

Pengumpulan, Dokumentasi dan Pemodelan Kebutuhan

Mia Fitriawati, M.Kom

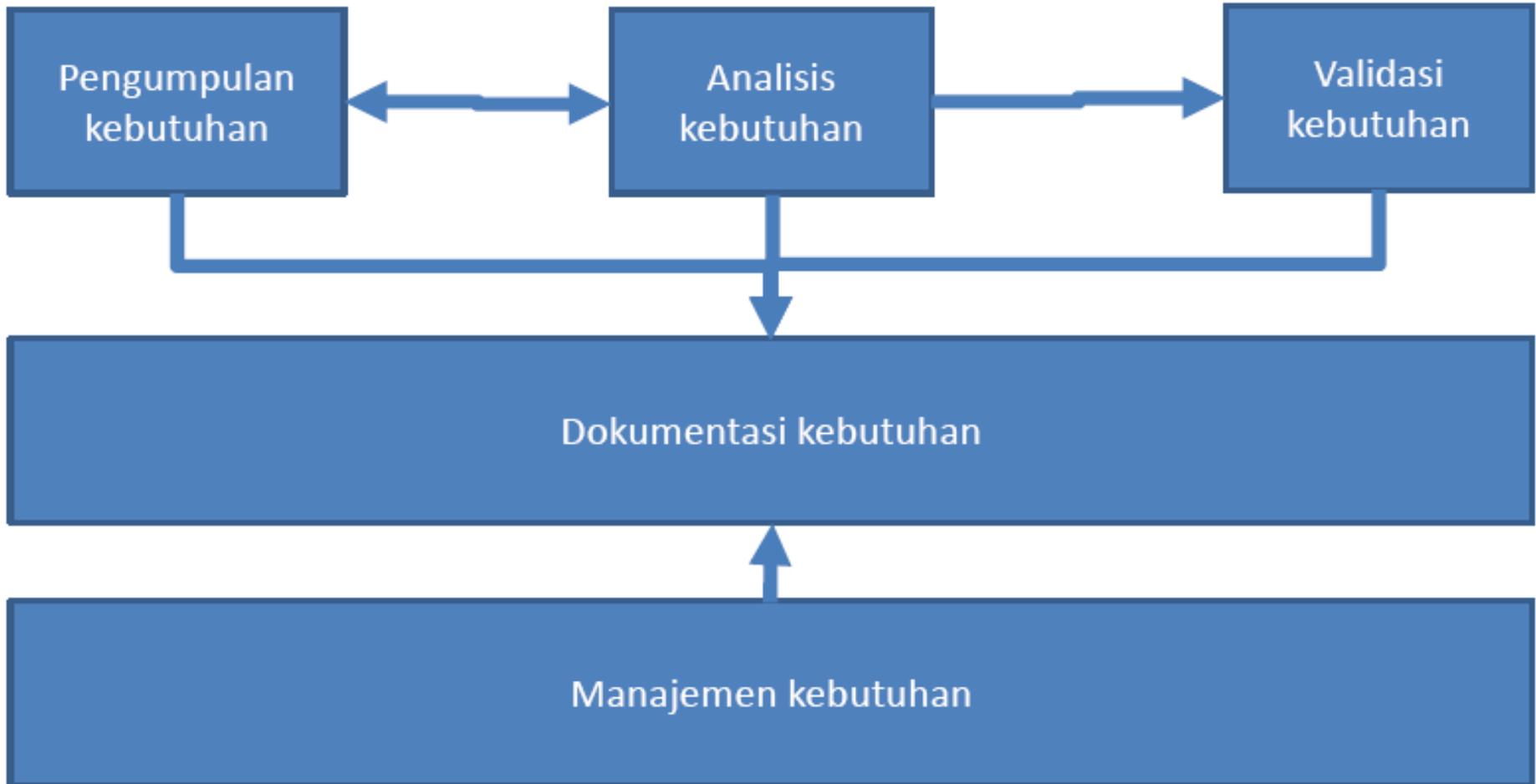
Pendahuluan

- ▶ Kebutuhan dianggap oleh pengguna sebagai suatu hal yang sederhana dalam pengembangan sistem baru.
- ▶ Di sisi lain kebutuhan adalah aspek paling bermasalah yang seringkali tidak terdefinisi dengan baik di awal proyek.
- ▶ Penerapan rekayasa kebutuhan dapat membantu memperbaiki masalah ini.

