

Analisis Kebutuhan

Merupakan proses menemukan, memperbaiki, memodelkan dan menspesifikasikan.

Terdiri dari lima langkah pokok:

1. Identifikasi Masalah
2. Evaluasi dan sintesis
3. Pemodelan
4. Spesifikasi
5. Review

Dalam menemukan Area permasalahan, perlu adanya komunikasi yang intensif dengan user. Hal yang perlu diperhatikan dalam berkomunikasi adalah menghindari salah interpretasi

Pertanyaan pertama memfokuskan pada pengertian dasar permasalahan:

1. Menemukan yang membutuhkan software tersebut:
 - a. Siapa yang membutuhkan sistem (serta personal di belakangnya) ?
 - b. Siapa yang akan menggunakan solusi
 - c. Apa yang akan menjadi keuntungan ekonomis dari solusi yang baik
 - d. Adakan sumber lain dari solusi yang dibutuhkan
2. Bentuk solusi yang diinginkan
 - a. Bagaimana user mengkarakteristikkan suatu output sistem yang baik yang akan dihasilkan oleh solusi yang benar
 - b. Masalah-masalah apa yang akan dicarikan solusinya?
 - c. Lingkungan solusi yang akan digunakan
 - d. Adakah isu atau kendala khusus yang berdampak kepada solusi
3. Efektifitas
 - a. Mendapatkan person yang benar/berhak atas jawaban pertanyaan,
 - b. Apakah pertanyaan yang diajukan relevan dengan permasalahan
 - c. Adakah personal lain yang dapat menambah informasi
 - d. Adakah hal lain yang perlu ditambahkan?

Jenis Kebutuhan:

1. Kebutuhan Fungsional

Pendefinisian layanan yang harus disediakan, bagaimana reaksi sistem terhadap input dan apa yang harus dilakukan sistem pada situasi khusus (Kebutuhan sistem dilihat dari kacamata pengguna)

2. Kebutuhan Non-Fungsional

Kendala pada pelayanan atau fungsi sistem seperti kendala waktu, kendala proses pengembangan, standard, dll. Contoh: kehandalan, waktu respon dan kebutuhan storage. Contoh kendala seperti: Keterbatasan kemampuan peralatan I/O, representasi sistem dll.

