

Materi 5

# REQUIREMENTS ENGINEERING

# Requirements Engineering

- ⦿ Membantu Software Engineer lebih memahami masalah yang mereka coba pecahkan.
- ⦿ Menghasilkan pemahaman tertulis untuk masalah pelanggan.
- ⦿ Dimulai sejak aktivitas komunikasi dalam rekayasa perangkat lunak dan dilanjutkan dengan aktivitas pemodelan.

# What is a Requirement ?

- ⦿ Ini dapat berkisar dari pernyataan abstrak tingkat tinggi dari layanan atau kendala sistem untuk spesifikasi fungsional matematika rinci.
- ⦿ Ini tidak bisa dihindari karena persyaratan dapat melayani fungsi ganda
  - Dapat menjadi dasar tawaran untuk kontrak - karena itu harus terbuka untuk interpretasi;
  - Dapat menjadi dasar untuk kontrak itu sendiri - sehingga harus didefinisikan secara rinci;
  - Kedua pernyataan ini dapat disebut kebutuhan.

# Types of Requirement

## ⦿ User requirements

- Pernyataan dalam bahasa natural dengan diagram dari layanan system yang diberikan dan kendala operasional. Dibuat untuk pelanggan

## ⦿ System requirements

- Sebuah dokumen terstruktur yang menetapkan deskripsi rinci dari fungsi sistem, layanan dan kendala operasional. Mendefinisikan apa yang harus dilaksanakan sehingga dapat menjadi bagian dari kontrak antara klien dan kontraktor.

# Perbedaan User dan System Requirement

PARAMETER PEMBANDING	USER REQUIREMENT	SYSTEM REQUIREMENT
<b>Kedetailan Informasi</b>	Tidak terlalu detail	Lebih detail
<b>Target Pengguna</b>	Pengguna sistem yang tidak mempunyai pengetahuan teknik yang detail	Developer (terkadang customer ingin mengetahui)
<b>Bentuk Informasi</b>	Bahasa natural dan diagram sederhana tentang layanan sistem	Model sistem

# Contoh User dan System Requirement

- **User Requirement**

Sistem bisa melakukan operasi dasar pengolahan data buku yang ada di perpustakaan

- **System Requirement**

1. Sistem bisa melayani proses penambahan data buku yang diinput oleh pengguna
2. Sistem bisa melayani pengeditan data buku yang sudah tersimpan dalam basis data
3. Sistem bisa melayani penghapusan data buku yang tidak sedang dipinjam atau dikembalikan

# User Requirements

- ⦿ Harus menggambarkan kebutuhan fungsional dan non-fungsional sedemikian rupa sehingga dimengerti oleh pengguna system yang tidak memiliki pengetahuan teknis rinci.
- ⦿ User requirements didefinisikan menggunakan Bahasa natural, tabel dan diagram sehingga dapat dimengerti oleh semua pengguna

# Problems With Natural Language

- ⦿ Lack of clarity (**Kurang kejelasan**)
  - Presisi sulit tanpa membuat dokumen yang sulit dibaca.
- ⦿ Requirements confusion (**Kebingungan menentukan kebutuhan**)
  - Kebutuhan fungsional dan non-fungsional cenderung campur aduk.
- ⦿ Requirements amalgamation (**Penggabungan kebutuhan**)
  - Beberapa kebutuhan yang berbeda dinyatakan bersama-sama.

# System Requirements

- ⦿ Spesifikasi lebih detail dari fungsi system, layanan dan kendala dari user requirements.
- ⦿ Dimaksudkan untuk menjadi dasar untuk merancang sistem.
- ⦿ Dapat dimasukkan ke dalam sistem kontrak.
- ⦿ Persyaratan sistem dapat didefinisikan atau digambarkan menggunakan model system.

# Functional and Non-Functional Requirements

- ① **Functional requirements**

Pernyataan mengenai layanan system yang harus disediakan, bagaimana system harus bereaksi terhadap input dan bagaimana system harus berperilaku dalam situasi tertentu.