Penyebab Kegagalan Proyek SI

- ▶ Banyak kesalahan (lebih dari 80%) pada saat tahap analisis kebutuhan
- ▶ Sedikit kesalahan (kurang dari 10%) pada saat tahap pengembangan
- ▶ Kebanyakan waktu proyek dialokasikan untuk tahap pengembangan dan pengujian
- ▶ Kurang dari 12% waktu proyek dialokasikan untuk tahap analisis kebutuhan
- ▶ Ketidaksesuaian sistem yang dikembangkan dengan strategi dan tujuan bisnis
- ▶ Manajemen kebutuhan yang buruk.

Proses Rekayasa Kebutuhan



Proses Rekayasa Kebutuhan

- ▶ Pihak–pihak yang terlibat dalam proses rekayasa kebutuhan:
 1. Perwakilan dari Perusahaan (business representative) yaitu Project Sponsor, Domain Expert, dan Business User.
 2. Tim Proyek (project team) yaitu Project Manager, Business Analyst, dan Developer.

1. Pengumpulan Kebutuhan

- ▶ Pengetahuan EXPLICIT: pengetahuan mengenai prosedur dan data yang ada di depan mata serta mudah diartikulasikan.
- ▶ Pengetahuan TACIT: aspek lain dalam pekerjaan yang tidak dapat diartikulasikan atau dijelaskan oleh stakeholder.

Pengetahuan TACIT vs EXPLICIT

	Pengetahuan TACIT	Pengetahuan EXPLICIT
Individu	Keahlian, nilai, pengetahuan yang didapat begitu saja, intuisi	Definisi tugas, uraian kerja, target, volume dan frekuensi
Perusahaan	Norma, latar belakang, budaya, komunitas praktek, sejarah organisasi	Prosedur, proses, database pengetahuan, panduan, laporan perusahaan

Membangun Daftar Kebutuhan

Kebutuhan	Sumber	Komentar
1. Perusahaan perlu mengurangi jumlah kertas digunakan antar departemen	Budi	Kebutuhan yang real?
2. Sistem harus menangkap pembayaran pelanggan	Budi	
3. Berkas pelanggan harus diproteksi <i>password</i>	Charli	Non-fungsional? Apakah ini solusi?
4. Kita memerlukan informasi lebih cepat	Ani	Kuantifikasikan! Informasi apa? Skala waktunya? Apa yang salah saat ini?
5. Kita harus merekam detail pelanggan serta dapat mengubah dan menghapus pelanggan lama	Ani	Perlu dipecah menjadi tiga kebutuhan?

2. Analisis Kebutuhan

- ▶ Memastikan bahwa semua kebutuhan yang diidentifikasi pada tahap pengumpulan telah dibuat menjadi kebutuhan yang jelas, terorganisir, dan terdokumentasi.
- ▶ Pada tahap ini Analisis memeriksa daftar secara detil untuk dapat membuat dokumentasi kebutuhan untuk memulai proyek.

2. Analisis Kebutuhan

- ▶ Aspek yang perlu dipertimbangkan dalam Analisis Kebutuhan:
 1. Kategorisasi kebutuhan menjadi kategori umum, teknik, fungsional, dan non-fungsional.
 2. Menerapkan filter untuk memastikan kebutuhan telah jelas.

Filter Kebutuhan

- ▶ Memeriksa duplikasi kebutuhan
- ▶ Memisahkan kebutuhan yang digabung
- ▶ Memeriksa kebutuhan yang diperlukan
- ▶ Evaluasi kelayakan
- ▶ Menghilangkan konflik
- ▶ Memeriksa untuk solusi
- ▶ Mengkonfirmasi kualitas

3. Validasi Kebutuhan

- ▶ Perlu konfirmasi Perwakilan dari Perusahaan bahwa dokumen kebutuhan telah menyajikan pernyataan akurat mengenai kebutuhan
- ▶ Tim peninjau terdiri dari perwakilan:
 - Business sponsor – Developer
 - Business owner – Tester
 - Domain expert – Project office