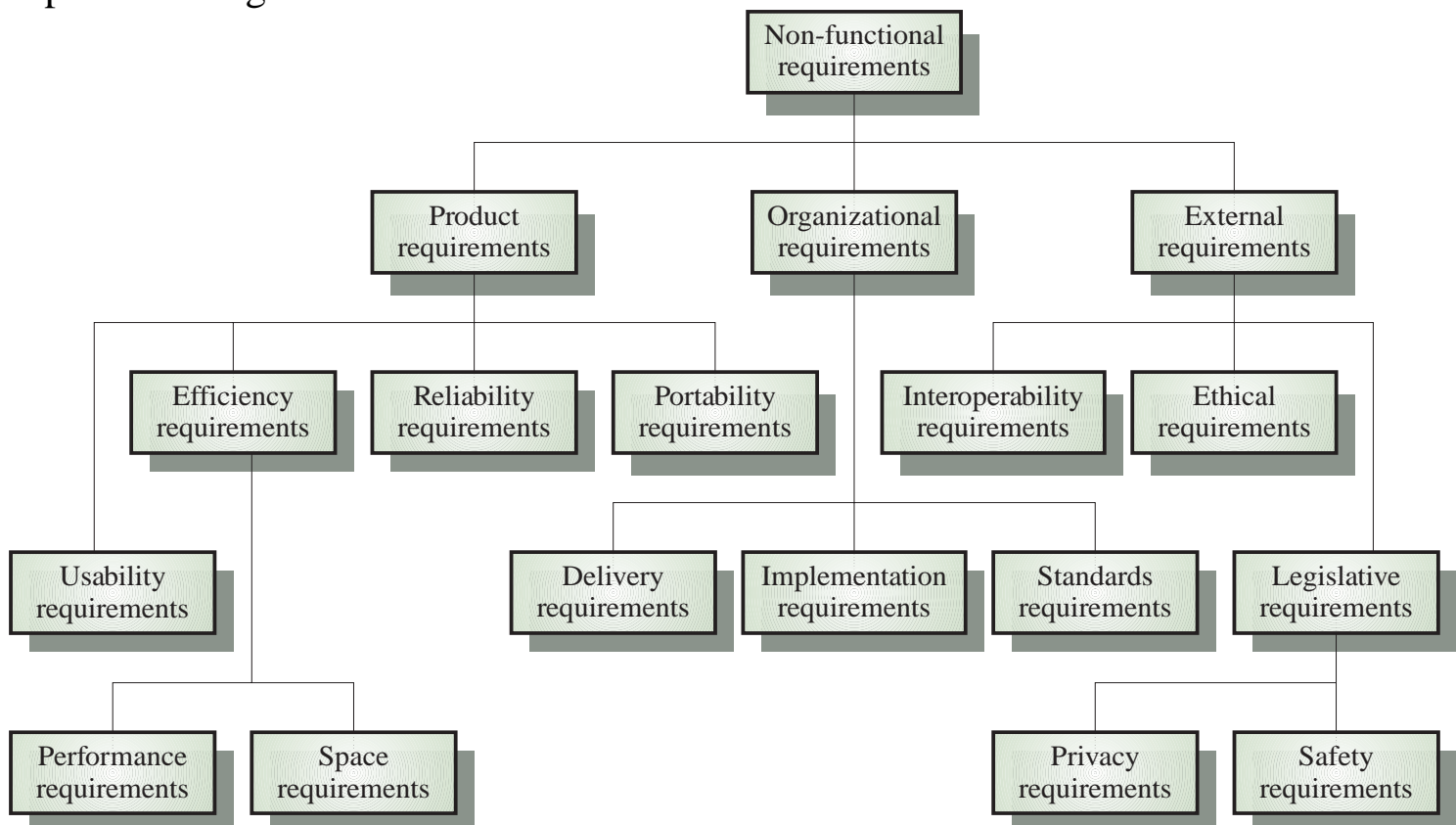
Domain Kebutuhan

Kebutuhan yang berasal dari domain aplikasi sistem dan merefleksikan karakteristik domain

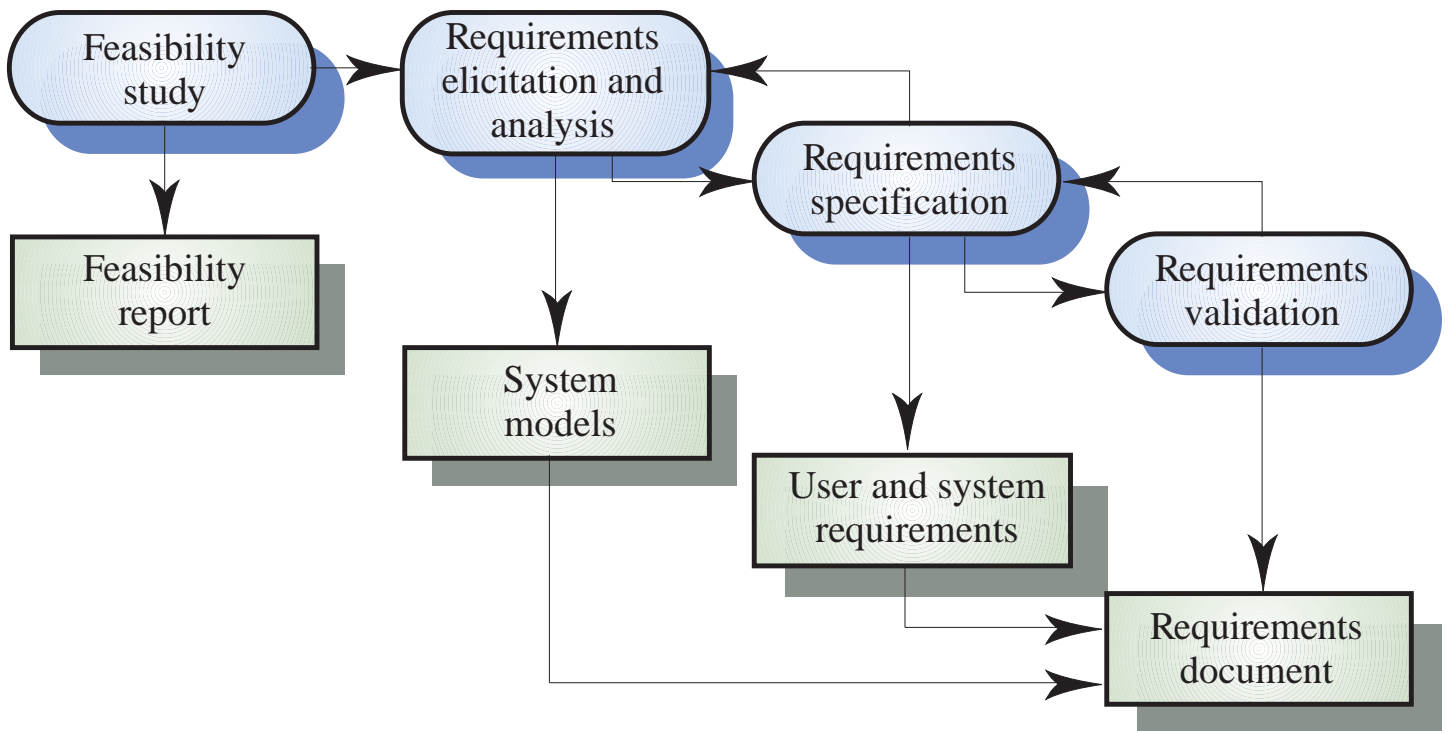
Secara Prinsip, spesifikasi Kebutuhan harus:

1. Lengkap: Mendeskripsikan semua fasilitas yang diinginkan
2. Konsisten: Tidak adanya konflik dan kontradiksi

Tipe Non-Fungsional



Proses Rekayasa Kebutuhan



Studi Kelayakan

Studi Kelayakan memutuskan apakah sistem software yang akan dibuat sudah mencakup seluruh aspek permasalahan

Melakukan studi untuk menguji apakah sistem:

- sudah sesuai dengan tujuan organisasi
- dapat dikembangkan dengan teknologi terkini dan dana yang tersedia
- dapat diintegrasikan dengan sistem lain yang sudah digunakan

Implementasi Studi Kelayakan

Berbasikan pada penilaian informasi (apa yg dibutuhkan), pengumpulan informasi dan penulisan laporan

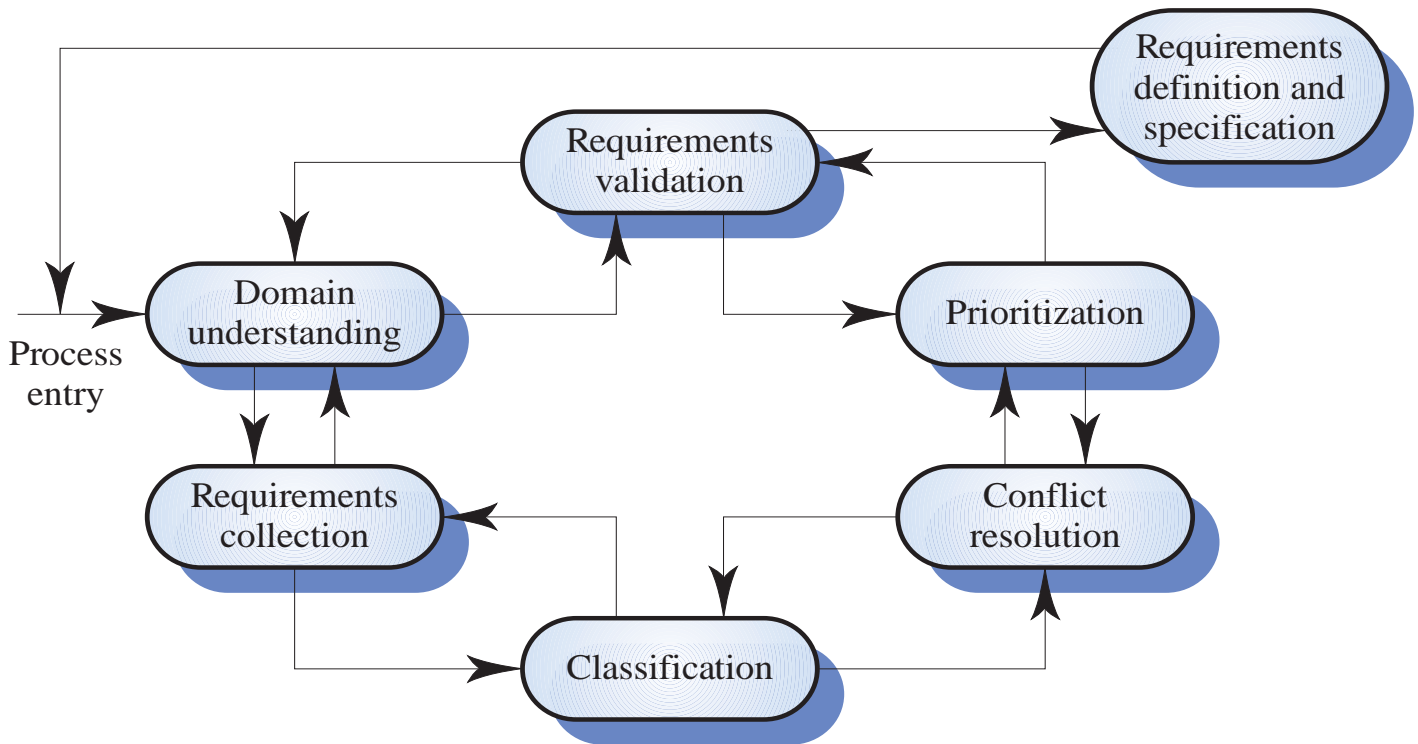
Pertanyaan ke personal di organisasi:

- Apa yang akan terjadi apabila sistem tidak diimplementasikan?
- Masalah proses apa yang ada ?
- Apa yang dapat dibantu oleh sistem ?
- Masalah apa yang akan muncul pada proses Integrasi ?
- Adakah teknologi baru yang dibutuhkan? Skill yang dibutuhkan ?
- Fasilitas apa yang harus didukung oleh sistem ?

Permasalahan pada Analisis Kebutuhan

- Pengguna (stakeholders) tidak mengetahui apa yang mereka butuhkan
- Pengguna menjelaskan kebutuhan dengan cara mereka sendiri sehingga sulit untuk dipahami
- Pengguna yang berbeda memiliki konflik kebutuhan
- Faktor politik dan organisasi yang dapat mempengaruhi kebutuhan sistem
- Perubahan kebutuhan selama proses analisis. Stakeholder baru mungkin akan merubah lingkungan bisnis.

Proses Analisis Kebutuhan



Pemodelan Sistem

Dapat dilakukan dalam beberapa cara, seperti model structural, state machine, state chart, dll

Pemodelan tersebut dapat pula direpresentasikan sebagai formaliasi sudut pandang pengguna (viewpoint-oriented)

Viewpoint-oriented elicitation

Stakeholder merepresentasikan sudut pandang suatu masalah dalam beberapa cara. Analisis Multi perspektif adalah penting jika tidak terdapat suatu cara yang benar untuk menganalisa kebutuhan sistem.

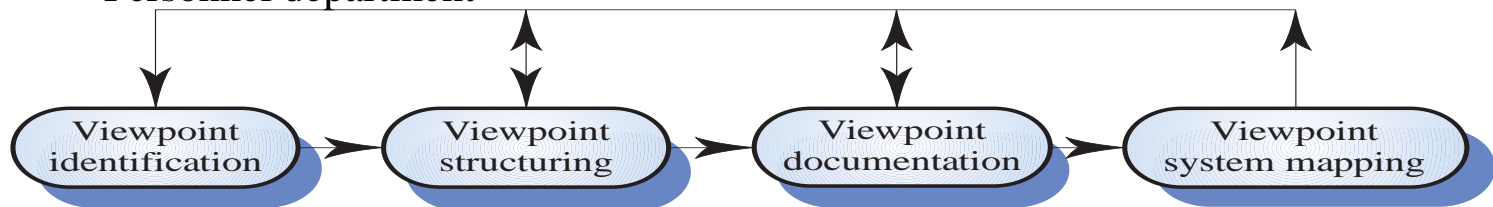
Contoh: Sistem ATM Bank

Sistem ATM dapat menyediakan pelayanan bank secara otomatis

Pelayanan tersebut mencakup: penarikan tunai, pengiriman pesan untuk permintaan layanan, pemensanan, dan transfer.

