

Software Requirements

Rekayasa & Analisa Kebutuhan Perangkat Lunak



Husni

husni@trunojoyo.ac.id

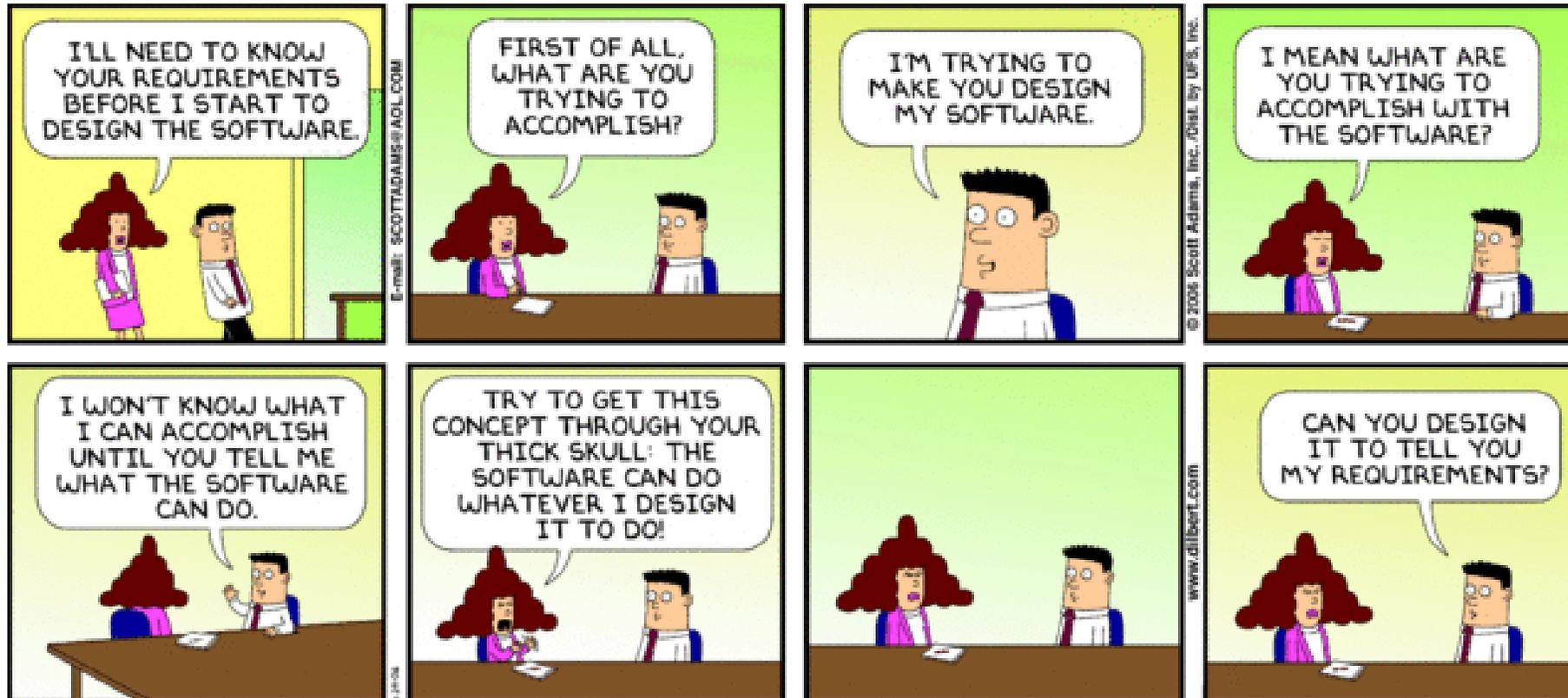
Garis Besar Bahasan

- Pendahuluan
- Kebutuhan Perangkat Lunak
- Rancangan Perangkat Lunak
- UML
- Dokumen SRS

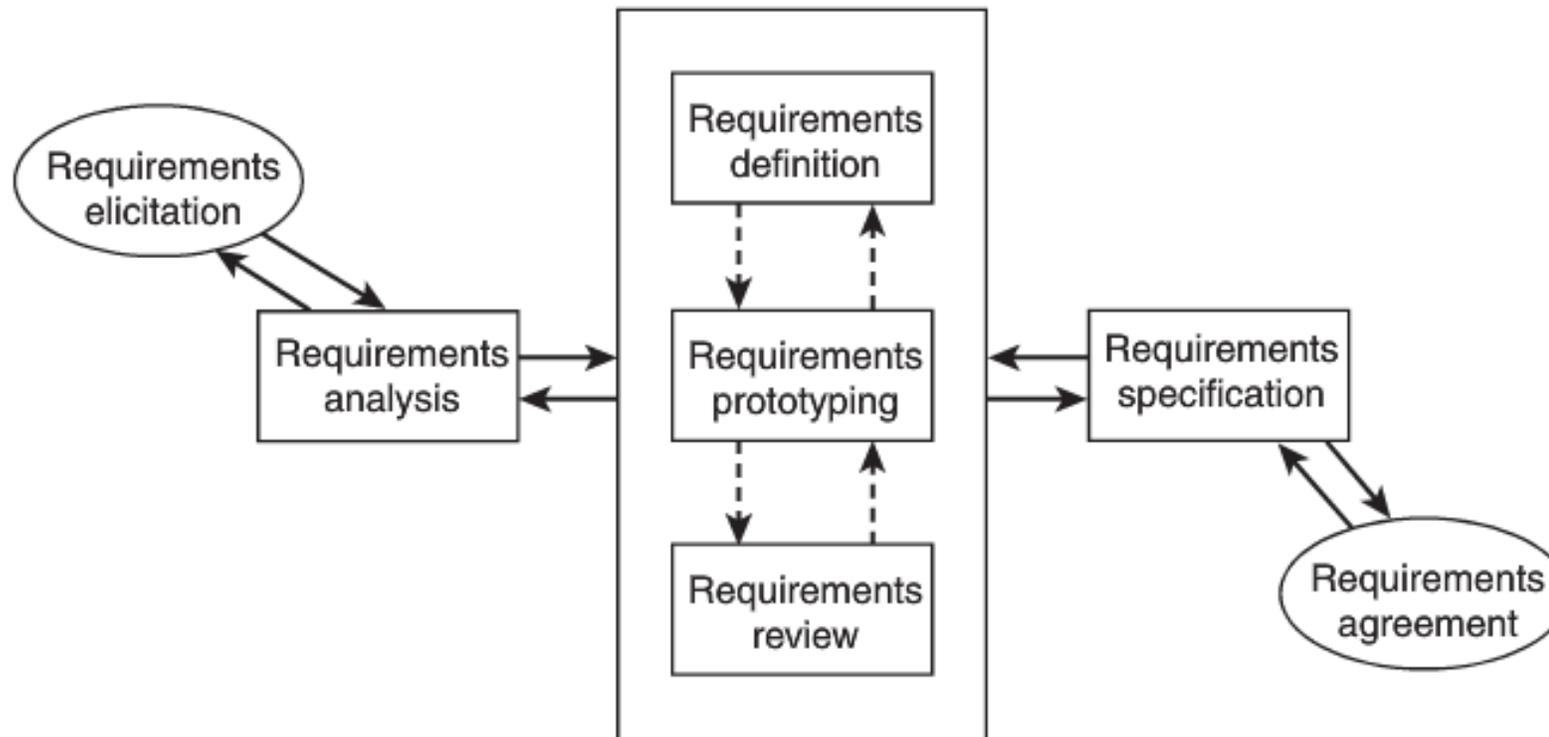
Kebutuhan Perangkat Lunak

Software Requirements

Requirements (Kebutuhan, Persyaratan) bagi sistem software mengatur apa yang sistem akan lakukan dan mendefinisikan batasan-batasan pada operasi dan implementasinya.



- **Requirements:** “Pernyataan yang mendeskripsikan apa yang akan sistem software lakukan tetapi bukan bagaimana itu dikonstruksikan.”
- **Requirements Engineering:** “Serangkaian kegiatan yang berkaitan dengan pengembangan dan persetujuan himpunan akhir dari spesifikasi kebutuhan”
- Langkah-langkah dalam proses *Requirements Engineering*:



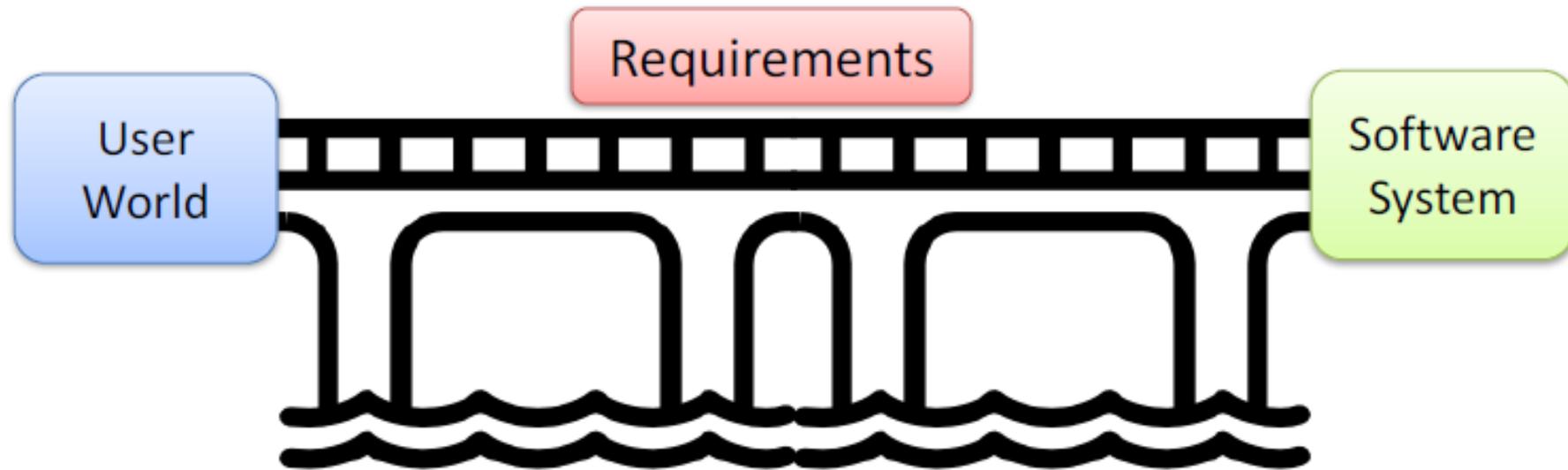
Requirements Engineering

Rekayasa Kebutuhan



Apa itu Rekayasa Kebutuhan?