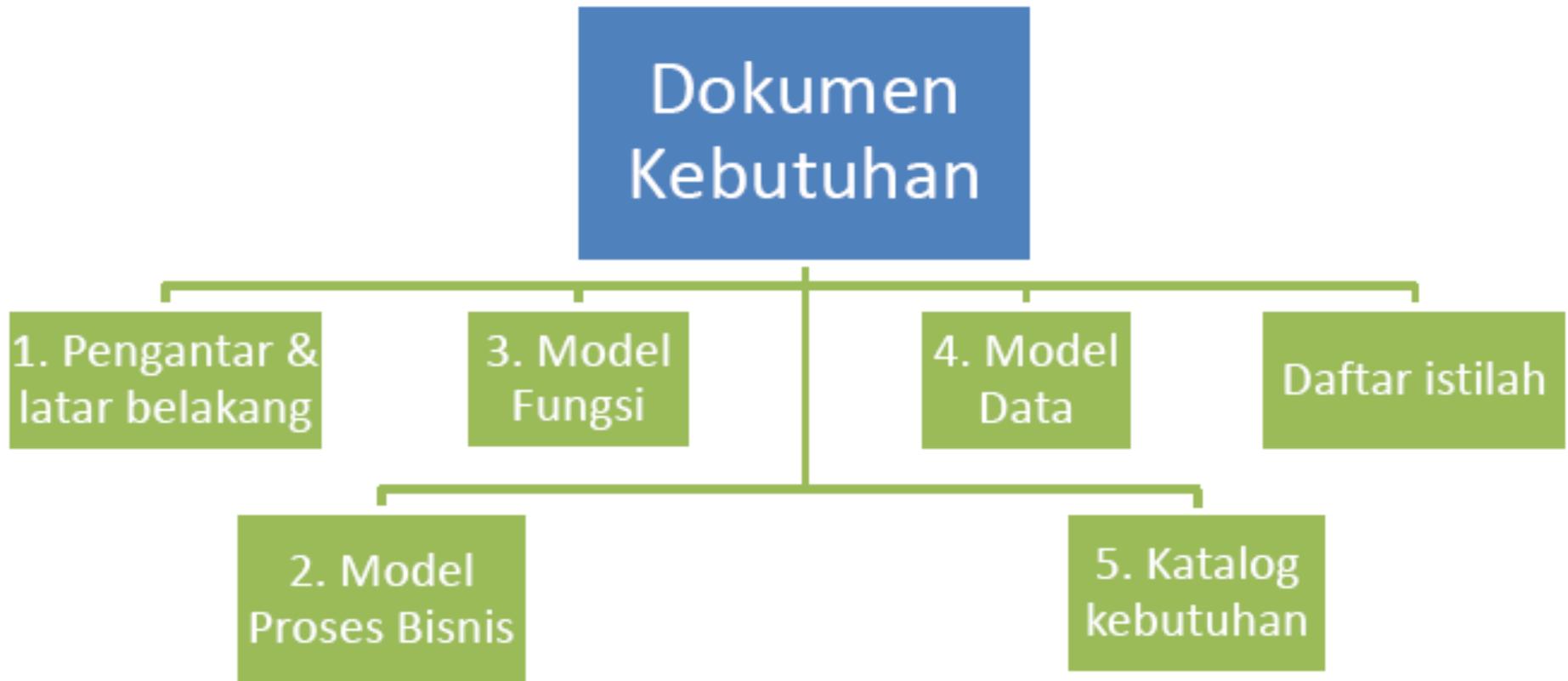
Hasil Tinjauan

1. Dokumen kebutuhan dinyatakan sebagai kebutuhan bisnis yang memuaskan.
2. Dokumen kebutuhan memerlukan sejumlah perubahan yang setelah selesai kemudian disetujui oleh business sponsor.
3. Dokumen kebutuhan memerlukan pengerjaan ulang yang signifikan dan harus ditinjau lagi setelah selesai.

4. Dokumentasi Kebutuhan

- ▶ Pentingnya dokumentasi:
 1. Memungkinkan komunikasi antara tim proyek dan memastikan semua kebutuhan yang berkaitan telah konsisten.
 2. Menyediakan dasar bagi business manager untuk memvalidasi dokumentasi yang secara tepat mencatat apa yang mereka butuhkan terhadap solusi yang diberikan.
 3. Pekerjaan lanjutan untuk mengembangkan dan menguji solusi bisnis akan menggunakan dokumentasi ini untuk memberikan masukan.

Struktur dan Content Dokumen Kebutuhan



5. Manajemen Kebutuhan

- ▶ Manajemen kebutuhan digunakan sebagai pendekatan sistematis untuk mendapatkan, mengorganisasi, mendokumentasikan dan mengatur perubahan kebutuhan aplikasi perangkat lunak.
- ▶ Selain itu manajemen kebutuhan juga memastikan pengembang perangkat lunak akan memecahkan permasalahan dan membangun sistem yang tepat.