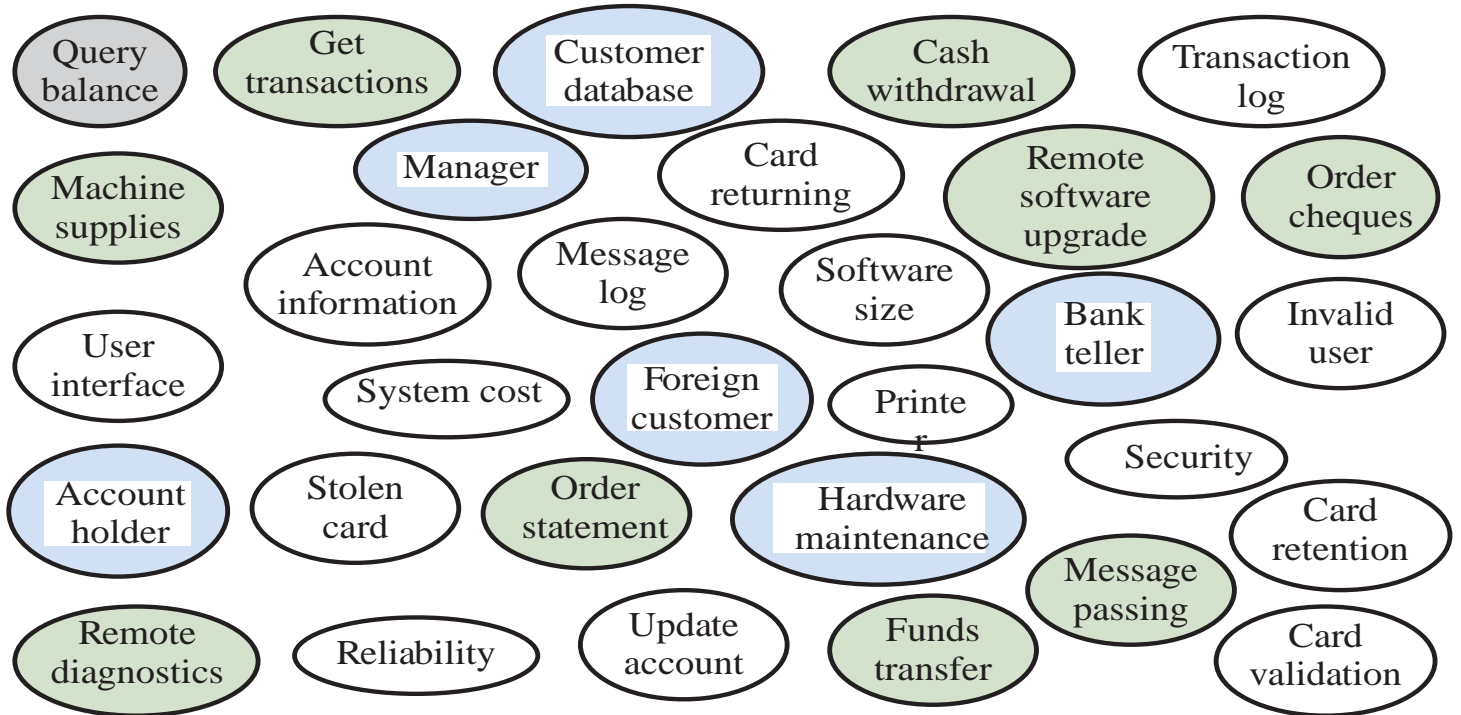
Autoteller viewpoint

- Bank customers
- Representatives of other banks
- Hardware and software maintenance engineers
- Marketing department
- Bank managers and counter staff
- Database administrators and security staff
- Communications engineers
- Personnel department



