

SOFTWARE REQUIREMENT SPESIFICATION TABUNGKEUN

untuk :

KAUM MILENIAL



Dipersiapkan oleh:


Sunflower

M.Fikri Firdaus(6706174033)

Thalia Veronica(6706170009)

Thalia Veronica(6706174132)

**D3 REKAYASA PERANGKAT LUNAK APLIKASI
FAKULTAS ILMU TERAPAN
TELKOM UNIVERSITY
2019**

	D3 Rekayasa Perangkat Lunak Aplikasi	Nomor Dokumen		Halaman
		02/SRS/III/2019		
		Revisi	1.0	Tgl: <isi tanggal>

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

I. PENDAHULUAN	5
1.1 Tujuan Penulisan Dokumen	5
1.2 Lingkup Masalah	5
1.3 Definisi dan Istilah	5
1.4 Aturan Penamaan dan Penomoran	5
1.5 Referensi	5
1.6 Ikhtisar Dokumen	5
II. DESKRIPSI PERANCANGAN GLOBAL	5
2.1 Rancangan Lingkungan Implementasi	5
2.2 Model Proses Berbasis Objek	5
2.2.1 Diagram Use Case	5
2.2.2 Diagram Class	5
2.2.2 Diagram Activity	5
2.2.3 Diagram Sequence	5
2.3 Model Data	6
2.3.1 Definisi Domain/Type	6
2.3.2 Data Model Logika	6
2.3.3 Data Model Fisik	6
2.3.4 Daftar Tabel Aplikasi	6
III. DESKRIPSI PERANCANGAN RINCI	6
3.1 Deskripsi Rinci Tabel	6
3.1.1 Tabel <Nama-tabel>	7
3.1.2 <Table 2: t_Detail ...>	7
3.2 Deskripsi Proses secara Rinci	7
3.2.1 Spesifikasi Proses <1>	7
3.2.1.1 Spesifikasi tabel input	7
3.2.1.2 Spesifikasi tabel Output	7
3.2.1.3 Spesifikasi Layar Utama	7
3.2.1.4 Spesifikasi Query (jika ada)	8
3.2.1.5 Spesifikasi field data pada layar	8
3.2.1.6 Spesifikasi Function Key / Objek-Objek pada layar	8
3.2.1.7 Spesifikasi layar pesan	8
3.2.1.8 Spesifikasi proses/algoritma (jika ada)	9
3.2.1.9 Spesifikasi State Chart (jika ada)	9
3.2.1.10 Spesifikasi Report	9
3.2.2 <Proses-2>	9
3.3 Dekomposisi Fisik Modul	9

I. PENDAHULUAN

1.1 Tujuan Penulisan Dokumen

Menabung dan Memanage Keuangan adalah Hal yang sangat penting bagi kaum milenial saat ini karena banyak sekali anak muda kaum milenial tidak memperhatikan keuangannya sehingga pengeluaran sering kali lebih besar dari pemasukan apalagi sering kali kita tidak mengetahui uang yang kita keluarkan untuk apa , untuk mengetahui pengeluaran atau pemasukan sehari hari biasanya kita menggunakan note atau dengan mencatat di buku akan tetapi di zaman yang terus berkembang ini kita sudah sangat jarang memegang buku catatan ataupun note karena telah di gantikan oleh teknologi smartphome meskipun sudah menggunakan smartphome untuk note kita masih lupa untuk menabung karena tidak ada reminder atau notifikasi untuk kita.

1.2 Lingkup Masalah

- Bagaimana cara menghemat pengeluaran uang kita?
- Bagaimana cara untuk memudahkan kita mendapatkan barang yang kita inginkan Dengan Menabung?
- Bagaimana cara menghitung kebutuhan yang harus kita keluarkan?