Manajemen Kebutuhan

Komponen manajemen kebutuhan:

Identifikasi
kebutuhan

Referensi-silang
kebutuhan

Sumber dan
kepemilikan
kebutuhan

Dukungan
perangkat lunak

Kendali
perubahan

Manajemen
konfigurasi

Pemodelan Kebutuhan

- ▶ Mendokumentasikan persyaratan dari sudut pandang pengguna dengan sebuah cara yang dapat mereka pahami, akan mendorong keterlibatan pengguna, yang tentunya mempertinggi kemungkinan

Pemodelan Kebutuhan

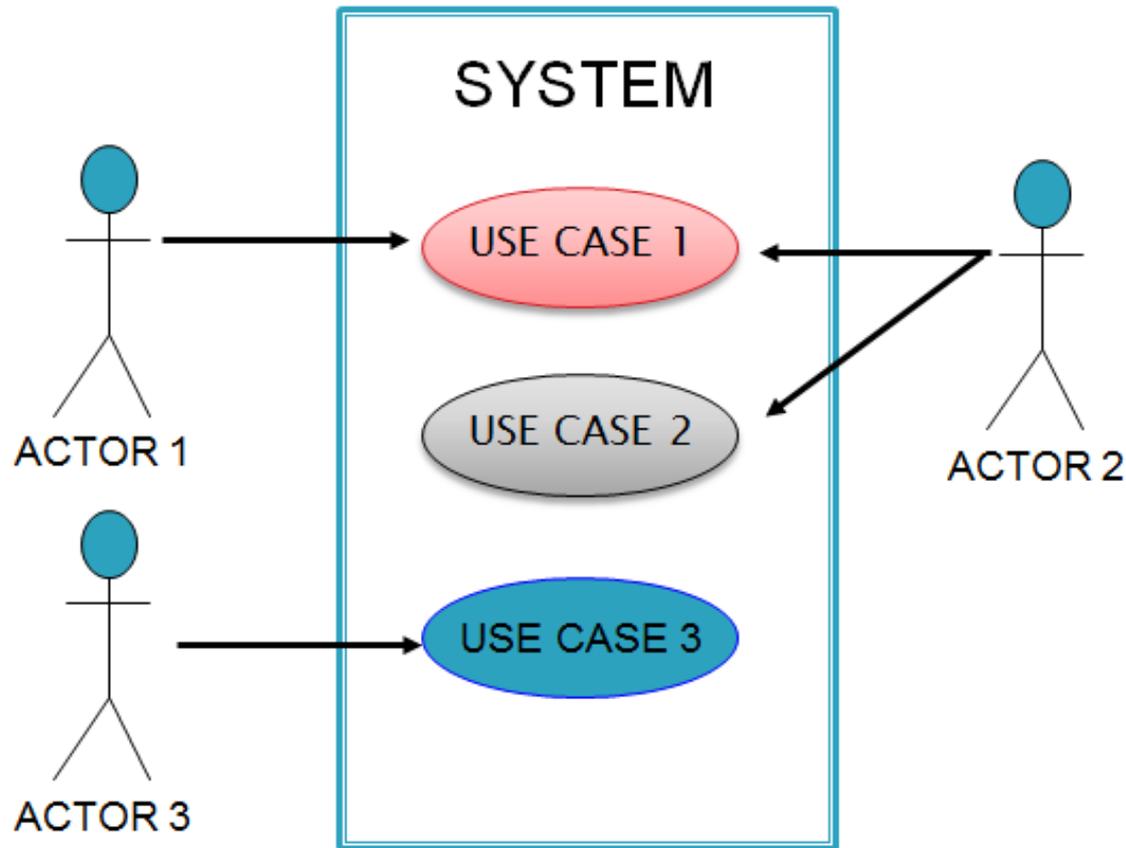
- ▶ Salah satu pemodelan kebutuhan adalah penggunaan Use case Diagram.
- ▶ Use-case diagram merupakan model diagram UML yang digunakan untuk menggambarkan requirement fungsional yang diharapkan dari sebuah sistem.
- ▶ Use-case diagram menekankan pada “siapa” melakukan “apa” dalam lingkungan sistem perangkat lunak akan dibangun.

Pemodelan Kebutuhan

- ▶ Ada dua alat utama yang digunakan saat menyajikan pemodelan Use Case:
 1. Use Case Diagram / Diagram Use Case.
 2. Use Case Narrative / Naratif Use Case

Pemodelan Kebutuhan

▶ Contoh Use Case Diagram



Element Use Case Diagram

▶ Use case

Urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal.

Disajikan secara grafis dengan elips horizontal dengan nama use case muncul di atas, bawah, atau di dalam elips tersebut.