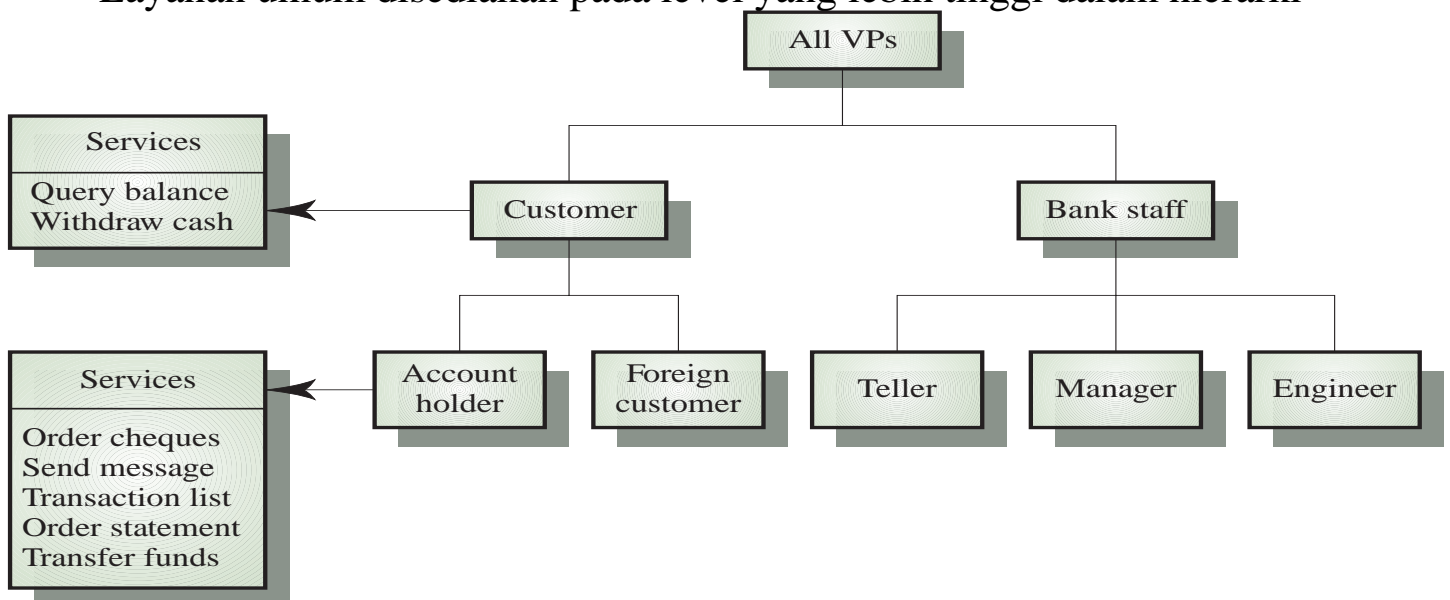
Identifikasi Viewpoint:

- Menemukan viewpoint sebagai penerima layanan sistem dan mengidentifikasikan layanan yang disediakan untuk masing-masing viewpoint
-



Pembentukan Struktur Viewpoint

- Mengelompokkan viewpoint yang saling berhubungan secara hierarki. Layanan umum disediakan pada level yang lebih tinggi dalam hierarki



Dokumentasi Viewpoint

- Memperbaiki deskripsi viewpoint dan layanan yang teridentifikasi
- Viewpoint system mapping
- Transformasi analisis ke perancangan berorientasi objek

Viewpoint Service Information

ACCOUNT HOLDER

Service list
Withdraw cash Query balance Order cheques Send message Transaction list Order statement Transfer funds

FOREIGN CUSTOMER

Service list
Withdraw cash Query balance

BANK TELLER

Service list
Run diagnostics Add cash Add paper Send message

Bentuk Standard VORD

Viewpoint template

Reference:	Customer
Attributes:	Account number PIN Start transaction
Events:	Select service Cancel transaction End transaction
Services:	Cash withdrawal Balance enquiry
Sub-VPs:	Account holder Foreign customer

service template

Reference:	Cash withdrawal
Rationale:	To improve customer service and reduce paperwork
Specification:	Users choose this service by pressing the cash withdrawal button. They then enter the amount required. This is confirmed and, if funds allow, the balance is delivered.
VPs:	Customer
Non-funct. requirements:	Deliver cash within 1 minute of amount being confirmed
Provider:	<i>Filled in later</i>

Skenario

Penggambaran bagaimana sistem akan digunakan

Membantu dalam menemukan kebutuhan dengan mempermudah dalam penggambaran proses dibandingkan pernyataan abstrak kebutuhan sistem