1.3 Definisi dan Istilah

SRS = Software Requirement Spesification (Spesifikasi Perangkat Lunak)

RSA = Rational Software Architect

1.4 Aturan Penamaan dan Penomoran

Tuliskan aturan penomoran dan penamaan yang dipakai dalam dokumen ini jika ada (misalnya aturan penomroan Fungsi/CSU, penomoran modul, penamaan file, dsb)

1.5 Referensi

- Dokumen SRS Pemutar musik
-

1.6 Ikhtisar Dokumen

1.6.1 PENDAHULUAN

1.6.2 PERANCANGAN GLOBAL

1.6.3 PERANCANGAN RINCI

II. DESKRIPSI PERANCANGAN GLOBAL

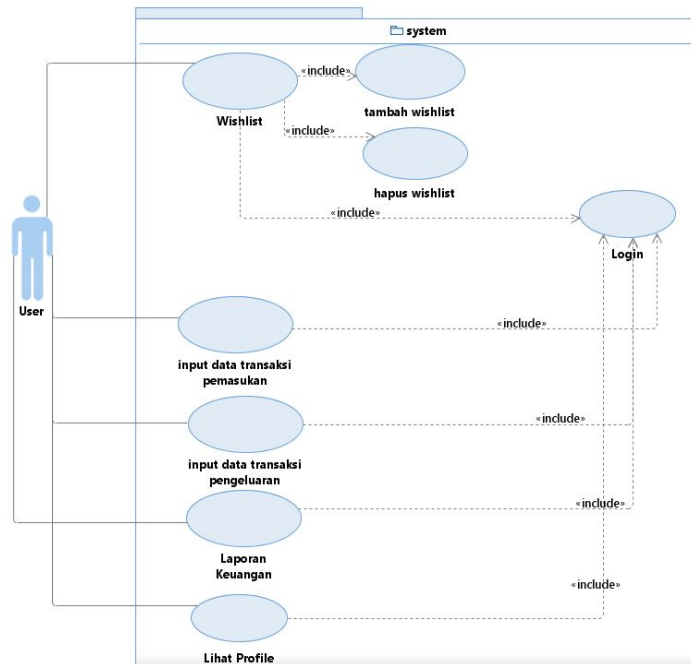
2.1. Rancangan Lingkungan Implementasi

Bahasa pemrograman yang dipakai adalah java dan oprating sistem windows , database yang digunakan Fiebase.

Sebutkan Operating system, DBMS, development tools, filing system, bahasa pemrograman yang dipakai

2.2. Model Proses Berbasis Obyek

2.2.1. Diagram Use Case



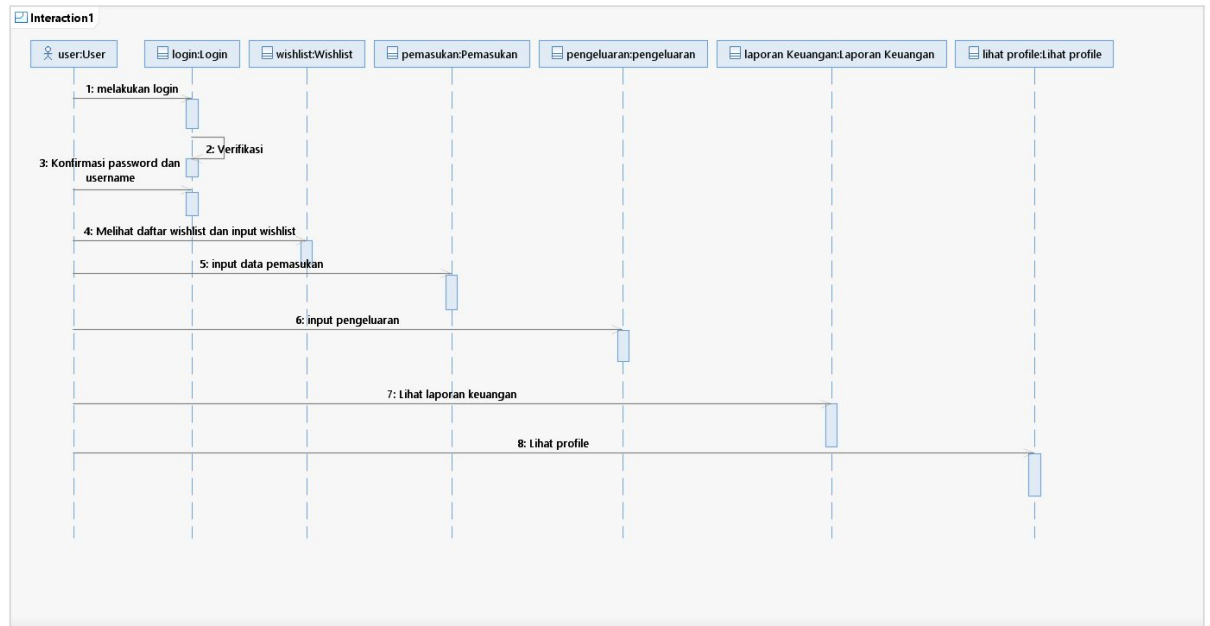
2.2.2. Diagram Class

Gambarkan diagram class yang sesuai dengan fungsionalitas sistem yang dianalisis. Lengkapi dengan atribut, operasi serta asosiasi (lengkap dengan derajat asosiasinya) yang terjadi antar kelas termasuk Inheritance, Aggregation jika memang ada.