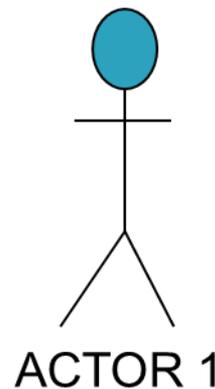
USE CASE

Element Use Case Diagram

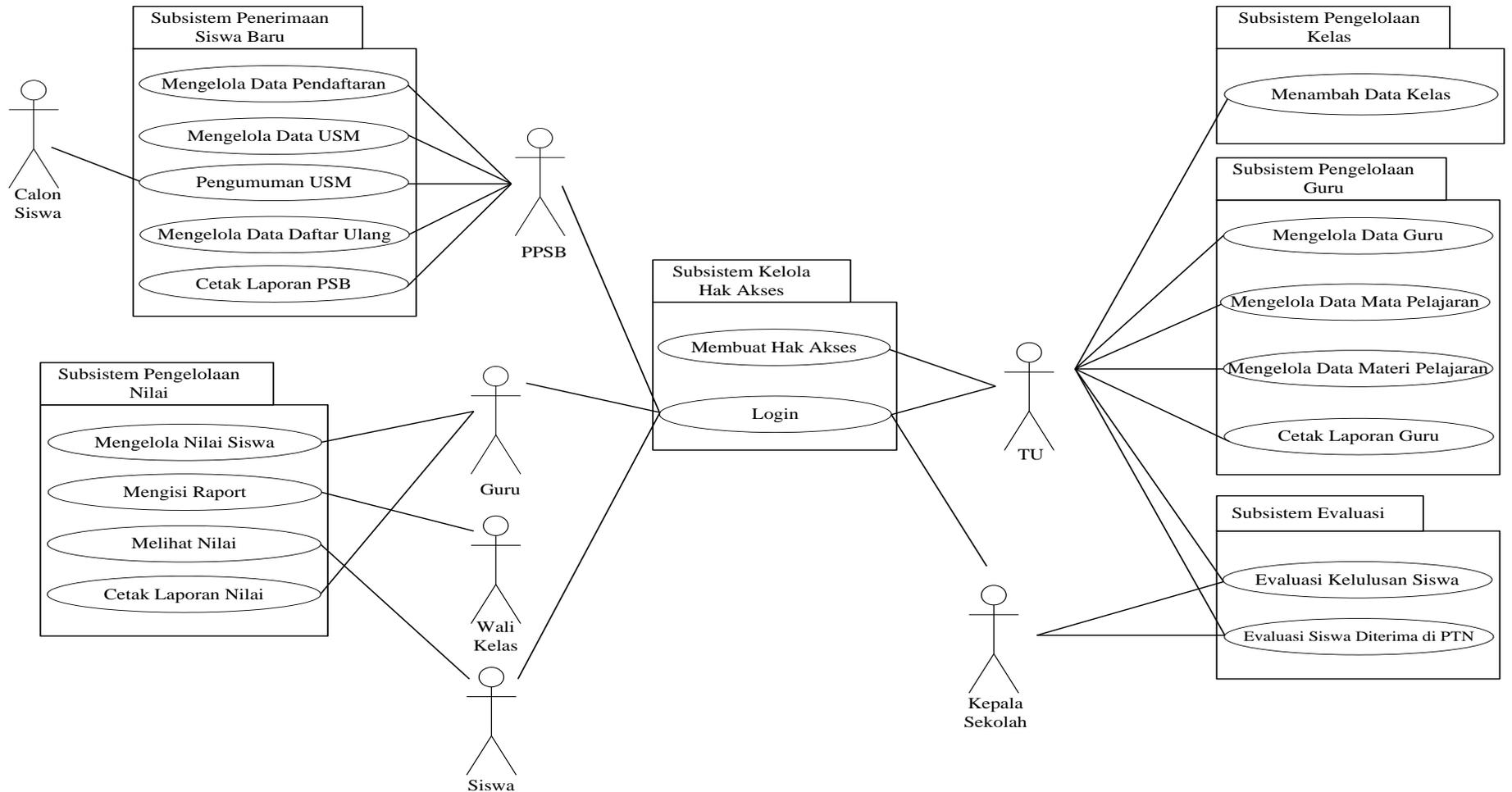
▶ Actor

Use case diawali atau dipicu oleh pengguna eksternal dinamakan actor / pelaku.

Actor adalah segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi



Contoh Use Diagram



Element Use Case Diagram

▶ Ada 4 macam tipe pelaku:

1. *Primary business actor* (pelaku bisnis utama)
2. *Primary system actor* (pelaku sistem utama)
3. *External server actor* (pelaku server eksternal)
4. *External receiving actor* (pelaku penerima eksternal)

Relasi Antar Use Case

Pada diagram use case, hubungan digambarkan sebagai sebuah garis antara dua simbol. Pemaknaan hubungan berbeda-beda tergantung bagaimana garis tersebut digambarkan dan tipe simbol yang digunakan untuk menghubungkan garis tersebut.