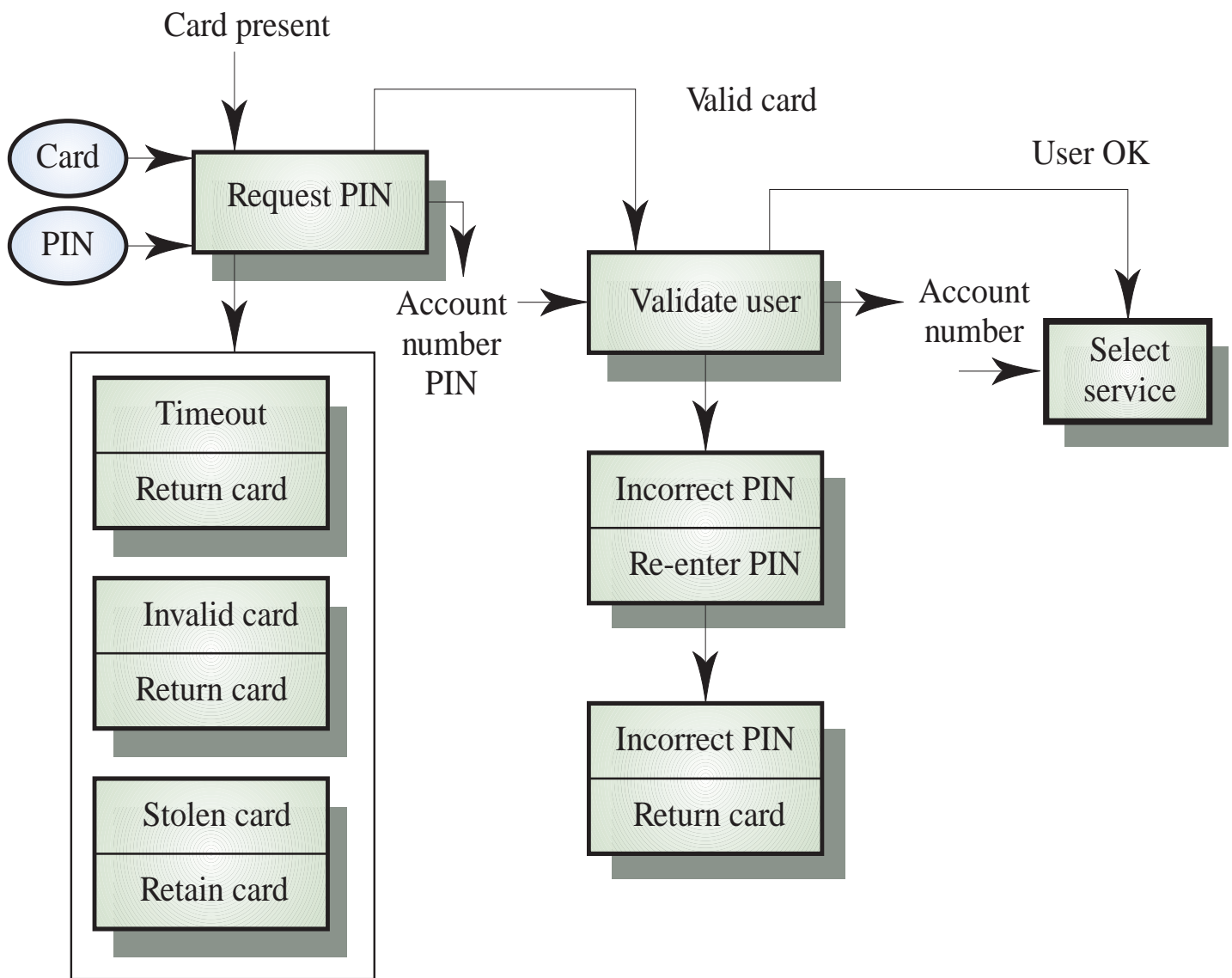
Menambahkan detail ke outline deskripsi kebutuhan

Deskripsi dalam Skenarion

- Sistem State pada awal scenario
- Alur Normal kejadian-kejadian di sistem
- Apa yang dapat berkembang dan bagaimana menanganinya
- Aktifitas-aktifitas yang bersamaan terjadi
- System state setelah proses selesai

Skenarion Kejadian

- Skenario kejadian dapat digunakan untuk menggambarkan bagaimana sistem merespon ke suatu kejadian tertentu seperti awal transaksi
- VORD dapat berupa diagram untuk menggambarkan scenario kejadian
 - Data yang dikirim dan disediakan
 - Kontrol Informasi
 - Pengecualiaan Proses
 - Kejadian berikutnya



Notasi:

Elips menyatakan data yang disediakan oleh dan dikirim ke viewpoint

Data keluar dari sisi kanan setiap kotak

Eksepsi ditunjukkan di bawah masing-masing box

Nama kejadian berikutnya berada di box dengan garis panah tebal

Pada contoh di atas, eksepsi adalah:

- Timeout: Pelanggan salah memasukkan nomor PIN selama waktu yang diberikan
- Invalid Card: Kartu tidak dikenal oleh sistem dan dikembalikan
- Stolen Card: Kartu sudah diregister sebagai kartu yang sudah dicuri/hilang dan akan diambil oleh sistem (tidak dikembalikan)

Validasi Kebutuhan

- Bertujuan untuk meyakinkan bahwa kebutuhan yang sudah didefinisikan sesuai dengan yang diinginkan pengguna
- Menghindari Kesalahan pendefinisian kebutuhan karena akan menyebabkan penambahan biaya yang besar
 - Memperbaiki definisi kebutuhan setelah software dikirim akan menyebabkan peningkatan biaya hingga 100 kali.