- ② **Non-functional requirements**

Batasan layanan atau fungsi yang ditawarkan oleh sistem seperti kendala waktu, kendala pada proses pembangunan, standar, dll.

- ③ **Domain requirements**

Persyaratan yang berasal dari domain aplikasi dari sistem dan yang mencerminkan karakteristik domain tersebut.

# Functional requirements

- ⦿ Menggambarkan fungsionalitas atau layanan system.
- ⦿ Tergantung pada jenis perangkat lunak, apa yang pengguna harapkan dan jenis sistem di mana perangkat lunak digunakan.
- ⦿ Kebutuhan fungsional **pengguna** mungkin pernyataan tingkat tinggi dari apa yang harus dilakukan sistem tetapi persyaratan fungsional **sistem** harus menjelaskan layanan sistem secara rinci.

# Non-functional Requirements

## Examples

- ◎ **Product requirement**

8.1 Antarmuka untuk SIADIN harus diimplemetasikan sebagai HTML sederhana tanpa frame atau Java applets.

- ◎ **Organisational requirement**

9.3.2 Proses pembangunan system dan pengiriman dokumen harus mengacu pada proses dan pengiriman yang didefinisikan dalam XYZCo-SP-STAN-95.

- ◎ **External requirement**

7.6.5 Sistem tidak akan mengungkapkan informasi pribadi apapun tentang pelanggan selain dari nama dan nomor referensi mereka kepada operator sistem

# Requirements Measures

<b>Properti</b>	<b>Pengukuran</b>
Speed	Pemrosesan transaksi/ detik Waktu Respon Waktu refresh layar
Size	M Bytes Jumlah chip ROM
Ease of use	Waktu pelatihan Jumlah frame bantuan
Reliability	Rata-rata waktu kegagalan Kemungkinan tidak tersedia Tingkat terjadinya kegagalan Ketersediaan
Robustness	Waktu restart setelah gagal Prosentase penyebab kegagalan Kemungkinan data hilang setelah kegagalan
Portability	Prosentase laporan tergantung sasaran Jumlah target sistem

# Requirements Interaction

- ⦿ Konflik antara kebutuhan non-fungsional yang berbeda yang umum dalam sistem yang kompleks.
- ⦿ Spacecraft system (**Sistem pesawat ruang angkasa**)
  - Untuk meminimalkan berat, jumlah chip yang terpisah dalam sistem harus diminimalkan.
  - Untuk meminimalkan konsumsi daya, chip daya yang rendah harus digunakan.
  - Namun, dengan menggunakan chip daya rendah mungkin berarti bahwa lebih banyak chip yang telah digunakan. Mana merupakan kebutuhan yang paling penting?

# Domain Requirements

- ⦿ Berasal dari domain aplikasi dan menggambarkan karakteristik sistem dan fitur yang mencerminkan domain.
- ⦿ Persyaratan domain menjadi persyaratan fungsional baru, kendala pada kebutuhan yang ada atau mendefinisikan komputasi tertentu.
- ⦿ Apabila persyaratan domain tidak memuaskan, sistem mungkin tidak bisa dijalankan.

# Domain Requirements Problems

- ⦿ Understandability (**Dapat dimengerti**)
  - Persyaratan disajikan dalam bahasa domain aplikasi;
  - Hal ini sering tidak dipahami oleh para insinyur perangkat lunak mengembangkan sistem.
- ⦿ Implicitness (**Bersifat implisit**)
  - Domain specialist memahami area dengan baik sehingga tidak membuat persyaratan yang eksplisit.

# Requirements Completeness and Consistency

- ⦿ Pada prinsipnya, kebutuhan harus lengkap dan konsisten.
  - **Lengkap**
    - Termasuk deskripsi dari semua fasilitas yang disyaratkan.
  - **Konsisten**
    - Seharusnya tidak ada konflik dan kontradiksi dalam deskripsi dari fasilitas sistem.
- ⦿ Dalam prakteknya, adalah mustahil untuk menghasilkan dokumen lengkap dan persyaratan yang konsisten.