Relasi Antar Use Case

- ▶ Association (gabungan) terdiri dari 2:
 1. Mengindikasikan bahwa use case diimitasi oleh pelaku di ujung lain dari garis.
 2. Mengindikasikan interaksi antara use case dan server eksternal atau pelaku penerima.



Relasi Antar Use Case

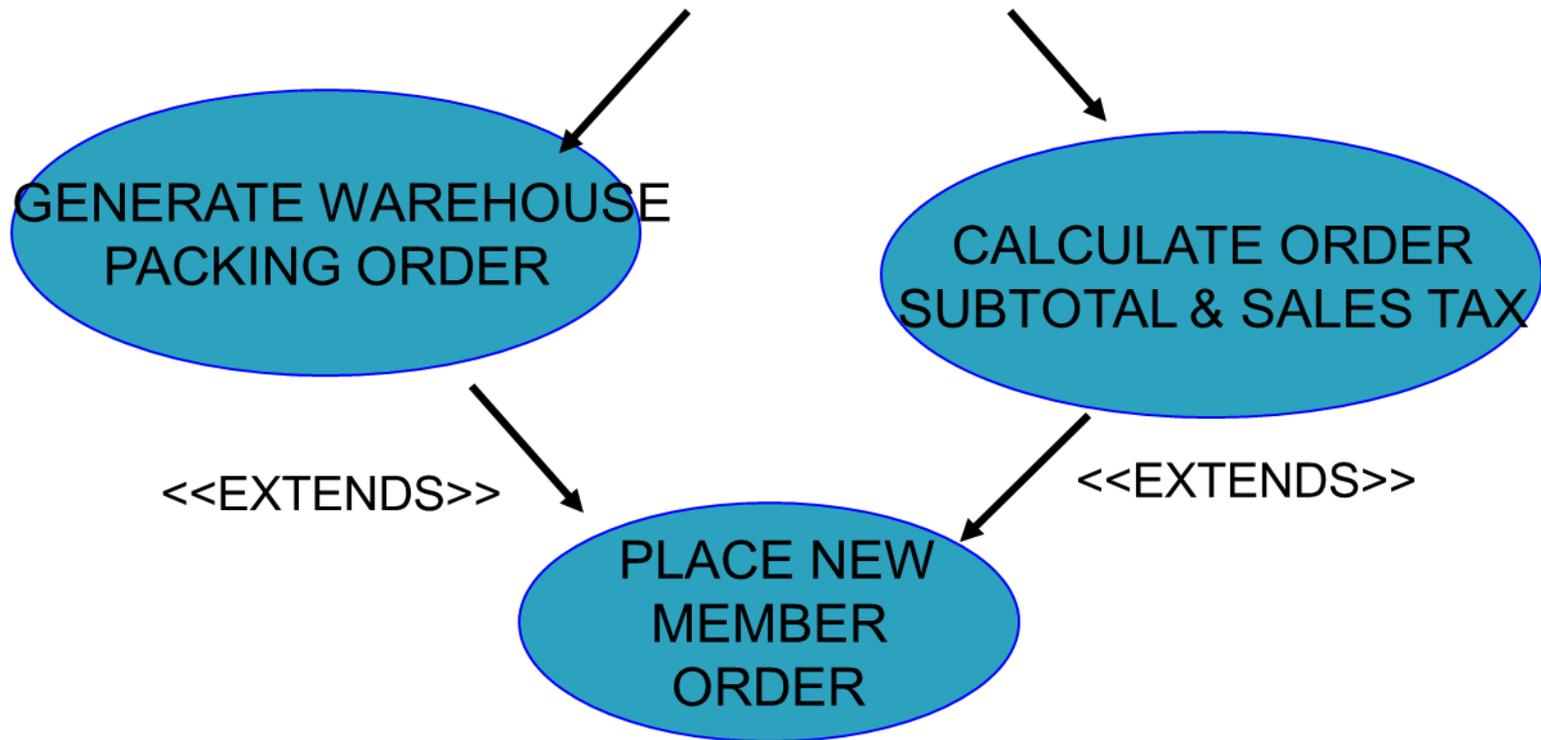
▶ Extension Use Case

Adalah use case yang terdiri dari langkah yang diekstraksi dari use case yang lebih kompleks untuk menyederhanakan masalah orisinal dan karena itu memperluas fungsinya.