Requirements adalah jembatan antara **dunia nyata** dan **sistem software**



You  Pengantar tentang *Requirements Engineering*: <https://youtu.be/Ec0s0z5uXQ8>

Rekayasa Kebutuhan

Yang diperoleh Customer



Yang sebenarnya dibutuhkan Customer



Kebutuhan & Rancangan Software

Requirements (WHAT):

- **WHAT** (Apa yang sistem akan lakukan)
- Mendeskripsikan apa yang sistem akan lakukan dengan Kata dan Gambar, dll.
- **SRS** – Dokumen Software Requirements Specification

Software Design (HOW):

- **HOW** (Bagaimana itu akan dilakukan)
- Contoh: Rancangan GUI, UML, diagram ER, CAD, dll.
- **SDD** – Software Design Document

Catat! Banyak praktisi tidak memisahkan dokumen SRS dan SDD, tetapi menggabungkan ke dalam Requirements & Design Document (**SRD**).

Dalam praktek, *requirements* dan *design* tidak dapat dipisahkan.

Customer Requirements

Kebutuhan Pengguna

Kebutuhan Sistem



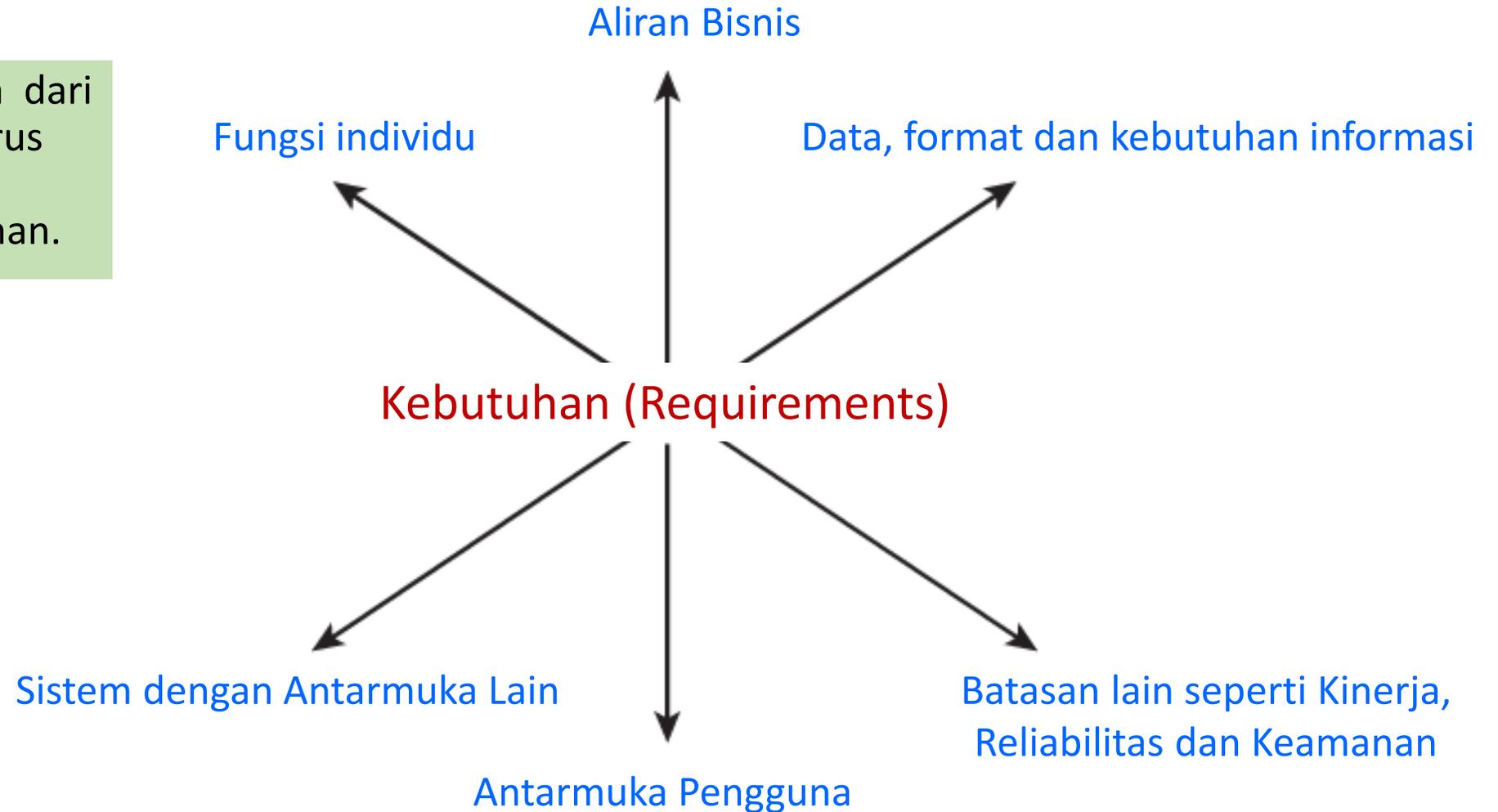
Komunikasi antara Customers/Stakeholders dan orang-orang Software/IT adalah tuntutan! Apa yang sesungguhnya Customer inginkan/butuhkan?

Kebutuhan Fungsi

Kebutuhan Non-Fungsi

6 Dimensi Kebutuhan

Ada 6 kategori utama dari informasi yang harus diamati dalam penentuan kebutuhan.



6 Dimensi Kebutuhan

Fungsi Individu

- There is a need to create a new system for the cinema to attract more customers. By creating new system, this system must allow the customers to book the tickets, reserve the seats, and pay for the prices. This system must also provide a place for the cinema owner to promote the advertisements.
- “Individual functionality” is the the natural starting point. Ask the Users and Customers what their problems are in terms of what functions need to be implemented in the product.

6 Dimensi Kebutuhan

Aliran Bisnis

These are few steps for the users to book the tickets, reserve the seats, and pay the prices.

- Open the application provided
- Choose the movie you would like to watch
- Choose the most convenient time for you to watch the movie
- Reserve the seat
- Get the reserved seat code
- Pay the price

6 Dimensi Kebutuhan

Data, Format dan Kebutuhan Informasi

- The input data and the information needs are the databases of movie including the title, synopsis, movie time, subtitle, etc.

Kinerja, Reliabilitas dan Keamanan

6 Dimensi Kebutuhan

Antarmuka Pengguna

The new system that the IT company will create is a user-friendly application that can work on every operating system. The features are the following

- Allows the users to book the tickets for the movie the users would like to watch
- Allows the users to reserve the seats
- Allows the users to see the movie show times, the available time of that movie on the cinema, and watch the trailer of them
- Allows the users to pay for the prices of their booked tickets
- Provides a place where the user can place their advertisements and promote them

Sistem dengan Antarmuka lainnya

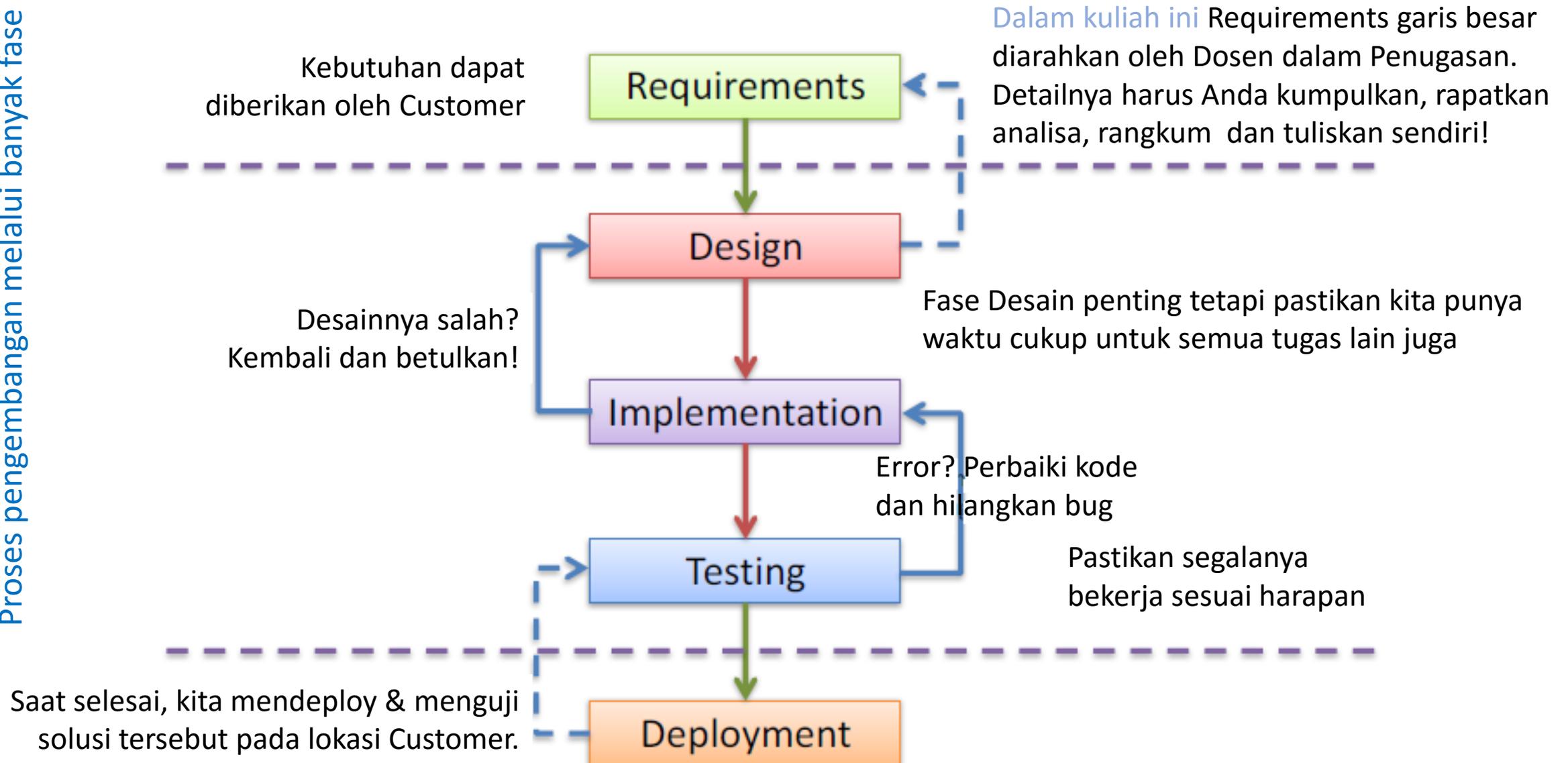
Siapa Akan Membacanya?