# Requirements Imprecision

- ⦿ Permasalahan akan muncul jika kebutuhan tidak ditetapkan.
- ⦿ Kebutuhan yang ambigu dapat ditafsirkan dalam berbagai cara oleh pengembang dan pengguna.
- ⦿ Mempertimbangkan istilah '*Appropriate Viewers*' (pemirsa yang tepat)
  - Interpretasi pengguna – Tujuan khusus untuk setiap dokumen yang berbeda;
  - Interpretasi pengembang – Menyediakan daftar isi yang menunjukkan isi dari dokumen

# Guidelines for Writing Requirements

- ⦿ Menciptakan sebuah format standar dan menggunakannya untuk semua kebutuhan.
- ⦿ Menggunakan bahasa dengan cara yang konsisten. Gunakan wajib untuk persyaratan yang wajib, **harus** untuk kebutuhan yang diinginkan.
- ⦿ Gunakan penyorotan teks untuk mengidentifikasi bagian penting dari kebutuhan.
- ⦿ Hindari penggunaan jargon computer.

# Problems with NL specification

## ⦿ **Ambiguity**

- Pembaca dan penulis kebutuhan harus menginterpretasikan kata yang sama dengan cara yang sama. Bahasa natural yang ambigu secara alami akan sangat sulit.

## ⦿ **Over-flexibility**

- Hal yang sama dapat dikatakan dalam sejumlah cara yang berbeda dalam spesifikasi.

## ⦿ **Lack of modularisation**

- Struktur bahasa natural tidak memadai untuk struktur persyaratan system.

