

IF-UNIKOM-UP-DOC-

DOKUMEN PEMBANGUNAN PERANGKAT LUNAK

<Nama Perangkat Lunak>

Versi: <Draft-tgl/Final>

untuk:


Universitas Komputer Indonesia

Dipersiapkan oleh:

<Nama Grup & Anggota>

Jurusan Teknik Informatika – Universitas Komputer Indonesia

Jl. Dipati Ukur Nomor 112-114, 40132

	Program Studi Teknik Informatika UNIKOM	Nomor Dokumen		Halaman
		<i>PP IF-UNIKOM-UP-DOC-2009</i>		<i>1/29</i>
		Revisi	<i>0</i>	<i>Tgl: <tgl></i>

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

1 Pendahuluan

1.1 Tujuan Penulisan Dokumen

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

1.2 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.

1.3 Aturan Penomoran

Bagian ini diisi dengan aturan penomoran yang digunakan dalam dokumen.

1.4 Referensi

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

1.5 Deskripsi Umum Dokumen (Ikhtisar)

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

2 Kebutuhan Perangkat Lunak

2.1 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.

2.2 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.

2.2.1 Kebutuhan Fungsional

Contoh:

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

2.2.2 Kebutuhan Non Fungsional

Contoh:

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

2.3 Model Use Case

2.3.1 Diagram Use Case

Bagian ini diisi dengan diagram use case keseluruhan.

2.3.2 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,

2.3.3 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.

2.3.4 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
<i>Skenario Normal</i>	
1. Memilih menu Daftar Produk	
	2. Menampilkan daftar produk dari basisdata ke layar
3. Menekan tombol navigasi (<i>next, prev</i>)	
	4. Me-refresh tampilan daftar produk
<i>Skenario Alternatif</i>	
1. Memilih menu Daftar Produk	
	2. Menampilkan pesan 'Tidak ada produk'

2.4 Spesifikasi Tambahan

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

2.5 Glossary

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

3 Model Analisis

3.1 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)

3.2 Diagram Kelas Keseluruhan

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

3.3 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.

3.4 Paket Analisis

3.4.1 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 ke dalam 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.

3.4.2 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.	

3.5 Deskripsi Arsitektur

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

3.6 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

4 Model Perancangan

4.1 Realisasi Use Case Tahap Perancangan

4.1.1 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.

4.1.1.1 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 Logik WAE	Stereotype
		<i>Server Page</i>
		<i>Server Page</i>
		<i>Client Page</i>

4.1.1.2 Identifikasi Kelas Perancangan

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

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait

4.1.1.3 Sequence Diagram

Buatlah diagram sequence untuk setiap skenario use case.

4.1.1.4 Diagram Kelas Perancangan

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

4.1.2 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.

4.1.2.1 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 Logik WAE	Stereotype
		<i>Server Page</i>
		<i>Server Page</i>
		<i>Client Page</i>

4.1.2.2 Identifikasi Kelas Perancangan

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

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait

4.1.2.3 Sequence Diagram

Buatlah diagram sequence untuk setiap skenario use case.

4.1.2.4 Diagram Kelas Perancangan

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

4.1.3 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.

4.1.3.1 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 Logik WAE	Stereotype
		<i>Server Page</i>
		<i>Server Page</i>
		<i>Client Page</i>

4.1.3.2 Identifikasi Kelas Perancangan

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

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait

4.1.3.3 Sequence Diagram

Buatlah diagram sequence untuk setiap skenario use case.