Pengguna dari dokumen kebutuhan Software

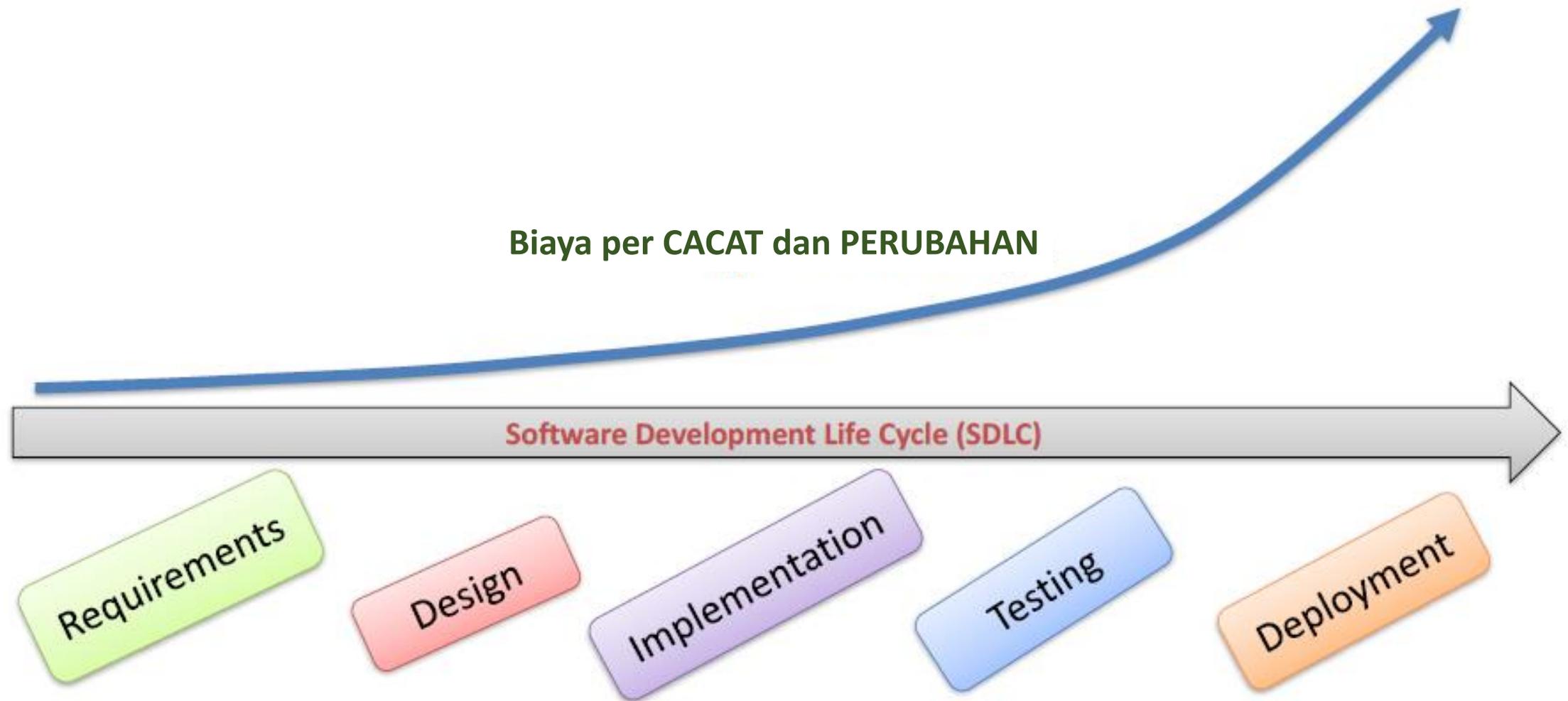


Proses Pengembangan

Proses pengembangan melalui banyak fase



Mengapa Banyak Waktu di Analisa Kebutuhan?



Software Requirements

Kebutuhan Perangkat Lunak

Sebaiknya diawali oleh
Customer

Kebutuhan
Tingkat Tinggi

...

....



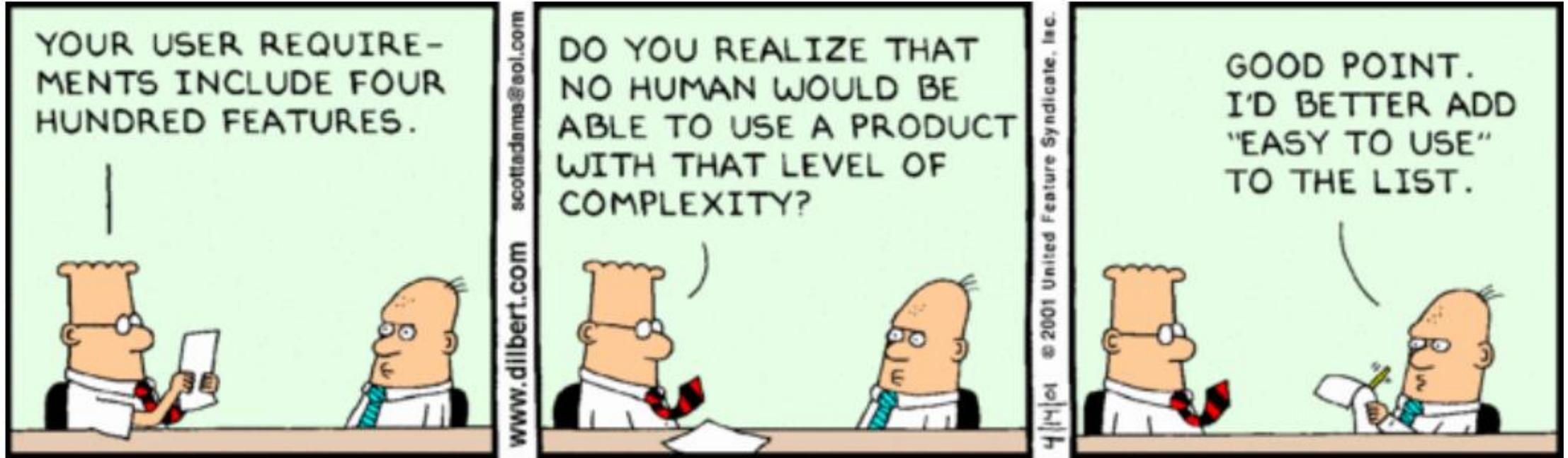
Harus ditulis oleh Perancang
Software (*Architect*)

Kebutuhan
Terperinci

...

...

Kebutuhan Level Tinggi vs. Terperinci



Persyaratan Level Tinggi

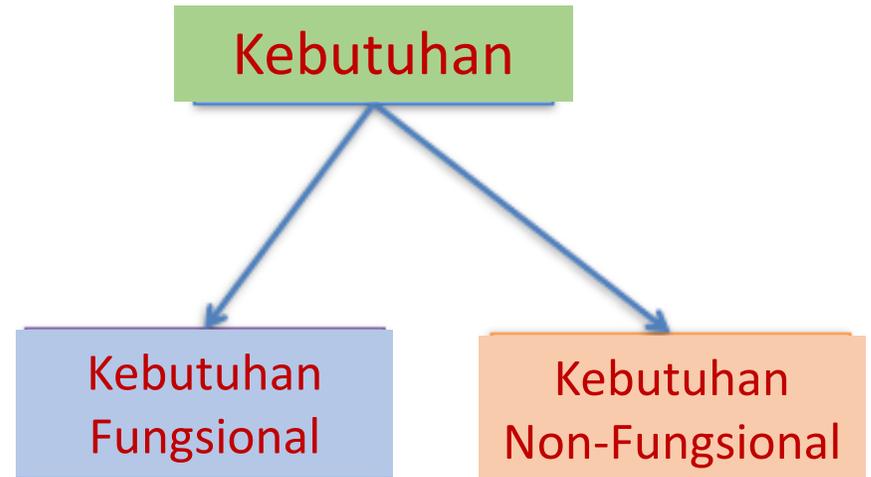
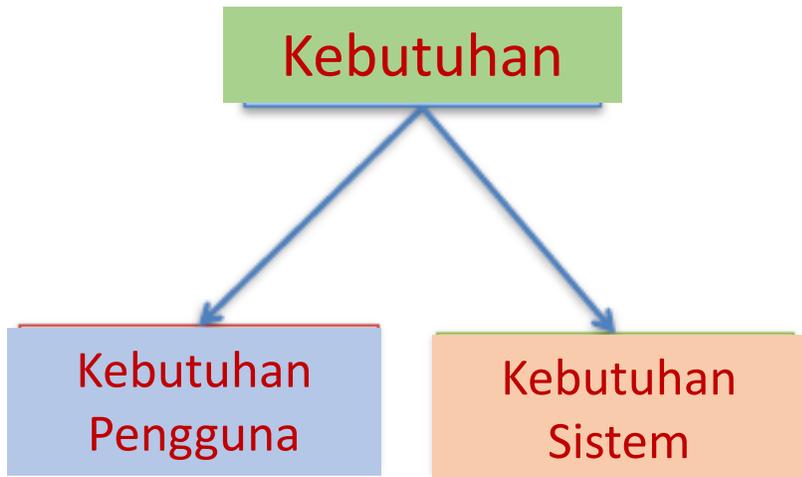
- What Apa yang sistem akan kerjakan?
- Who Siapa akan menggunakan sistem?
- What Apa tujuan dari sistem?
- Seperti Apa kinerja diharapkan?
- Komponen apa saja yang terdapat di dalam sistem
- Platform apa yang akan digunakan (PC, Tablet, Web?, ...)
- dll.