Hubungan antara extension use case dan use case yang diperluas disebut **extend relationship** <<extends>>

Relasi Antar Use Case

► Contoh Extension Use Case



Relasi Antar Use Case

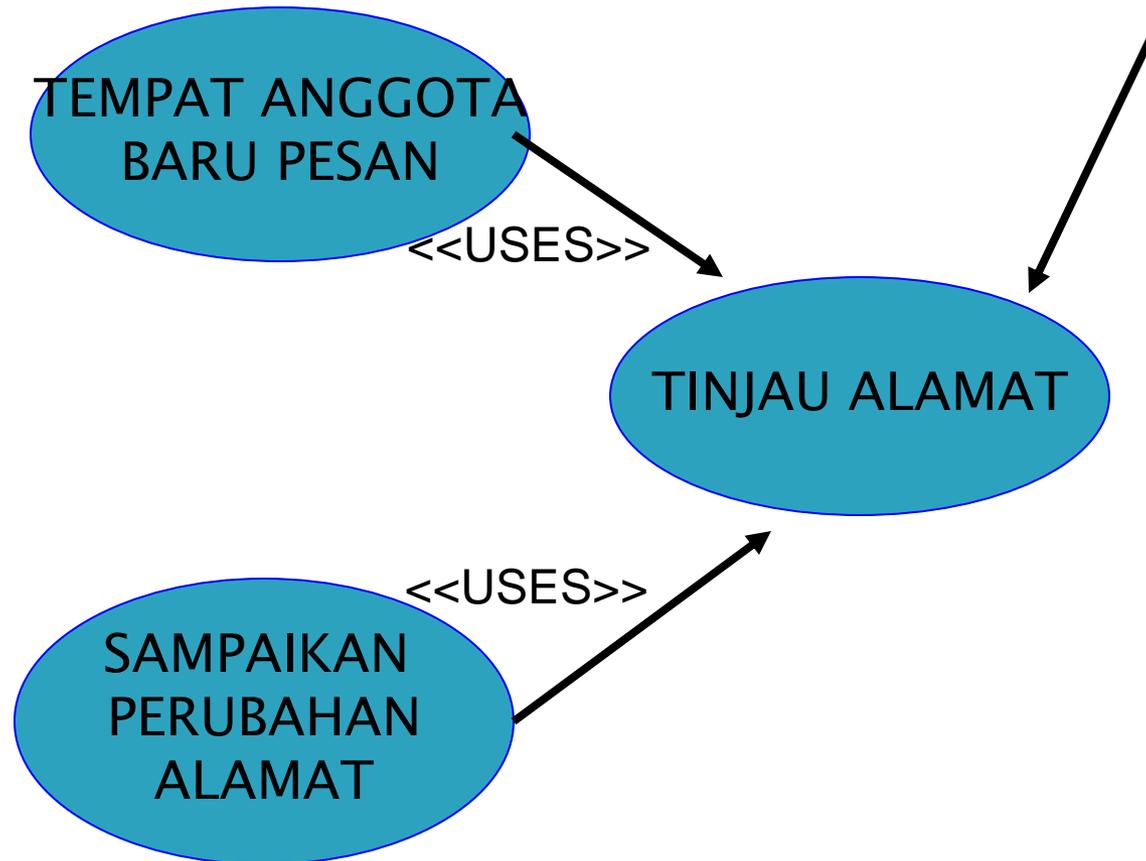
▶ *Abstract Use Case*

Use case yang mengurangi redundansi antara dua atau lebih use case lain dengan menggabungkan langkah-langkah yang biasa ditemukan pada use case tersebut.

Hubungan di antara abstract use case dan use case yang menggunakannya disebut `uses <<uses>>`.