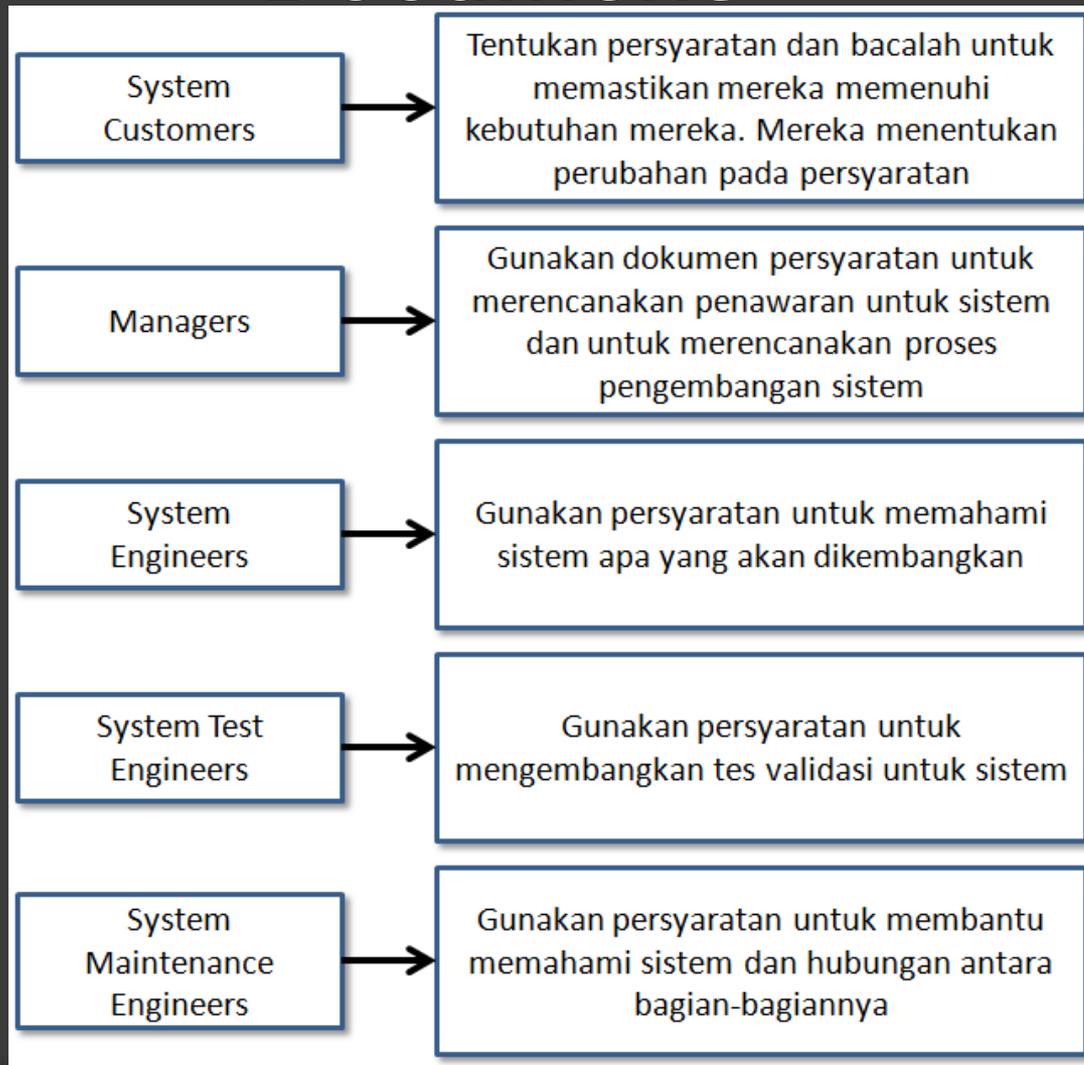
# Alternatives to NL specification

<b>Notasi</b>	<b>Deskripsi</b>
Structured natural language	Pendekatan ini tergantung pada mendefinisikan bentuk standar atau template untuk mengungkapkan spesifikasi persyaratan.
Design description languages	Pendekatan ini menggunakan bahasa seperti bahasa pemrograman tetapi dengan fitur yang lebih abstrak untuk menentukan persyaratan dengan mendefinisikan model operasional sistem. Pendekatan ini jarang banyak digunakan meskipun dapat berguna untuk spesifikasi antarmuka.
Graphical notations	Sebuah bahasa grafis, dilengkapi dengan penjelasan teks digunakan untuk mendefinisikan kebutuhan fungsional untuk sistem. Contoh awal dari bahasa grafis seperti itu SAD. Sekarang, deskripsi use case dan sequence diagram yang umum digunakan.
Mathematical specifications	Ini adalah notasi berdasarkan konsep-konsep matematika seperti mesin finite-state atau set. Spesifikasi ambigu mengurangi argumen antara pelanggan dan kontraktor tentang fungsi sistem. Namun, sebagian besar konsumen tidak mengerti spesifikasi formal dan enggan menerimanya sebagai sistem kontrak.

# The Requirements Document

- ⦿ Dokumen persyaratan adalah pernyataan resmi dari apa yang dibutuhkan dari para pengembang sistem.
- ⦿ Harus mencakup baik definisi dari kebutuhan pengguna dan spesifikasi persyaratan sistem.
- ⦿ Hal ini bukan dokumen desain. Se jauh mungkin, harus menge-set APA yang harus system lakukan daripada CARA melakukannya.

# Users of a Requirements Document



# IEEE Requirements Standard

- ⦿ Mendefinisikan struktur umum untuk dokumen yang harus dipakai untuk setiap system tertentu.
  - Pengantar.
  - Gambaran umum.
  - Persyaratan tertentu.
  - Lampiran.
  - Index.

# Requirements Document Structure

- Preface (**Kata pengantar**)
- Introduction (**Pendahuluan**)
- Glossary (**Daftar kata-kata**)
- User requirements definition (**Definisi kebutuhan pengguna**)
- System architecture (**Arsitektur sistem**)
- System requirements specification (**Spesifikasi kebutuhan sistem**)
- System models (**Model sistem**)
- System evolution (**Evolusi sistem**)
- Appendices (**Lampiran**)
- Index (**Indeks**)