Gunakan Kata dan Gambar dalam rangka mendeskripsikan Kebutuhan ini!

Kebutuhan & Rancangan Terperinci

- Platform apa yang akan digunakan (Windows, iOS, ...) secara detail
- Perangkat (*tool*) dan Bahasa
- Arsitektur Software ()
- Frameworks (.NET, ASP.NET, ...)
- Sketsa rancangan GUI terperinci
- Diagram UML
- Diagram ER (Database)
- Gambar CAD
- dll.

Kategori Persyaratan Software



Kategori Persyaratan Software

Requirements: Pernyataan dalam bahasa alami plus diagram dari layanan yang sistem sediakan dan batasan-batasan operasionalnya. Ditulis untuk customers.

User Requirements

Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.

A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.

System Requirements

Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. May state what the system should not do.

Functional Requirements

Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc. Often apply to the system as a whole rather than individual features or services.

Non-Functional Requirements

Persyaratan Pengguna vs. Sistem

Persyaratan Pengguna (*User Requirements*):

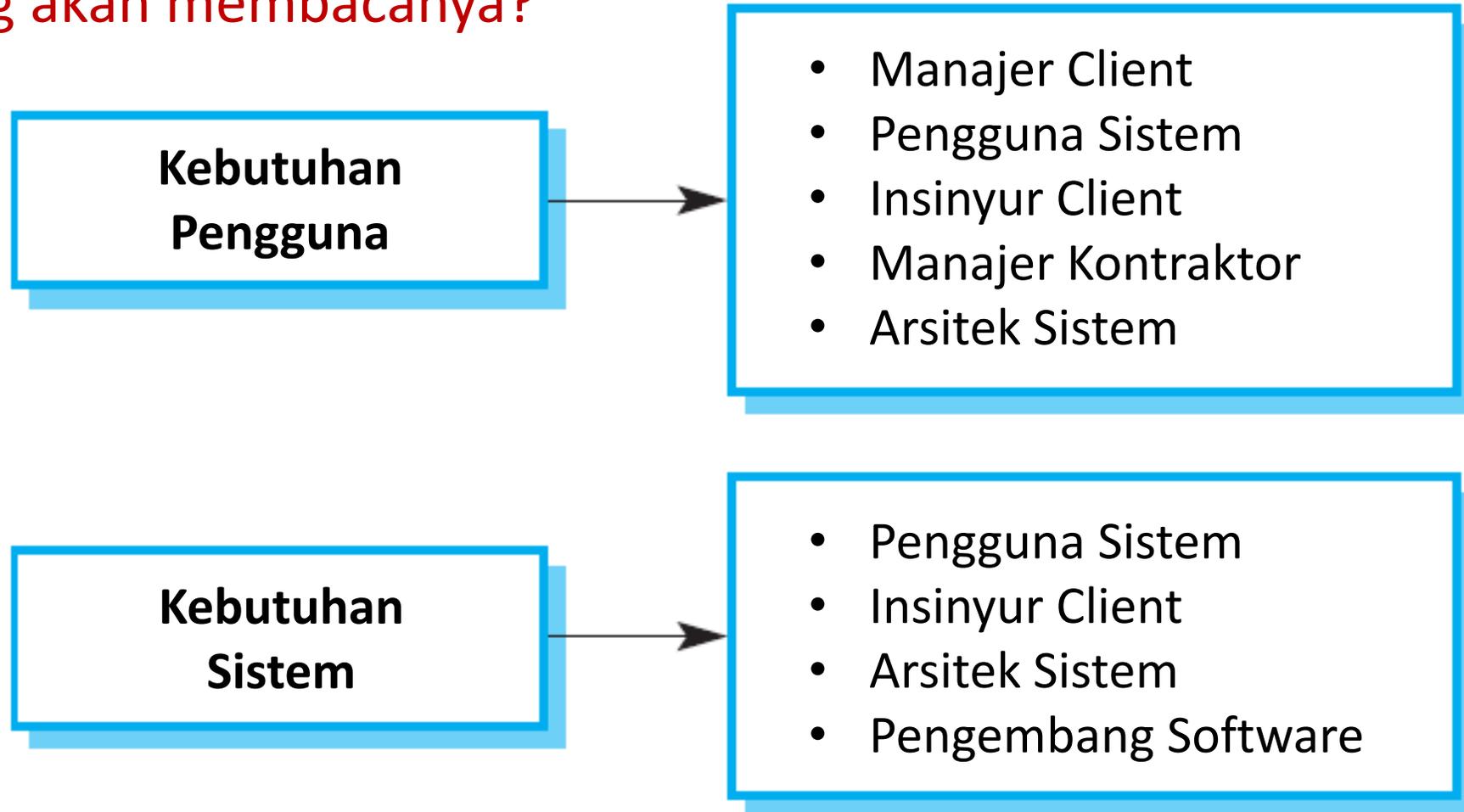
- Pernyataan dalam bahasa alami plus diagram dari layanan yang disediakan oleh sistem dan batasan-batasan operasionalnya. Ditulis untuk customers.

Persyaratan Sistem (*System Requirements*):

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

Persyaratan Pengguna vs. Sistem

Siapa yang akan membacanya?



Kebutuhan Pengguna vs. Sistem

Contoh:

Definisi Kebutuhan Pengguna

1. The MHC-PMS shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

Spesifikasi Kebutuhan Sistem

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.

1.2 The system shall automatically generate the report for printing after 17.30 on the last working day of the month.

1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.

1.4 If drugs are available in different dose units (e.g. 10mg, 20 mg, etc.) separate reports shall be created for each dose unit.

1.5 Access to all cost reports shall be restricted to authorized users listed on a management access control list.

Kebutuhan Functional vs. Non-Functional

Kebutuhan Fungsional

- Pernyataan mengenai layanan-layanan yang akan disediakan sistem, bagaimana sistem akan bereaksi terhadap input tertentu dan bagaimana sistem akan berperilaku dalam situasi tertentu.
- Mungkin menyatakan apa yang tidak akan dilakukan oleh sistem.

Kebutuhan Non-Fungsional

- Kendala pada layanan atau fungsi yang ditawarkan oleh sistem seperti kendala waktu, kendala pada proses pembangunan, standar, dll
- Biasanya berlaku pada sistem secara keseluruhan, bukan per layanan atau individu.

Kebutuhan Fungsional

- Mendeskripsikan fungsi atau layanan sistem.
- Tergantung pada jenis software, pengguna yang diharapkan dan jenis sistem dimana software digunakan.
- Kebutuhan pengguna fungsional mungkin pernyataan tingkat-tinggi dari apa yang sistem harus melakukan.
- Persyaratan sistem fungsional harus menjelaskan layanan sistem secara rinci.

Persyaratan Non-Fungsional

- Ini mendefinisikan properti & kendala sistem misalnya kehandalan, waktu respon dan persyaratan penyimpanan. Kendala adalah kemampuan perangkat I/O, representasi sistem, dll.
- Kebutuhan proses juga menetapkan penggunaan IDE, bahasa pemrograman atau metode pengembangan tertentu.
- Kebutuhan non-fungsional mungkin lebih penting dari kebutuhan fungsional. Jika ini tidak terpenuhi, sistem mungkin berguna.

Persyaratan Non-Fungsional

Contoh:

Persyaratan Produk:

Sistem ini harus tersedia untuk semua klinik selama jam kerja normal (Senin-Jumat, 0.830-17,30). Downtime dalam jam kerja normal tidak melebihi lima detik dalam satu hari.

Kebutuhan Organisasi:

Pengguna sistem akan mengotentikasi sendiri menggunakan otoritas kartu identitas kesehatan mereka.

Persyaratan Eksternal:

Sistem ini harus menerapkan ketentuan privasi pasien sebagaimana ditetapkan dalam HStan-03-2006-priv.

