampilkan galeri foto tempat wisata*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
|  | *1. Menampilkan deskripsi/info wisata yang dipilih* |
| *2. Menekan menu Galleri* |  |
|  | *3. Menampilkan galeri foto dari info wisata yang dipilih* |
| *Skenario Alternatif* |  |
| *1. Memilih menu Galleri* |  |
|  | *2. Menampilkan pesan “Image Does Not exist or Network Error”* |

*Nama Use Case: Lihat Peta*

*Aktor : User*

*Kondisi Awal : Sistem menampilkan deskripsi/info wisata yang dipilih*

*Kondisi Akhir : Sistem menampilkan peta lokasi tempat wisata*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
|  | *1. Menampilkan deskripsi/info wisata yang dipilih* |
| *2. Menekan menu Galleri* |  |
|  | *3. Menampilkan marker lokasi wisata di peta* |
| *Skenario Alternatif* |  |
| *1. Memilih menu Galleri* |  |
|  | *2. Menampilkan pesan “No Images Exist”* |

*Nama Use Case: Lihat Rute*

*Aktor : User*

*Kondisi Awal : Sistem menampilkan deskripsi/info wisata yang dipilih*

*Kondisi Akhir : Sistem menampilkan rute perjalanan*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
|  | *1. Menampilkan deskripsi/info wisata yang dipilih* |
| *2. Menekan menu Get Direction* |  |
|  | *3. Mengalihkan user ke aplikasi maps* |
|  | *4. Menampilkan rute perjalanan ke tempat wisata* |
| *4. Menentukan starting point perjalanan* |  |
|  | *5. Menghitung ulang dan menampilkan rute perjalanan* |
| *Skenario Alternatif* |  |
| *1.* |  |
|  | *2.* |

## Spesifikasi Tambahan

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

## Glossary

|  |  |  |
| --- | --- | --- |
| Webservice | : | Sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah *database* ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan di platform apa sebuah data itu dikonsumsi. |

# 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

Pada fase ini, bagian ini diisi dengan diagram kelas keseluruhan. Lengkapi dengan kalimat pengantar.

## Kelas Analisis

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

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas* | *Jenis* |
|  |  |  |
|  |  |  |
|  |  |  |

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

* identifikasi tanggung-jawab (responsibility)
* identifikasi atribut

|  |  |  |
| --- | --- | --- |
| *Nama Kelas* | *Daftar Tanggung-Jawab* | *Daftar Atribut* |
| *Kelas xxxx* | *1.*  *2.*  *3.* | *1.*  *2.*  *3.* |
|  |  |  |
|  |  |  |

## Paket Analisis

### Identifikasi Paket Analisis

Jika perlu, pemaketan dapat dilakukan untuk menyederhanakan persoalan.

Bagian ini dapat diisi dengan daftar paket analisis dengan mengacu pada diagram use case. Satu atau lebih use case dapat digabung kedalam satu paket. Satu use case hanya boleh berada pada satu paket.

*Contoh:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Paket* | *Use Case Terkait* |
| *1.* | *Paket Pengelolaan Informasi* | *1. Pengelolaan Informasi Pelanggan*  *2. Pengelolaan Informasi Pegawai*  *3. Pengelolaan Informasi Produk* |
|  |  |  |

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.

Arsitektur aplikasi info wisata bandung ini menggunakan database dari. Database yang digunakan adalah MySQL Database. Arsitektur aplikasi sistem info wisata bandung berupa aplikasi berbasis android. Spesifikasi yang dibutuhkan untuk mengoperasikan aplikasi ini adalah Android dengan API level 14 (Ice Cream Sandwich) sampai API level 21 (Lollipop).

## Pedoman Perancangan

Bagian ini diisi dengan uraian mengenai framework atau arsitektur khusus yang digunakan pada saat perancangan dan implementasi. Jika tidak ada, hilangkan subbab ini

# Model Perancangan

## Realisasi Use Case Tahap Perancangan

### Use Case <nama use case 1>

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

Buatlah diagram sequence untuksetiap skenario use case.

#### Diagram Kelas Perancangan

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

### Use Case <nama use case 2>

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

Buatlah diagram sequence untuksetiap skenario use case.

#### Diagram Kelas Perancangan

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

### Use Case <nama use case 3>

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

Buatlah diagram sequence untuksetiap skenario use case.

#### 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 <nama kelas>

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

*Nama Kelas : ……..*

|  |  |  |
| --- | --- | --- |
| ***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* |
|  |  |  |
|  |  |  |

### Kelas <nama kelas>

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

*Nama Kelas: ……..*

|  |  |  |
| --- | --- | --- |
| ***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* |
|  |  |  |
|  |  |  |

## 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 :*

*Nama Operasi :*

*Algoritma : (Algo-xxx)*

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

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Tuliskan fungsi dari querynya* |
|  |  |  |
|  |  |  |

## Diagram Statechart

Bagian ini hanya diisi jika ada kelas yang kompleks. Perubahan status kelas tersebut harus digambarkan dalam bentuk diagram statechart. Boleh dibuat subba per kelas.

## Perancangan Antarmuka

Pada fase Inception:

Bagian ini diisi dengan versi awal prototipe antarmuka.