Pengujian Pendefinisian Kebutuhan

- Validasi. Apakah sudah sesuai dengan yang diinginkan
- Konsistensi. Adakah konflik dengan kebutuhan lainnya
- Lengkap: Apakah sudah termasuk semua fungsi yang dibutuhkan
- Realisasi: Dapatkan kebutuhan diimplementasikan ke dana dan teknologi yang tersedia
- Dapat diverifikasi: Dapatkah spesifikasi kebutuhan dicek

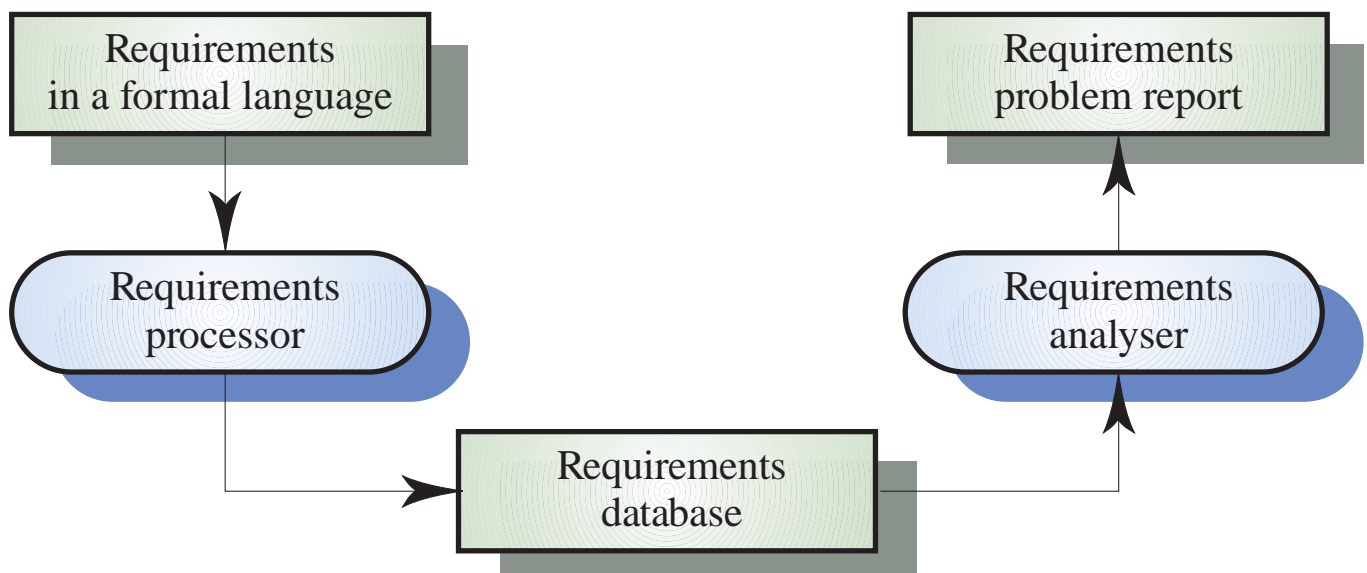
Teknik Validasi Kebutuhan

Review:

Prototyping

Test-Case Generator

Analisis Konsistensi Otomatis



Managemen Perubahan Kebutuhan



Outline Spesifikasi Kebutuhan Software

1. Pendahuluan
 - a. Referensi Sistem
 - b. Deskripsi Umum Sistem
 - c. Kendala Proyek Pengembangan Software
2. Deskripsi Informasi
 - a. Informasi representasi Alur
 - i. Alur Data
 - ii. Alur Kontrol
 - b. Representasi Isi Informasi
 - c. Deskripsi Interface Sistem
3. Deskripsi Fungsional
 - a. Partisi Fungsional
 - b. Deskripsi Fungsional
 - i. Deskripsi proses secara naratif
 - ii. Keterbatasan Sistem
 - iii. Performa yang dibutuhkan
 - iv. Perancangan kendala
 - v. Support diagram
 - c. Deskripsi Kontrol
 - i. Spesifikasi Kontrol
 - ii. Perancangan Kendala
4. Deskripsi Lingkungan
 - a. System State
 - b. Events dan Aksi
5. Kriteria Validasi
 - a. Performance Bound
 - b. Kelas Test
 - c. Respon Software yang diharapkan
 - d. Pertimbangan-pertimbangan khusus
6. Daftar Kepustakaan
7. Appendiks