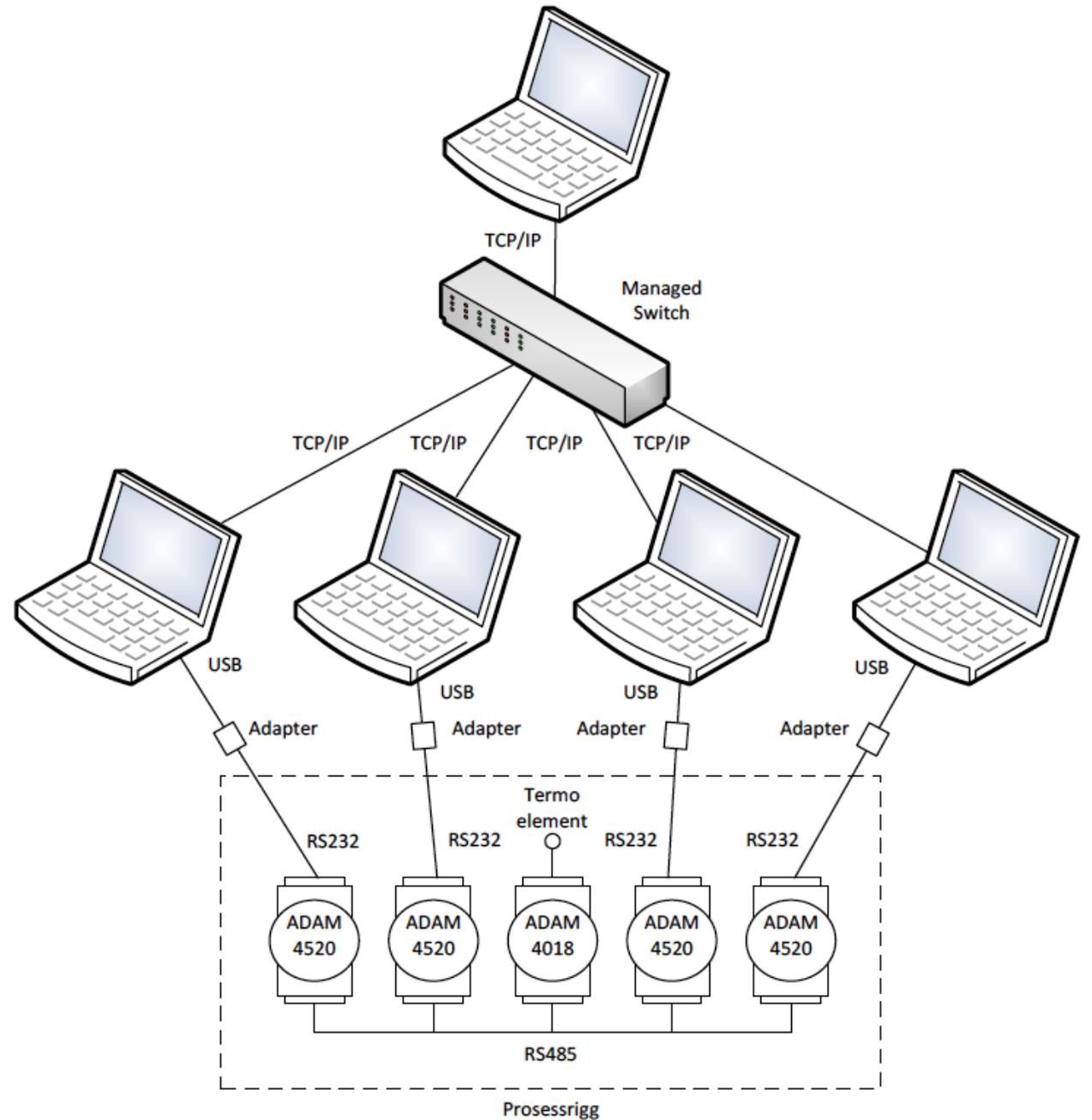
Software Design

(Rancangan Perangkat Lunak)

Ikhtisar & Arsitektur Sistem

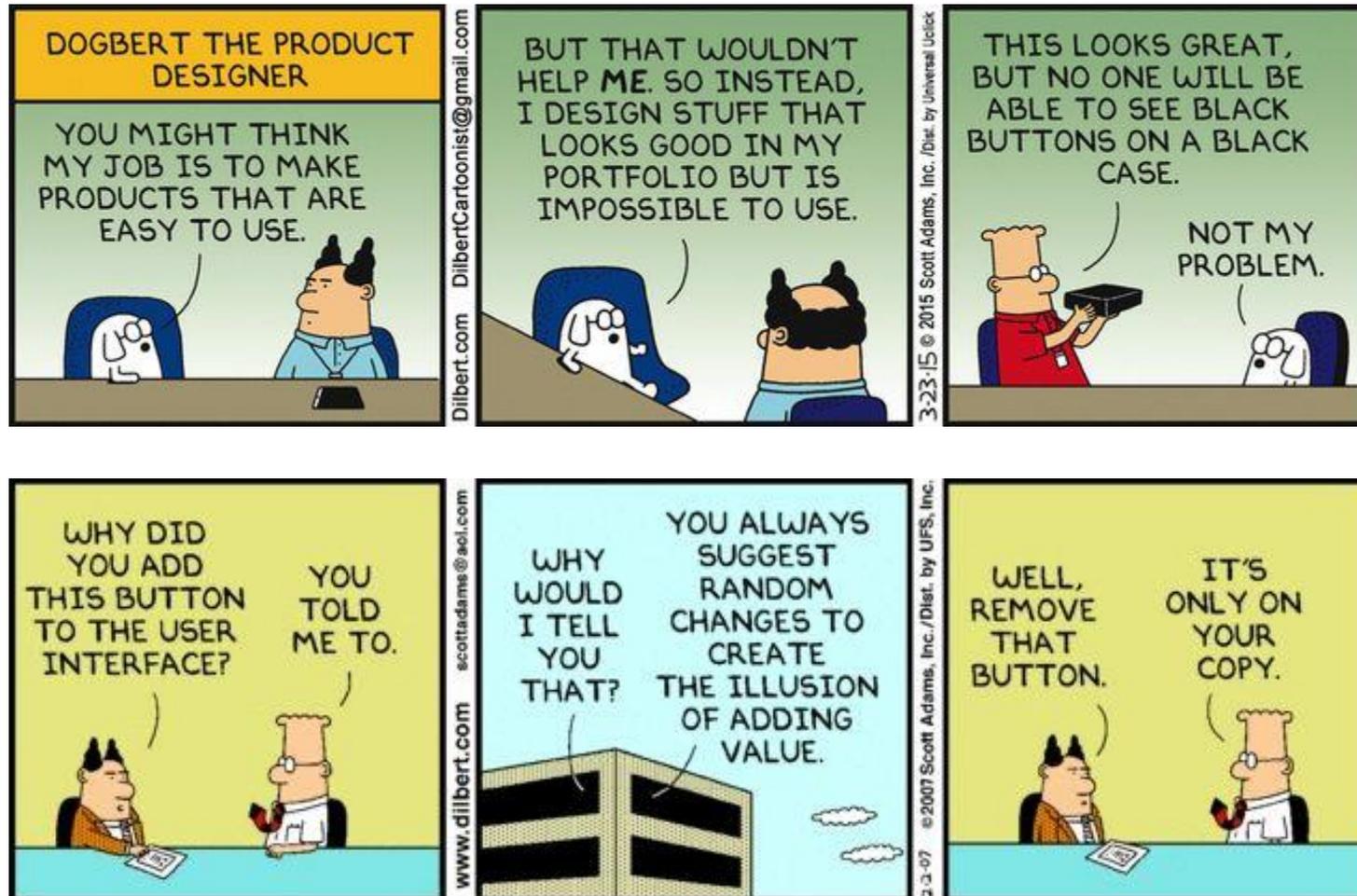
- Gambaran besar
- Memberikan pengantar mengenai sistem dan komponen-komponen utama serta bagaimana semua itu terhubung, dll
- PowerPoint atau Visio cukup bagus untuk keperluan ini.

Ikhtisar & Arsitektur Sistem: Contoh

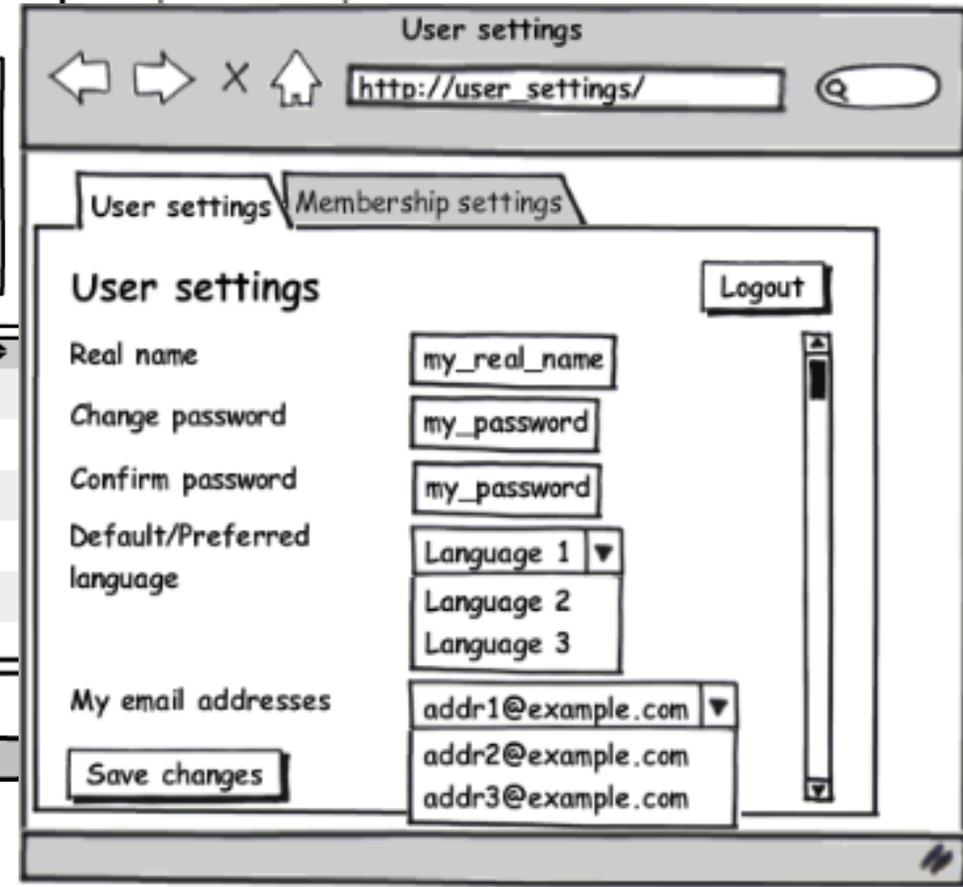
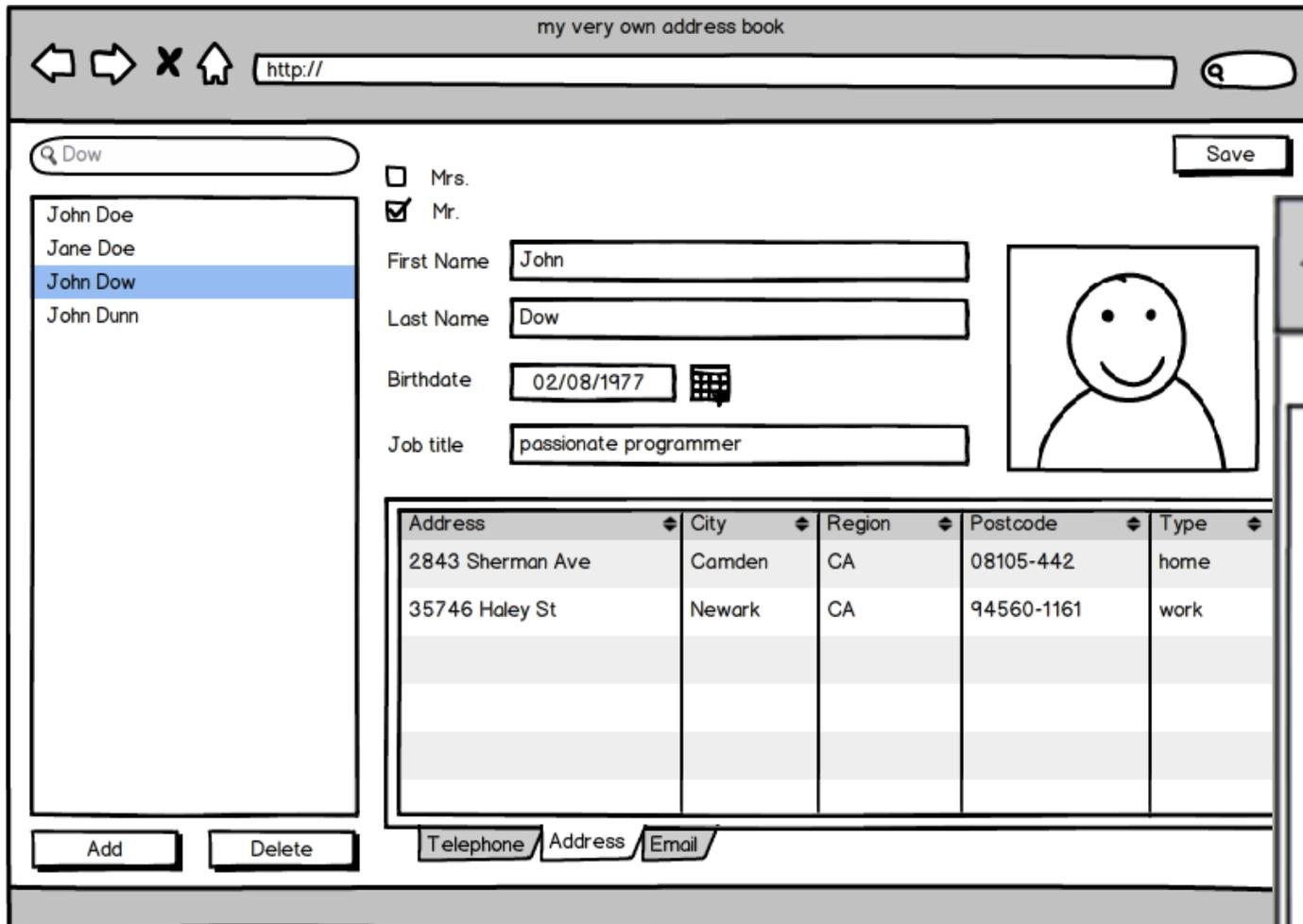