4.1.3.4 Diagram Kelas Perancangan

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

4.2 Perancangan Detil Elemen Logical View

No	Nama Elemen Logik WAE	Stereotype
		Server Page
		Server Page
		Client Page

Untuk setiap elemen:

- identifikasi atribut
- identifikasi operasi

4.2.1 Stereotyped Class <nama elemen logical view>

Nama Kelas :

Jenis : Server Page / Client Page / HTML Form

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>Diisi dengan signature operasi</i>		
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>Diisi dengan nama atribut</i>		<i>Tuliskan tipenya sesuai dengan yang dikenal pada bahasa pemrograman yang digunakan</i>

4.2.2 Stereotyped Class <nama elemen logical view>

Nama Kelas :
Jenis : Server Page / Client Page / HTML Form

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>Diisi dengan signature operasi</i>		
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>Diisi dengan nama atribut</i>		<i>Tuliskan tipenya sesuai dengan yang dikenal pada bahasa pemrograman yang digunakan</i>

4.3 Perancangan Detil Kelas

Bagian ini diisi dengan daftar kelas perancangan dalam tabel berikut:

<i>No</i>	<i>Nama Kelas Perancangan</i>	<i>Nama Kelas Analisis Terkait</i>

Untuk setiap kelas:

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

4.3.1 Kelas <nama kelas>

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

Nama Kelas :

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>Diisi dengan signature operasi</i>		
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>Diisi dengan nama atribut</i>		<i>Tuliskan tipenya sesuai dengan yang dikenal pada bahasa pemrograman yang digunakan</i>

4.3.2 Kelas <nama kelas>

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

Nama Kelas:

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
<i>Diisi dengan signature operasi</i>		
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
<i>Diisi dengan nama atribut</i>		<i>Tuliskan tipenya sesuai dengan yang dikenal pada bahasa pemrograman yang digunakan</i>

4.4 Diagram Kelas Keseluruhan

Bagian ini diisi dengan diagram kelas keseluruhan.

4.5 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 :

<i>No Query</i>	<i>Query</i>	<i>Keterangan</i>
<i>Q-xxx</i>		<i>Tuliskan fungsi dari querynya</i>

4.6 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.

4.7 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
		<i>Diisi dengan string yg tampil pd layar</i>	<i>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.</i>
<i>Button1</i>	<i>Button</i>	<i>OK</i>	<i>Jika diklik, akan mengaktifkan Proses AlgoXXX.</i>
<i>RTF1</i>	<i>RTF Box</i>		<i>Isi Teks yang disimpan pada File xxx</i>

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.

4.8 Perancangan Representasi Persistensi Kelas

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

4.9 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.

4.10 Deployment Diagram

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

Program Studi Teknik Informatika UNIKOM	PP IF-UNIKOM-UP-DOC-2011	Halaman 16/ dari 20 halaman
--	---------------------------------	------------------------------------

Template dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UNIKOM dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika UNIKOM.

5 Implementasi

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

5.1 Lingkungan Implementasi

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

5.2 Implementasi Kelas

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

<i>No</i>	<i>Nama Kelas</i>	<i>Nama File Fisik</i>	<i>Nama File Executable</i>
<i>1</i>	<i>Account</i>	<i>Account.java</i>	<i>Account.class</i>

5.3 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.

5.4 Implementasi Antarmuka

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

<i>No</i>	<i>Antarmuka</i>	<i>Nama File Fisik</i>	<i>Nama File Executable</i>

5.5 File Lain

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

<i>No</i>	<i>Nama File</i>	<i>Keterangan</i>

6 Pengujian

6.1 Rencana dan Prosedur Pengujian

6.1.1 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

6.1.2 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.

6.2 Kasus Uji

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

6.2.1 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	<ul style="list-style-type: none">Buka File data pelangganCari record dengan data modul pemasukan yang diinginkan	Kode modul 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	<ul style="list-style-type: none">Lihat tanggal lahir pelangganLihat kode pelangganBandingkan dengan rumus pembangkitan kode pelanggan	Kode modul 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

6.3 Defect dan Status Perbaikan

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

6.4 Evaluasi Pengujian

Bagian ini diisi dengan uraian evaluasi hasil pengujian.

Program Studi Teknik Informatika UNIKOM	PP IF-UNIKOM-UP-DOC-2011	Halaman 18/ dari 20 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UNIKOM dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika UNIKOM.		

7 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).

Tracebility Use Case

SRS-ID	Use Case ID	Skenario	Analisis		Perancangan			Pengujian
			Sequence Diagram	Diagram Kelas	Sequence Diagram	Diagram Kelas	Antarmuka	Kasus 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