2.2.3. Diagram Activity

Berisikan diagram perilaku sistem sesuai dengan pemahaman pada use case narrative untuk menggambarkan proses bisnis yang terjadi pada sistem yang dianalisis

2.2.4. Diagram Sequence



Berisi diagram sequence untuk setiap use case. Tunjukkan hubungan antar obyek dalam class secara terurut melalui operasi pada class diagram dari atas ke bawah.

Untuk setiap nomor proses, buatlah spesifikasi rincinya (layout dan scenario) pada Deskripsi rancangan Rinci

2.4. Model Data

Berisi deskripsi tabel-tabel basis data seperti yang dirancang pada class diagram jika aplikasi berbasis data. Awali dengan daftar tabel basisdata dan deskripsi isinya. Untuk setiap tabel, harus mengandung Nama tabel, jenisnya, Volume, laju, primary key, constraint integrity dengan tabel lain(jika ada). Volume dan laju harus minimal mengandung angka kira-kira.

Boleh berasal dari “dumb” dari database yang digunakan.

2.4.1. Definisi Domain/Type

Sebutkan nama domain (type terdefinisi) yang anda rancang pada aplikasi ini dengan mengisi tabel sebagai berikut

Domain name	Format	Power Designer Type
Rupiah		NUM ...
Kode Transaksi	XX.yy.nnnnn XX = jenis transaksi (surat jalan='SJ', invoice='IV', ...) yy = tahun nnnnn = nomor urut, kembali ke 00001 untuk tahun yang lain.	A11

2.4.2. Data Model Logika

Gambar ini diambil dari Case Tools. Gunakan class diagram untuk melihat conceptual Data Modelnya. Hasilnya diprint di bagian ini

2.4.3. Data Model Fisik

Jika ada, Gambar ini adalah hasil generate diambil dari Case Tools. Gunanya supaya nantinya langsung diterjemahkan menjadi tabel atau bahkan mungkin dipakai untuk membangkitkan tabel secara otomatis.

2.4.4. Daftar Tabel Aplikasi

Awali dengan daftar tabel basisdata, primary key dan deskripsi isinya.

Nama Tabel	Primary key	Data Store	E/R	Deskripsi isi

Untuk setiap tabel, buatlah deskripsi rincinya pada Sub-bab Deskripsi Rinci Tabel

III. DESKRIPSI PERANCANGAN RINCI

3.1. Deskripsi Rinci Tabel

Setiap tabel pada rancangan global, dirinci satu per satu. Bisa di-generate dari CASE-Tool.

i. Tabel <Nama-tabel>

Identifikasi>Nama : t_master
Deskripsi Isi :
Jenis : tabel data induk/referensi/transaksi
Volume :
Laju :
Primary Key :

Id Field	Deskripsi	Tipe & length	Boleh NULL	Default	Keterangan
Id_master		CHAR(8)	NO		
Id_ref					Refer ke t_ref

Catatan: kolom “Boleh NULL” berisi “NO” artinya tidak boleh kosong, berisi “YES” artinya boleh NULL

ii. <Table 2: t_Detail ...>

Buat seperti di atas

3.2. Deskripsi Proses Secara Rinci

Setiap proses sesuai use case pada rancangan global, dirinci satu per satu.

3.2.1. Spesifikasi Proses <1>

Identifikasi>Nama :
Deskripsi Isi :
Jenis : Form Entry columnar/Tabular/Master-Detail
Report Columnar/tabular/Master-Detail
Form berisi dialog/button saja
Proses tanpa layar

3.2.1.1. Spesifikasi Tabel Input

Nama Tabel : 1.
2.

3.2.1.2. Spesifikasi Tabel Output

Nama Tabel : 1.
2.

3.2.1.3. Spesifikasi Layar Utama

Gambarkan layar dan percabangan ke layar lain function key/pilihan yang dilakukan.
Jika layar mengandung field dan label, gambarkanlah pada posisi nya, supaya siap dikoding. Jika ada zoning/frame, gambarkan pula dan jelaskan pada spesifikasi Objek pada layar.