Rancangan = Desain

Rancangan GUI dan Rancangan Teknis (UML, ER diagram, CAD)

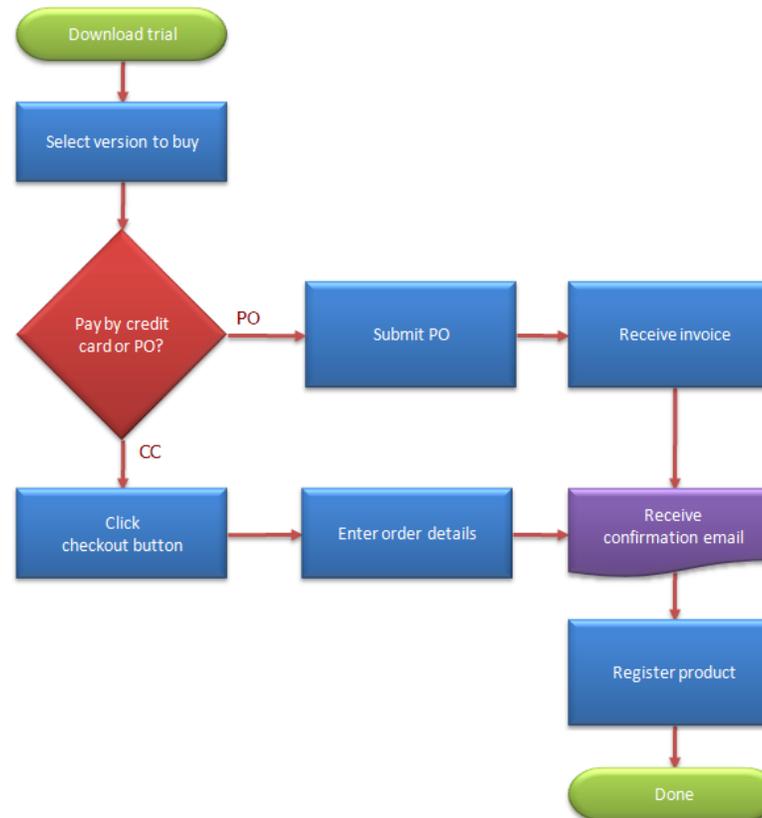
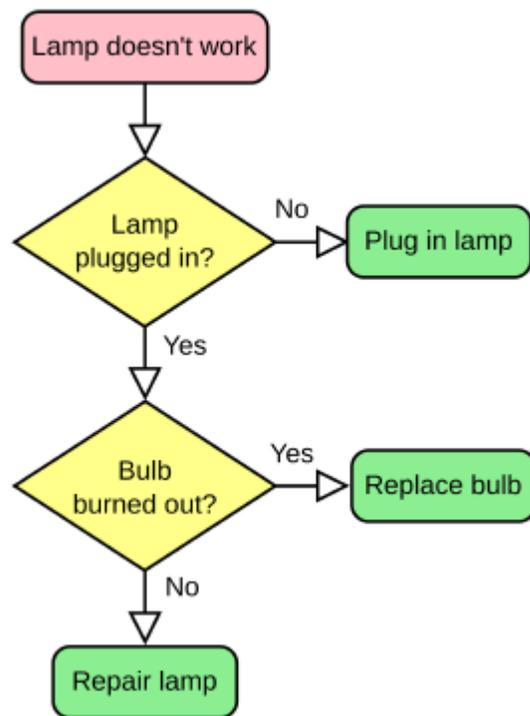


Sketsa Desain GUI "Mockups"



Flow Charts

High-Level Flow Charts memudahkan kita melihat bagaimana sistem akan bekerja



Flow Charts harus dipahami oleh non-programmer seperti Stakeholder, Project Manager dan Customer

Catatan! Nanti kita mungkin membuat diagram lebih rinci seperti diagram UML, dll.

Visio atau PowerPoint punya fitur bawaan untuk pembuat Flow Charts

Simbol Flow Chart

	Annotation		Decision		Direct access storage		Off-page reference - arrow		Tagged document
	Card		Delay		Display		Off-page reference - incoming		Tagged process
	Check / audit (diamond, square)		Disk storage		Document		Off-page reference - outgoing		Page
	Collate		Divided process (table view)		Entity (rounded)		On-page reference/ Inspection point		Paper tape
	Comment / note symbol (brace)		Event		Flow line		Or		Prepare conditional
	Communication link		Extract		Junction symbol		Predefined process		Stored data
	Compare (diamond, square)		Internal storage		Multi process		Process step		Terminal point
	Completed form		Lined document		Loop limit		Reference point		Start (circle)
	Concurrency symbol (fork / join)		Lined / shaded process		Manual file		Sequential access storage		Start (ellipse)
	Conditional selector		Database		Manual input		Merge		Stickman pictogram
	Control transfer		Decision Indicator (Yes, No)		Manual operation		Sort		
	Data input / output				Multi document		Summary		

Symbol Flow Chart



Terminator

Indicates the beginning or end of a program flow in your diagram.



Process

Indicates any processing function.



Decision

Indicates a decision point between two or more paths in a flowchart.



Delay

Indicates a delay in the process.



Data

Can represent any type of data in a flowchart.



Document

Indicates data that can be read by people, such as printed output.



Multiple documents

Indicates multiple documents.



Subroutine

Indicates a predefined (named) process, such as a subroutine or a module.



Preparation

Indicates a modification to a process, such as setting a switch or initializing a routine.



Display

Indicates data that is displayed for people to read, such as data on a monitor or projector screen.



Manual input

Indicates any operation that is performed manually (by a person).



Manual loop

Indicates a sequence of commands that will continue to repeat until stopped manually.



Loop limit

Indicates the start of a loop. Flip the shape vertically to indicate the end of a loop.



Stored data

Indicates any type of stored data.



Connector

Indicates an inspection point.



Off-page connector

Use this shape to create a cross-reference and hyperlink from a process on one page to a process on another page.



Off-page connector



Off-page connector



Off-page connector



Or

Logical OR



Summing junction

Logical AND



Collate

Indicates a step that organizes data into a standard format.



Sort

Indicates a step that organizes items list sequentially.



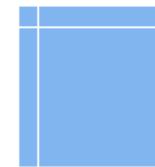
Merge

Indicates a step that combines multiple sets into one.



Database

Indicates a list of information with a standard structure that allows for searching and sorting.