# REQUIREMENT ENGINEERING TASKS

# Requirement Engineering Tasks

## ◎ Inception

- Software Engineer menggunakan pertanyaan bebas konteks untuk membangun pemahaman dasar tentang masalah, orang-orang yang menginginkan solusi, sifat dari solusi, dan efektivitas kolaborasi antara klien dan pengembang

## ◎ Elicitation

- Mencari tahu dari pelanggan apa yang menjadi tujuan produk, apa yang harus dilakukan, bagaimana produk cocok dengan kebutuhan bisnis, dan bagaimana produk digunakan

## ◎ Elaboration

- Berfokus pada pengembangan model teknis fungsi software, fitur, dan kendala menggunakan informasi yang diperoleh selama **inception** dan **elicitation**

# Requirement Engineering Tasks (2)

## ⦿ **Negotiation**

- Persyaratan dikategorikan dan disusun dalam himpunan bagian, hubungan antara persyaratan diidentifikasi, persyaratan dibahas untuk verifikasi kebenarannya, kemudian persyaratan diprioritaskan berdasarkan kebutuhan pelanggan

## ⦿ **Specification**

- Menggambarkan fungsi, kinerja, dan kendala pengembangan untuk sistem berbasis komputer

## ⦿ **Requirements validation**

- Tinjauan teknis formal yang digunakan untuk menguji spesifikasi produk untuk memastikan kualitas persyaratan dan bekerja sesuai dengan standar yang telah disepakati untuk proses, proyek, dan produk

# Initiating Requirements Engineering Process

- ⦿ Identifikasi stakeholders (pemangku kepentingan)
- ⦿ Mengakui keberadaan beberapa sudut pandang stakeholder
- ⦿ Bekerja kolaborasi antara stakeholder
- ⦿ Pertanyaan bebas konteks ini fokus pada pelanggan, stakeholder, tujuan keseluruhan, dan manfaat dari sistem
  - Siapa yang meminta untuk bekerja?
  - Siapa yang akan menggunakan solusi?
  - Apa yang akan menjadi keuntungan ekonomi dari solusi yang sukses?
  - Apakah ada sumber lain untuk solusi yang dibutuhkan?

# Initiating Requirements Engineering Process

- ◎ Set pertanyaan berikutnya memungkinkan pengembang untuk lebih memahami masalah dan persepsi pelanggan berdasarkan solusi
  - Bagaimana ciri output yang bagus untuk solusi yang sukses?
  - Apa masalah dari solusi ini?
  - Dapatkah Anda menggambarkan lingkungan bisnis dimana solusi digunakan?
  - Adakah kendala yang mempengaruhi dalam pendekatan solusi?
- ◎ Set pertanyaan terakhir berfokus pada efektivitas komunikasi
  - Apakah Anda orang terbaik untuk memberikan jawaban “resmi” atas pertanyaan ini?
  - Apakah pertanyaan saya relevan dengan masalah Anda?
  - Apakah saya terlalu banyak bertanya?
  - Dapatkah orang lain memberikan informasi tambahan?
  - Haruskah saya meminta Anda menjawab apapun?

# Eliciting Requirements

- ◎ Pengumpulan persyaratan secara kolaboratif
  - Rapat dihadiri oleh pengembang dan pelanggan
  - Aturan untuk persiapan dan partisipasi ditetapkan
  - Agenda yang fleksibel digunakan
  - Fasilitator mengatur pertemuanMekanisme (misalnya, stickers, flip sheets, electronic bulletin board) digunakan untuk mengukur konsensus kelompok
  - Tujuannya adalah untuk mengidentifikasi masalah, mengusulkan elemen solusi, negosiasi pendekatan, dan menentukan set awal persyaratan berdasarkan solusi

