Dear all,

SKPL-001

DOKUMEN PEMBANGUNAN PERANGKAT LUNAK

Aplikasi E-Magang Berbasis Website

Versi: 0.0.1-27–Juni–2014

untuk:

Universitas Komputer Indonesia

Dipersiapkan oleh:

Ahmad Paudji HS 10111104

Ismail Zakky 10111099

Handoyo 10111078

Wupi 10111068

Andrew Tooy 10111931

Jurusan Teknik Informatika – Universitas Komputer Indonesia

Jl. Dipati Ukur Nomor 112-114, 40132

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| unikom | **Program Studi**  **Teknik Informatika UNIKOM** | Nomor Dokumen | | Halaman |
| *SKPL01* | | *1/29* |
| Revisi | *0* | *Tgl: 27 – Juni 2014* |

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

Bagian ini diisi dengan uraian ringkas mengenai isi dokumen, tujuan pembuatan dokumen, dan target pembaca dokumen ini.

## Lingkup Masalah

Bagian ini diisi dengan uraian lingkup masalah yaitu membangun perangkat lunak <nama perangkat lunak>. Jelaskan dengan singkat dan jelas keterkaitan perangkat lunak tersebut dengan perangkat lunak lainnya.

## 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)

Bagian ini diisi dengan sistematika pembahasan dokumen ini. Uraikan bab per bab.

# Kebutuhan Perangkat Lunak

## Deskripsi Umum Sistem

Bagian ini diisi dengan *System Overview*, dalam bentuk gambar dan narasi yang dapat memberikan gambaran tentang aplikasi dan konteksnya (harus jelas batasan dan keterkaitan antara sistem yang akan dibangun dengan sistem lain di luarnya). Deskripsi umum sistem perlu dilengkapi pula dengan pemodelan kasus yang dihadapi. Berikan narasi yang jelas mengenai gambar yang ditampilkan, dalam kalimat yang jelas.

## Fitur Utama Perangkat Lunak

Bagian ini diisi dengan fitur utama perangkat lunak, yang terdiri dari kebutuhan fungsional dan kebutuhan non fungsional. Deskripsi fitur harus dinyatakan dengan jelas dalam kalimat yang lengkap (jelas subyek, predikat, dan objek-nya). Setiap fitur diberi kode dengan aturan penomoran yang telah dijelaskan sebelumnya.

### Kebutuhan Fungsional

Contoh:

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

### Kebutuhan Non Fungsional

Contoh:

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

## Model Use Case

### Diagram Use Case

Bagian ini diisi dengan diagram use case keseluruhan.

### 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* | *Guest* | *Actor dengan role ini mempunyai wewenang untuk melakukan registrasi serta melihat informasi-informasi yang sifatnya umum seperti profil perusahaan, ….* |

### 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* | *Melihat daftar produk* | *Sistem menampilkan daftar produk yang boleh dipilih untuk pengguna.* |
|  |  |  |

### 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.

*Contoh:*

*Nama Use Case: Melihat daftar produk*

*Skenario:*

|  |  |
| --- | --- |
| ***Aksi Actor*** | ***Reaksi Sistem*** |
| *Skenario Normal* |  |
| *1. Memilih menu Daftar Produk* |  |
|  | *2. Menampilkan daftar produk dari basisdata ke*  *layar* |
| *3. Menekan tombol navigasi (next, prev)* |  |
|  | *4. Me-refresh tampilan daftar produk* |
| *Skenario Alternatif* |  |
| *1. Memilih menu Daftar Produk* |  |
|  | *2. Menampilkan pesan ‘Tidak ada produk’* |

## 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

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.

## 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 | OK | Jika diklik, akan mengaktifkan Proses AlgoXXX. |
| *RTF1* | *RTF Box* |  | Isi Teks yang disimpan pada File xxx |

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 nod