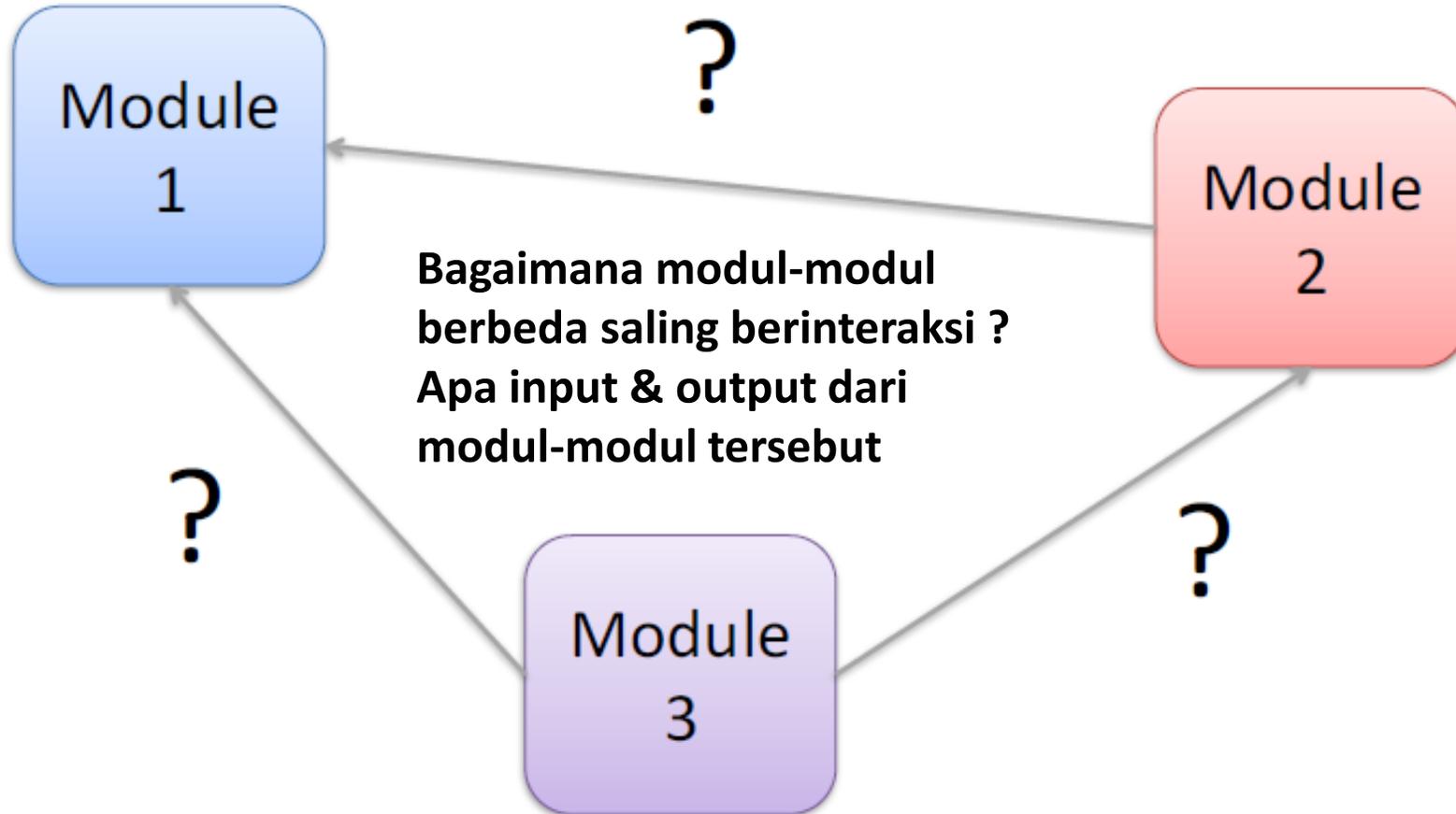
Internal storage

Indicates an internal storage device.

Symbol Flow Chart

Flow Chart Symbol	Meaning	Explanation
	Start and end	The symbol denoting the beginning and end of the flow chart.
	Step	This symbol shows that the user performs a task. (Note: In many flow charts steps and actions are interchangeable.)
	Decision	This symbol represents a point where a decision is made.
	Action	This symbol means that the user performs an action. (Note: In many flow charts steps and actions are interchangeable.)
	Flow line	A line that connects the various symbols in an ordered way.

Spesifikasi Antarmuka



UML

Unified Modelling Language

Mengapa Menggunakan UML?

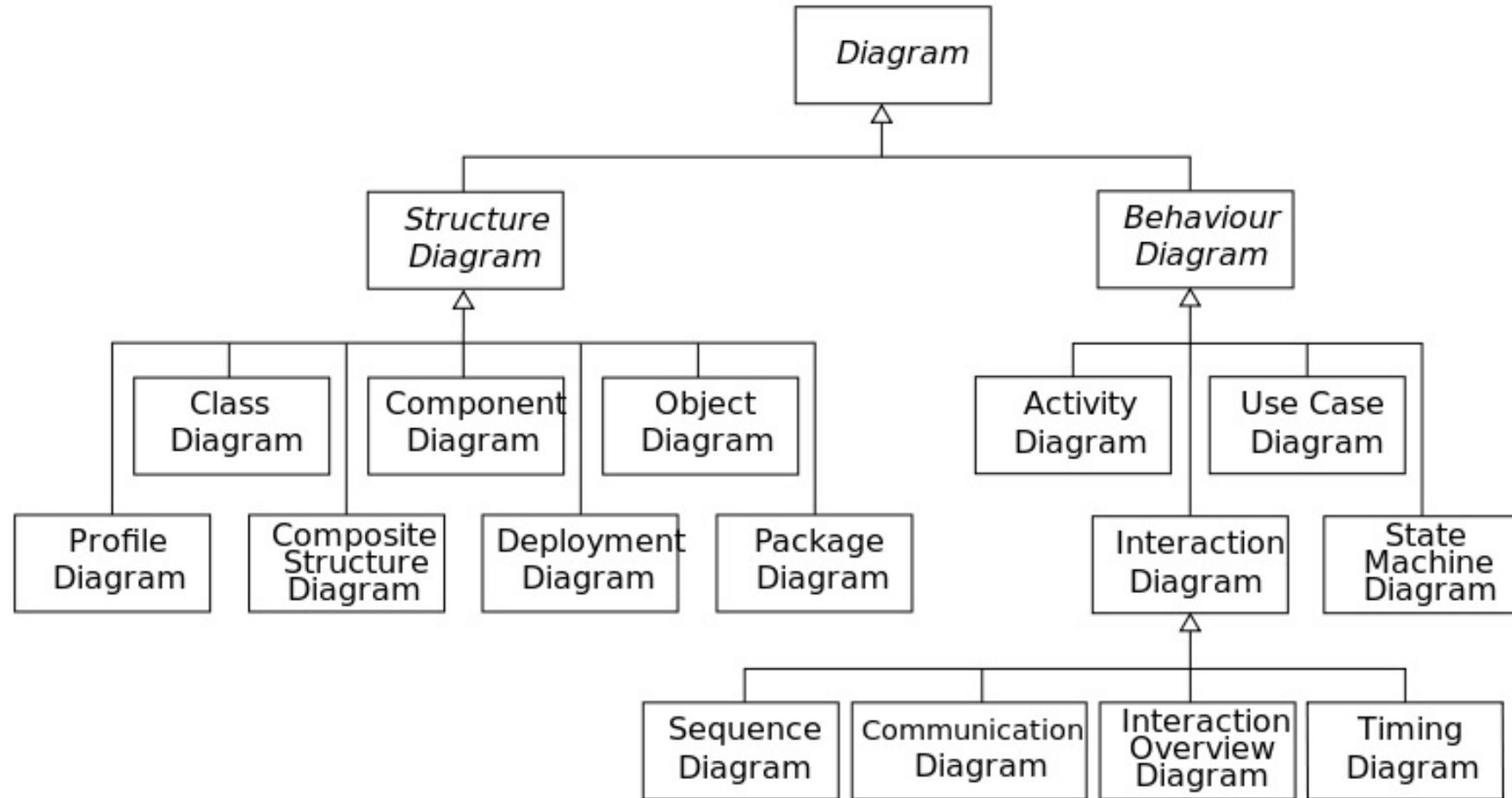
Rancangan

- Forward Design: Menggunakan UML sebelum coding. Akan lebih mudah untuk membuat kode dengan cara yang terstruktur
- Backward Design: menggunakan UML setelah coding sebagai dokumentasi.
- Saat ada perubahan di Rancangan, pastikan untuk mengupdate Kode sesuai dengan Rancangan baru tersebut

Kode Program (Source)

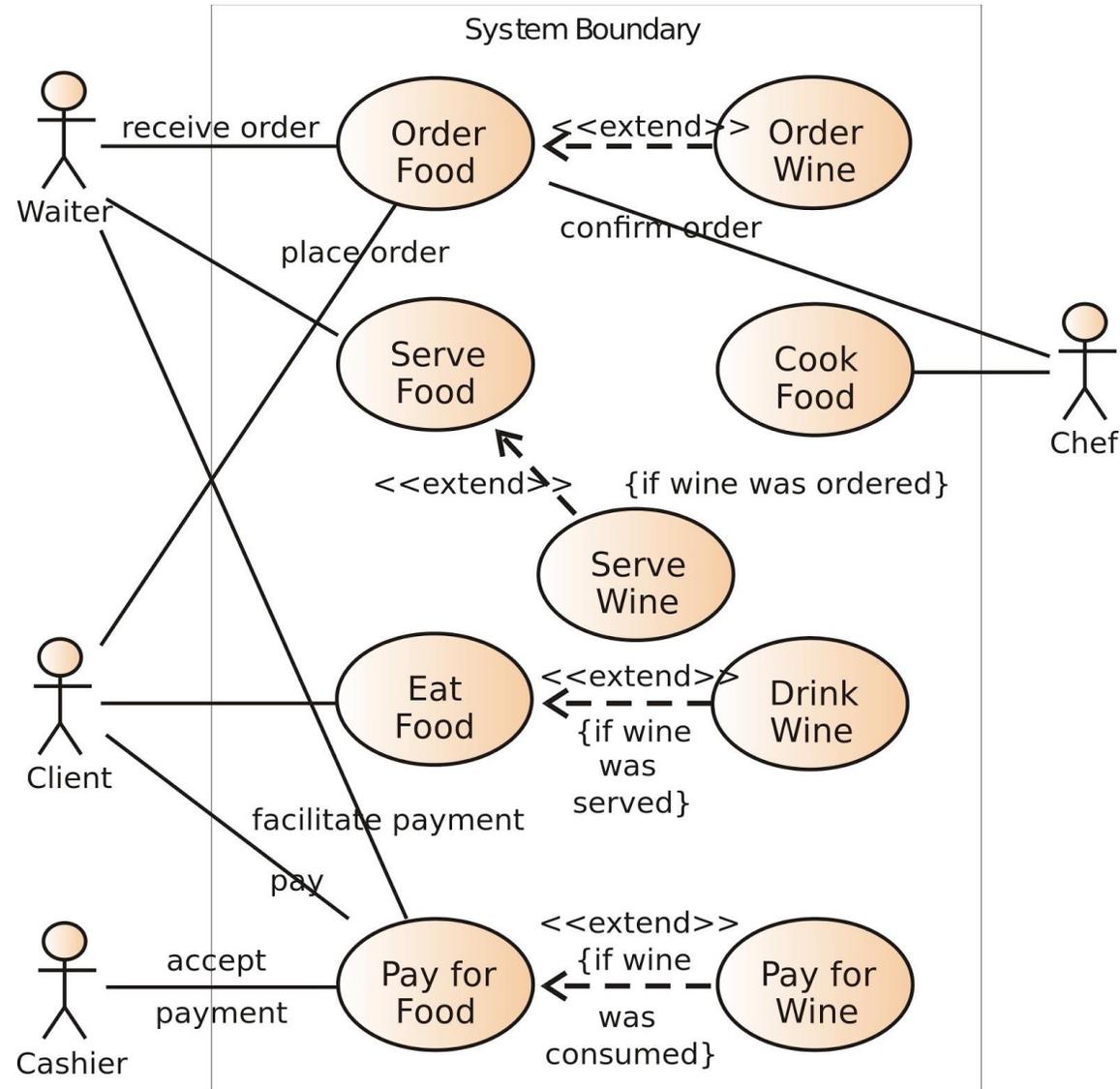
- Beberapa Tool dapat membangkitkan Kode dari diagram UML
- Saat ada perubahan dalam Kode, pastikan mengupdate diagram UMLnya