# Eliciting Requirements (2)

- ◎ Quality function deployment (QFD)
  - Identifikasi tiga jenis kebutuhan (normal, expected, exciting)
  - Dalam pertemuan klien **function deployment** digunakan untuk menentukan nilai masing-masing fungsi yang diperlukan untuk sistem
  - **Information deployment** mengidentifikasi kedua objek data dan peristiwa yang dihasilkan sistem (terkait dengan fungsi)
  - **Task deployment** meneliti perilaku sistem dalam lingkungannya
  - **Value analysis** dilakukan untuk menentukan prioritas relative dari kebutuhan masing-masing yang dihasilkan oleh kegiatan deployment
- ◎ User-scenarios
  - Juga dikenal sebagai use-case, menggambarkan bagaimana sistem akan digunakan
  - Pengembang dan pengguna membuat satu set rangkaian penggunaan untuk sistem yang akan dibangun

# Elicitation Work Products

- Pernyataan tentang kebutuhan dan kelayakan
- Pernyataan dibatasi untuk sistem atau produk
- Daftar pemangku kepentingan yang terlibat dalam elisitasi persyaratan
- Deskripsi lingkungan teknis sistem
- Daftar persyaratan yang diatur oleh fungsi dan kendala domain yang berlaku
- Set skenario (use-cases) yang menyediakan wawasan pengoperasian sistem yang dikerahkan
- Prototipe yang dikembangkan untuk lebih memahami kebutuhan

# Requirements Elaboration

- ① Developing Use-Cases
  - Setiap use-case bercerita tentang bagaimana pengguna akhir berinteraksi dengan sistem dalam keadaan tertentu
- ① Analysis Model
  - Mempunyai maksud untuk memberikan deskripsi dari informasi yang diperlukan, fungsional dan domain perilaku untuk sistem berbasis komputer
  - Analysis Model Elements
    - Scenario-based elements (menggambarkan sistem dari perspektif pengguna)
    - Class-based elements (hubungan antara objek-objek yang dimanipulasi beserta actor dan atributnya)
    - Behavioral elements (menggambarkan sistem dan perilaku kelas sebagai state dan transisi antar state)
    - Flow-oriented elements (Menunjukkan bagaimana informasi mengalir melalui sistem dan ditransformasikan oleh fungsi sistem)

# Negotiating Requirements

- ⦿ Negotiation activities
  - Identifikasi stakeholder kunci
  - Menentukan stakeholders' "win conditions"
  - Melakukan negosiasi untuk mendamaikan stakeholder "win conditions" menjadi "win-win" untuk semua stakeholders (termasuk developers)
- ⦿ Key points
  - Ini bukan kompetisi
  - Memetakan strategi
  - Mendengarkan secara aktif
  - Fokus pada kepentingan pihak lain
  - Jangan egois
  - Jadilah kreatif
  - Bersiaplah untuk komitmen

# Requirements Validation

- ⦿ Memperhatikan bahwa persyaratan menentukan system yang benar-benar diinginkan pelanggan
- ⦿ Biaya kesalahan yang tinggi menyebabkan validasi persyaratan sangat penting
  - Memperbaiki kesalahan persyaratan mungkin membutuhkan biaya hingga 100 kali biaya memperbaiki kesalahan implementasi

# Requirements Checking

- ⦿ **Validity**. Apakah sistem menyediakan fungsi terbaik yang mendukung kebutuhan pelanggan?
- ⦿ **Consistency**. Apakah ada konflik persyaratan?
- ⦿ **Completeness**. Apakah semua fungsi yang dibutuhkan oleh pelanggan sudah ada?
- ⦿ **Realism**. Dapatkah persyaratan diimplementasikan sesuai anggaran dan teknologi yang tersedia
- ⦿ **Verifiability**. Dapatkah persyaratan diperiksa?

# Requirements Validation Techniques

- ⦿ Requirements reviews
  - Analisis manual persyaratan secara sistematis.
- ⦿ Prototyping
  - Menggunakan model eksekusi dari sistem untuk memeriksa persyaratan.
- ⦿ Test-case generation
  - Mengembangkan tes persyaratan untuk memeriksa testability.