Pada fase Elaboration:

Bagian ini diisi dengan perbaikan rancangan antarmuka. Lengkapi daftar perubahan jika terjadi perbaikan. Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya, misalnya seperti di bawah ini:

*Antarmuka : {diisi dengan no. layar atau no gambar rancangan antarmuka}*

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
|  |  | *Diisi dengan string yg tampil pd layar* | Diisi dengan penjelasan reaksi sistem, misalnya membuka layar apa, link kemana. Jika menyangkut suatu kode yang cukup rumit, acu algoritma yang telah diuraikan di atas. |
| *Button1* | Button | Info Wisata | Jika diklik, akan mengaktifkan Proses AlgoXXX. |
| *Button2* | Button | *Kategori* |  |
| *Button3* | Button | *Info* |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

Pada fase Construction:

Bagian ini diisi dengan versi final dari rancangan antarmuka. Lengkapi daftar perubahan jika terjadi perbaikan.

## 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:

Bagian ini diisi dengan *coding standard* dan *naming convention* yang akan digunakan.

Pada fase Construction:

Bagian ini diisi dengan versi final dari *coding standard* dan *naming convention.* Lengkapi daftar perubahan jika terjadi perbaikan.

## Deployment Diagram

Bagian ini diisi dengan deployment diagram yang menggambarkan alokasi proses pada node.

# Implementasi

Bagian ini diisi dengan informasi tentangelemen dari perangkat lunak yang dikembangkan (*executable files*, *configuration files, data files,* dsb) serta perubahannya.

## Lingkungan Implementasi

Bagian ini diisi dengan lingkungan implementasi yaitu hardware dan software yang digunakan untuk implementasi.

## 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* | *Direct Activity* | *DirectActivity.java* | *DirectActivity.class* |
| *2* | *Gallery Activity* | *GalleryActivity.java* | *GalleryActivity.class* |
| *3* | *Info Activity* | *InfoActivity.java* | *InfoActivity.class* |
| *4* | *Kategori Activity* | *KategoriActivity.java* | *KategoriActivity.class* |
| *5* | *Main Activity* | *MainActivity.java* | *MainActivity.class* |
| *6* | *Splash Activity* | *SplashActivity.java* | *SplashActivity.class* |
| *7* | *Single Contact Activity* | *SingleContactActivity.java* | *SingleContactActivity.class* |
| *8* | *Service Handler* | *ServiceHandler.java* | *ServiceHandler.class* |
| *9* | *Menu Activity* | *MenuActivity.java* | *MenuActivity.class* |
| *10* | *Mapsb Activity* | *MapsbActivity.java* | *MapsbActivity.class* |

## 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* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 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* | *xxx* | *1. Skenario normal*  *2. Skenario xxx (acu no.skenario)*  *3. Skenario yyy* | *1. Black box dan White Box*  *2. Black Box*  *3.* | *U-1-xxx*  *U-1-xxx*  *U-1-xxx* |
|  |  |  |  | *U-2-xxx* |

### Prosedur Pengujian

Bagian ini diisi dengan prosedur pengujian, misalnya persiapan pengujian, urutan pengujian yang harus dilakukan, dll.

Bagian ini diisi dengan prosedur pengujian versi final. Lengkapi daftar perubahan.

## 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 <nama use case>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Identifikasi | Deskripsi | Prosedur Pengujian | Masukan | Keluaran yang Diharapkan | Kriteria Evaluasi Hasil | Hasil yang Didapat | Kesimpulan |
| *U-1-01* | Pengujian hasil pemasukan data pelanggan oleh operator | * Buka File data pelanggan * Cari rekord dengan data modus pemasukan yang diinginkan * Lihat tanggal lahir pelanggan * Lihat kode pelanggan * Bandingkan dengan rumus pembangkitan kode pelanggan | Kode modus pemasukan operator (01) | 01<tgl\_lahir>001  01<tgl\_lahir>002  01<tgl\_lahir>003  dst | 01<tgl\_ lahir> <nomor terurut> | 01<tgl\_ lahir><no\_ loncat | ditolak |
| U-1-02 | Pengujian hasil pemasukan data pelanggan oleh pelanggan secara on-line | Kode modus pemasukan on-line  (02) | 02<tgl\_lahir>001  02<tgl\_lahir>002  02<tgl\_lahir>003  dst | 02<tgl\_ lahir> <nomor terurut> | 02<tgl\_ lahir><no\_ terurut> | diterima |

## Defect dan Status Perbaikan

Bagian ini diisi dengan defect yang ditemukan setelah melakukan pengujian dan status perbaikannya

## Evaluasi Pengujian

Bagian ini diisi dengan uraian evaluasi hasil pengujian.

# Lampiran

Bagian lampiran diisi dengan

* pembagian tugas,
* rencana iterasi,
* notulen koordinasi
* user manual,
* contoh brosur promosi produk perangkat lunak.,
* tabel *traceability* (contoh lihat halaman berikutnya).

LAMPIRAN 1 : Pembagian Tugas

|  |  |  |  |
| --- | --- | --- | --- |
| No | NIM | Nama | Tugas |
| 1 | 10112 | Aprial Rajana | Membuat aplikasi, laporan |
| 2 | 10112046 | Fuji Nuriany Agustina | Membuat presentasi, membuat video presentasi, membuat video demo, membuat brosur produk. |
| 3 | 10112729 | Rudi Kurnia | Membuat database, membuat aplikasi, laporan |
| 4 | 10112747 | Rahmat Syaparudin | Membuat Aplikasi, membuat webserver, membuat hosting, laporan |

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 | | | |  |  |  |  |  |