The diagram shows a main screen layout. On the left side, there is a vertical column of four empty rectangular boxes, likely for a menu or list. To the right of this column is a large rectangular area. Inside this area, there is a box labeled 'Objek' with a smaller empty box to its right. At the bottom right of the large area, there is an 'OK' button.

3.2.1.4. Spesifikasi Query (jika ada)

Tuliskan ekspresi Query jika layar didasari dari suatu Query dan bukan langsung dari Tabel:
<nama Query> : Ekspresi

3.2.1.5. Spesifikasi field data pada layar

Catatan:

- Hanya ada isinya jika layar “dibelakang”nya memproses basisdata
- kolom validasi hanya diisi jika layar entry/Update (field ybs bukan hanya tampilan)

Label	Field	Tabel/ Query	I/O	Format	Validasi	Keterangan

3.2.1.6. Spesifikasi Function Key / Objek-Objek pada layar

Id Objek	Jenis	Keterangan
		Diisi dengan penjelasan reaksi sistem, misalnya membuka layar apa, link kemana. Jika menyangkut suatu kode yang cukup rumit, beri nama Prosedur terkait, dan uraikan algoritmanya di bagian Spesifikasi Proses
Button1	Button	Jika diklik, akan mengaktifkan Proses AlgoXXX.
RTF1	RTF Box	Isi Teks yang disimpan pada File xxx
DB1	Data control	Diasosiasikan ke QueryXYZ yang dijelaskan pada bagian Query

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

3.2.1.7. Spesifikasi Layar Pesan

Dialog, pesan, jika ada. Jika ada banyak layar pesan, dapat terdiri dari banyak dialog. Jika tidak ada pesan, tuliskan : Tidak Ada

3.2.1.8. Spesifikasi Proses/Algoritma (Jika Ada)

Kebanyakan layar “standard” tidak ada algoritmanya. Gambarkan menggunakan diagram aktifitas. Berisi transisi dari operasi yang dilalui oleh obyek. Gambarkan diagram aktifitas untuk use case yang mempunyai perilaku dinamis untuk mengetahui alur (flowchart) dari use case. Berikut ini contohnya, jika ada.

Mungkin lebih dari satu, jika ada banyak objek layar yang harus dikode

<Id. Proses > :
Objek terkait :
Event :

Initial State (IS): Tabel ... sudah dibuat strukturnya tetapi mungkin masih kosong
Final State (FS): Menghasilkan tabel TRANSAKSI yang telah ditambahkan satu record
Spesifikasi Proses/algoritma: 1. Cek apakah data yang dimasukkan ada 2. IF belum ada, tambahkan 1 record pada tabel t_petugas ELSE Jika sudah ada, tampilkan pesan kesalahan sesuai layar XXXX (nomor layar)

Catatan : jika satu layar mengandung banyak button, dan untuk satu button ditulis sebuah prosedur, maka akan ada beberapa algoritma

3.2.1.9. Spesifikasi State Chart (jika ada)

Berisi state yang dilalui oleh obyek. Gambarkan diagram state chart untuk use case yang mempunyai perilaku dinamis.

3.2.1.10. Spesifikasi Report

Berikan layoutnya, harus jelas kolom barisnya.

Dapat anda manfaatkan untuk coding (copy/paste sebagai teks)

Jika layar utama tidak terkait kepada Report, maka isi :TIDAK ADA

3.2.2. <Proses-2>

Untuk setiap fungsi, buat detailnya (sub-sub-bab 1-9) seperti di atas

Dst... spt di atas.

3.3. Dekomposisi Fisik Modul

Berisi dekomposisi “fisik” dari modul. Minimal berisi tabulasi dengan kolom: Sub Aplikasi, Modul, Nama File, Input, Output. Sub Aplikasi biasanya dibuat per pengguna. Dibuat per modul.

Berisi struktur direktori dan pengumpulan fungsi menjadi file. Minimal berisi tabulasi dengan kolom: Modul, Proses, Keterangan. Kolom keterangan hanya diisi jika proses tidak tergambar dalam diagram use case. Misalnya untuk proses-proses yang mewakili suatu library umum.

Nama Direktori	Nama File	Nama Modul	Nama Fungsi	Keterangan