Relasi Antar Use Case

► Contoh *Abstract Use Case*



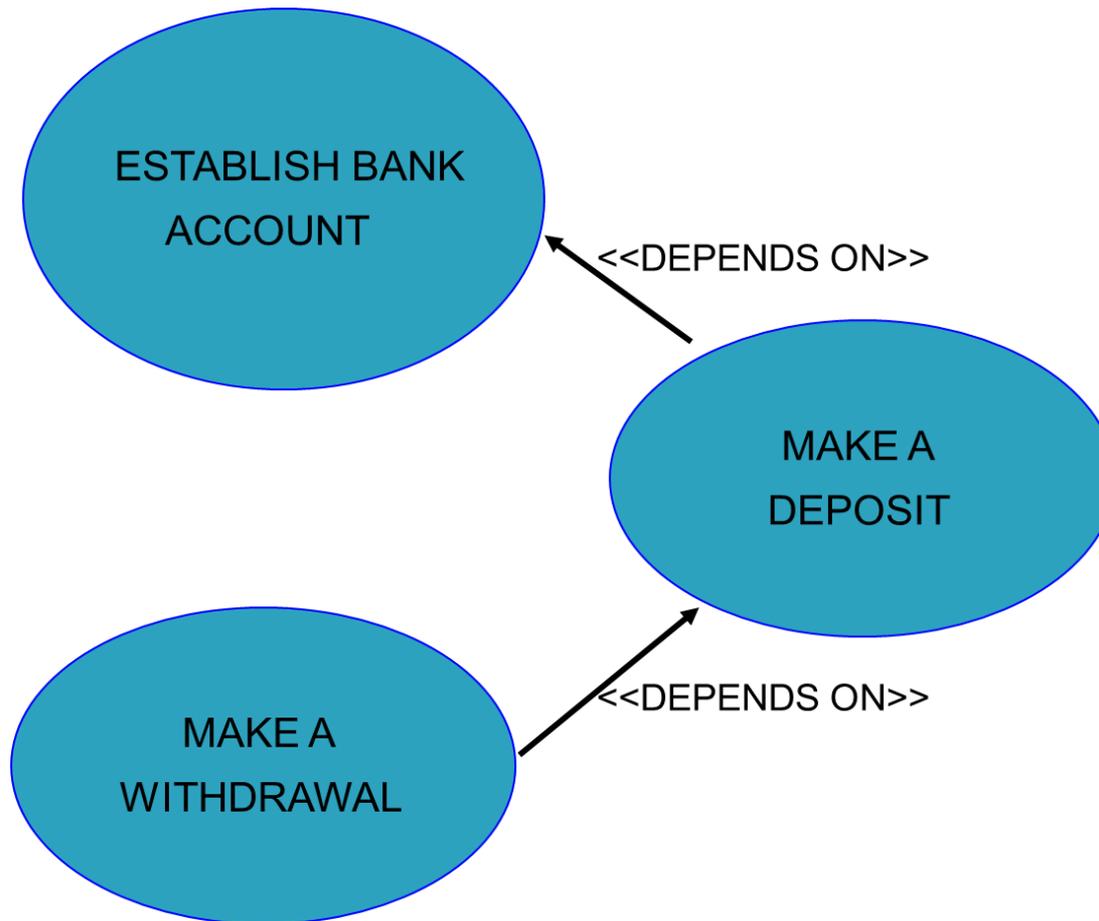
Relasi Antar Use Case

▶ *Depends On*

Manajer proyek atau developer utama sangat perlu mengetahui use case mana yang memiliki **ketergantungan** pada use case lain untuk menetapkan rangkaian use case yang perlu dikembangkan.

Relasi Antar Use Case

▶ Contoh *Depends On*



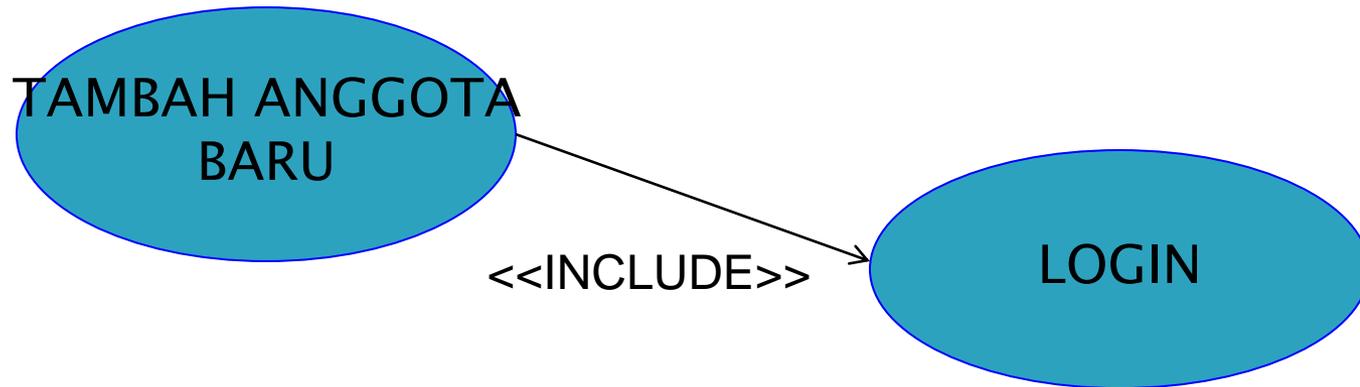
Relasi Antar Use Case

▶ *Include*

- ▶ Relasi include memungkinkan terjadinya penambahan perilaku (behaviour) ke dalam use case awal yang pada dasarnya use case ini tidak dapat berdiri sendiri tanpa adanya penambahan use case, dan use case awal tidak akan lengkap tanpa adanya use case tambahan ini.

Relasi Antar Use Case

▶ Contoh *Include*



Handwritten scribbles at the top of the page.

Just a note to say

**Thank
You**