# Requirement Review

- ⦿ Apakah setiap kebutuhan konsisten dengan proyek secara keseluruhan atau tujuan sistem?
- ⦿ Apakah semua persyaratan yang ditentukan sesuai pada tingkat abstraksi?
- ⦿ Apakah setiap kebutuhan penting untuk tujuan sistem atau merupakan suatu add-on fitur?
- ⦿ Apakah setiap kebutuhan dibatasi dan tidak ambigu?
- ⦿ Apakah Anda tahu sumber untuk setiap kebutuhan?
- ⦿ Apakah persyaratan bertentangan satu sama lain?
- ⦿ Apakah persyaratan dicapai dalam lingkungan teknis yang diusulkan untuk sistem atau produk?
- ⦿ Apakah setiap kebutuhan dapat diuji?
- ⦿ Apakah model kebutuhan mencerminkan informasi, fungsi, dan perilaku sistem yang akan dibangun?
- ⦿ Apakah model persyaratan telah dipartisi dengan cara mengekspos informasi sistem yang lebih rinci secara progresif?
- ⦿ Apakah semua pola persyaratan telah benar divalidasi dan mereka konsisten dengan kebutuhan pelanggan?

# Requirements Management

- ⦿ Membantu tim proyek untuk mengidentifikasi, mengendalikan, dan melacak persyaratan serta perubahan sebagai hasil proyek
- ⦿ Banyak dari kegiatan ini adalah identik dengan proses manajemen konfigurasi perangkat lunak (Software Configuration Management)

# Requirements Management

## ◎ Process:

- Persyaratan diidentifikasi,
- Tandai dengan identifier unik dan
- diklasifikasikan berdasarkan jenis (fungsional, data, perilaku, antarmuka, atau output)

## ◎ Tools:

- Traceability tables
  - Dikembangkan dan diperbarui setiap persyaratan dimodifikasi
- Database systems
  - Berharga dalam membantu tim software melacak perubahan persyaratan

# Requirements Change

- ⦿ Prioritas kebutuhan dari sudut pandang yang berbeda berubah selama proses pembangunan.
- ⦿ Pelanggan sistem dapat menentukan persyaratan dari perspektif bisnis yang bertentangan dengan kebutuhan pengguna akhir.
- ⦿ Bisnis dan lingkungan teknis dari perubahan sistem selama perkembangannya.

# Requirements Classification

<b>Tipe Requirements</b>	<b>Deskripsi</b>
Mutable requirements	Persyaratan yang berubah karena perubahan lingkungan di mana organisasi beroperasi. Misalnya, dalam sistem rumah sakit, dana perawatan pasien dapat berubah sehingga membutuhkan informasi berbeda yang dikumpulkan.
Emergent requirements	Persyaratan yang muncul sebagai pemahaman pelanggan dari sistem berkembang selama pengembangan sistem. Proses desain dapat mengungkapkan kebutuhan baru yang muncul.
Consequential requirements	Persyaratan yang dihasilkan dari pengenalan sistem komputer. Memperkenalkan sistem komputer dapat mengubah proses organisasi dan membuka cara-cara kerja baru yang menghasilkan persyaratan sistem baru.
Compatibility requirements	Persyaratan yang bergantung pada sistem atau proses bisnis tertentu dalam sebuah organisasi. Seperti perubahan ini, kompatibilitas persyaratan pada sistem ditugaskan atau disampaikan juga mungkin harus berevolusi.

# CASE Tool Support

- ⦿ Requirements storage
  - Persyaratan harus dikelola secara aman
- ⦿ Change management
  - Proses manajemen perubahan adalah alur kerja proses yang dapat didefinisikan dan informasi yang mengalir antara tahap-tahap ini sebagian otomatis.
- ⦿ Traceability management
  - Pengambilan otomatis hubungan antara persyaratan.

# Requirements Change Management

- ⦿ Harus berlaku untuk semua perubahan yang diusulkan untuk persyaratan.
- ⦿ Tahap pokok
  - **Problem analysis**. Mendiskusikan masalah persyaratan dan mengusulkan perubahan;
  - **Change analysis and costing**. Menilai dampak perubahan pada persyaratan lainnya;
  - **Change implementation**. Memodifikasi dokumen persyaratan dan dokumen lainnya untuk mencerminkan perubahan.

Terima Kasih