Jenis Diagram UML

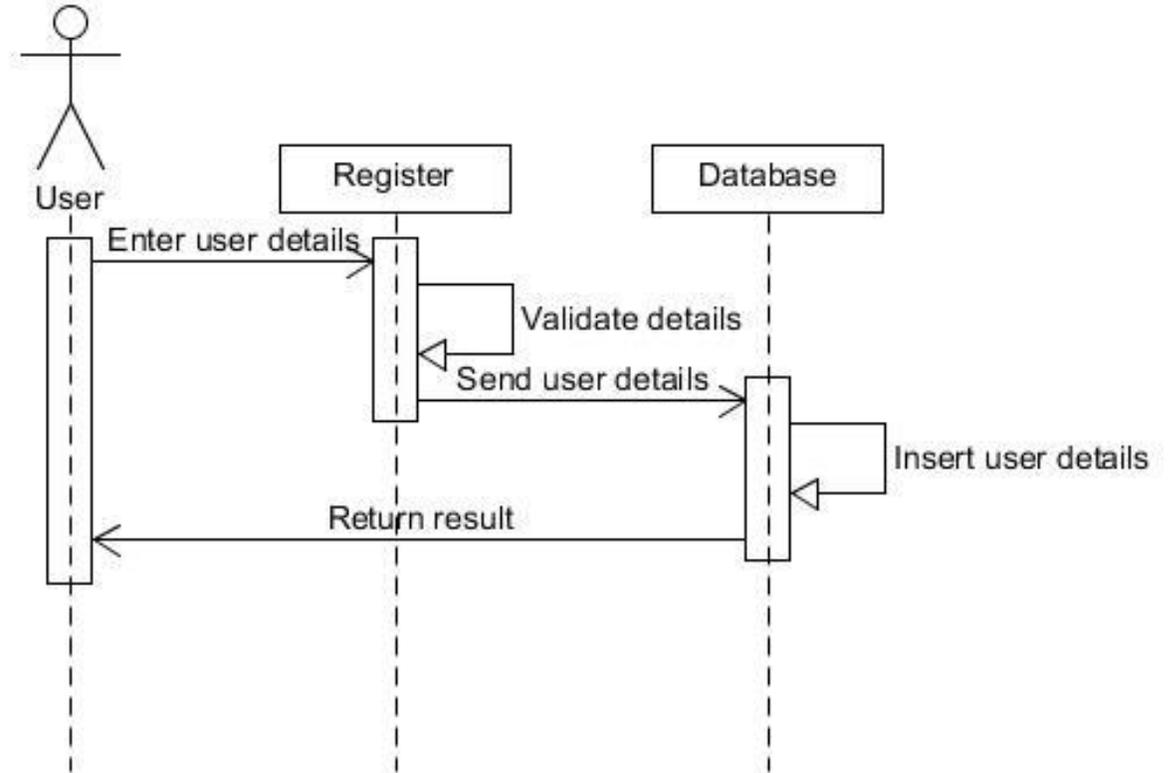
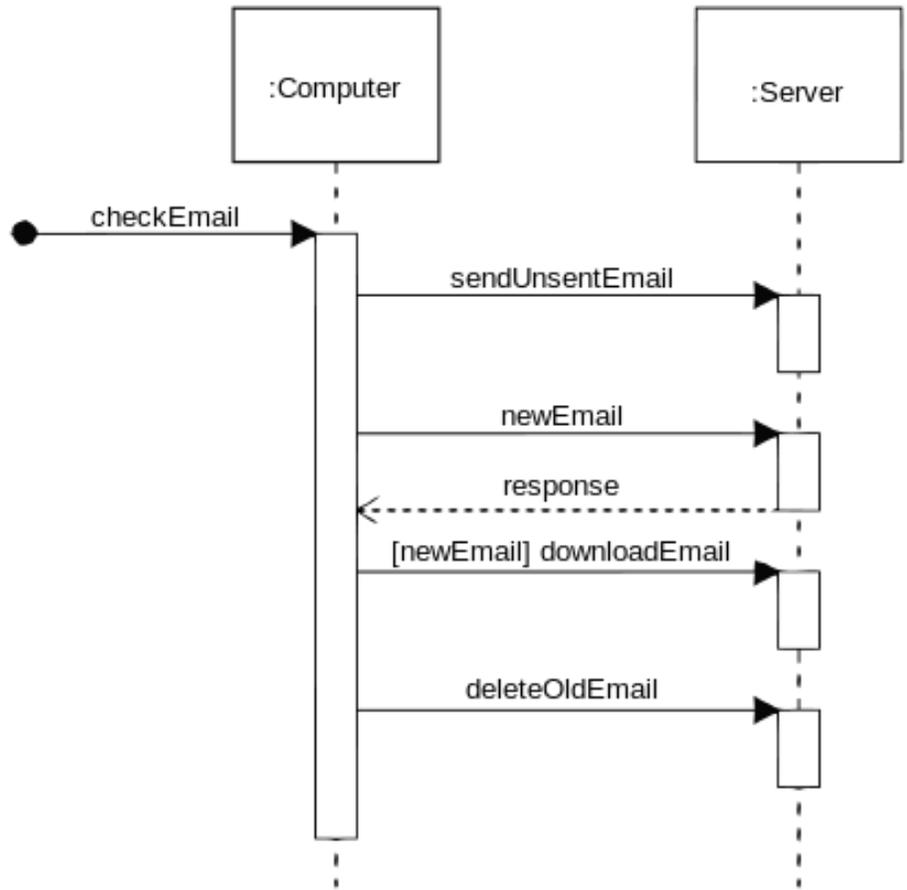


http://en.wikipedia.org/wiki/Unified_Modeling_Language

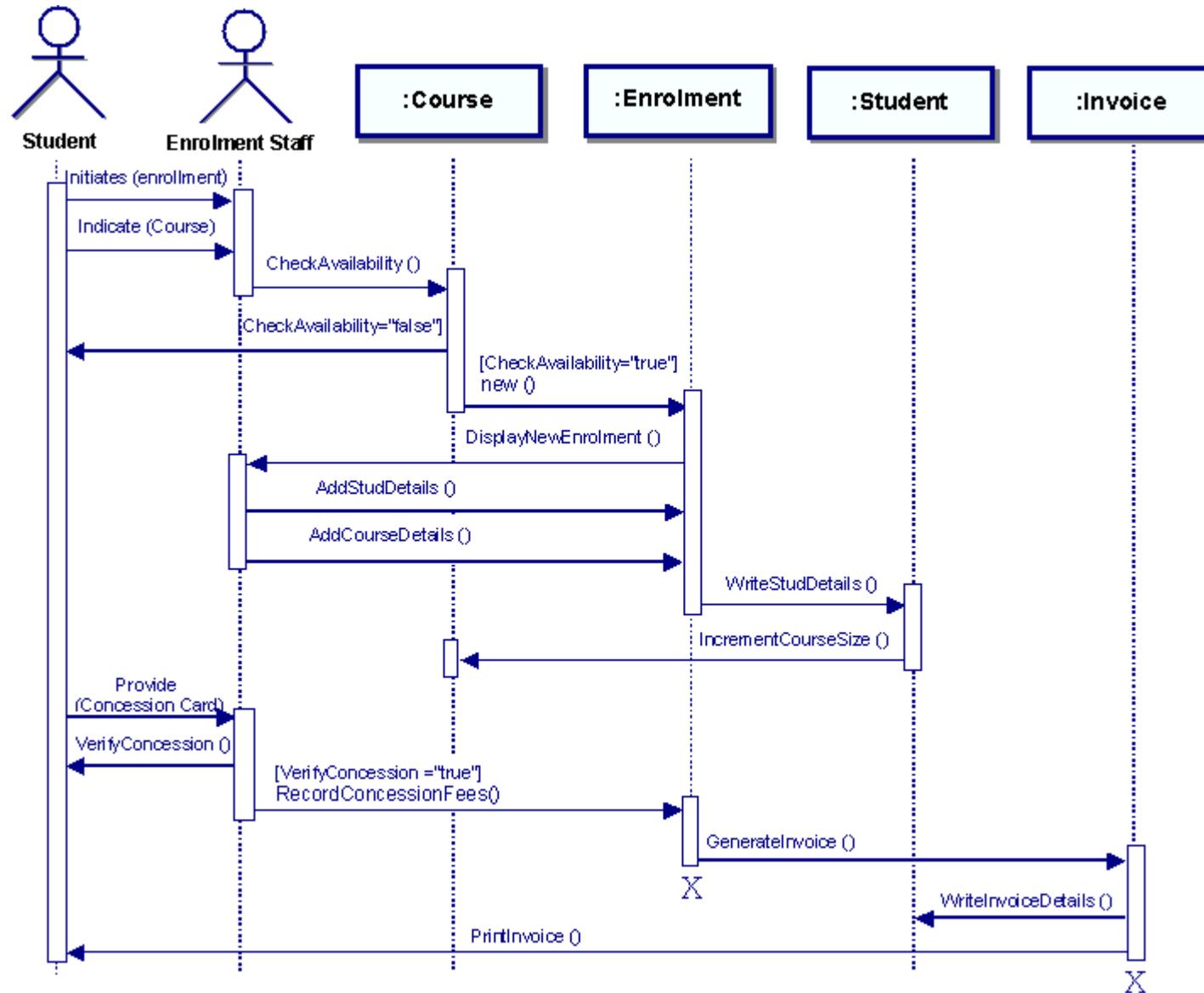
Use Case Diagram: Contoh



Sequence Diagram: Contoh

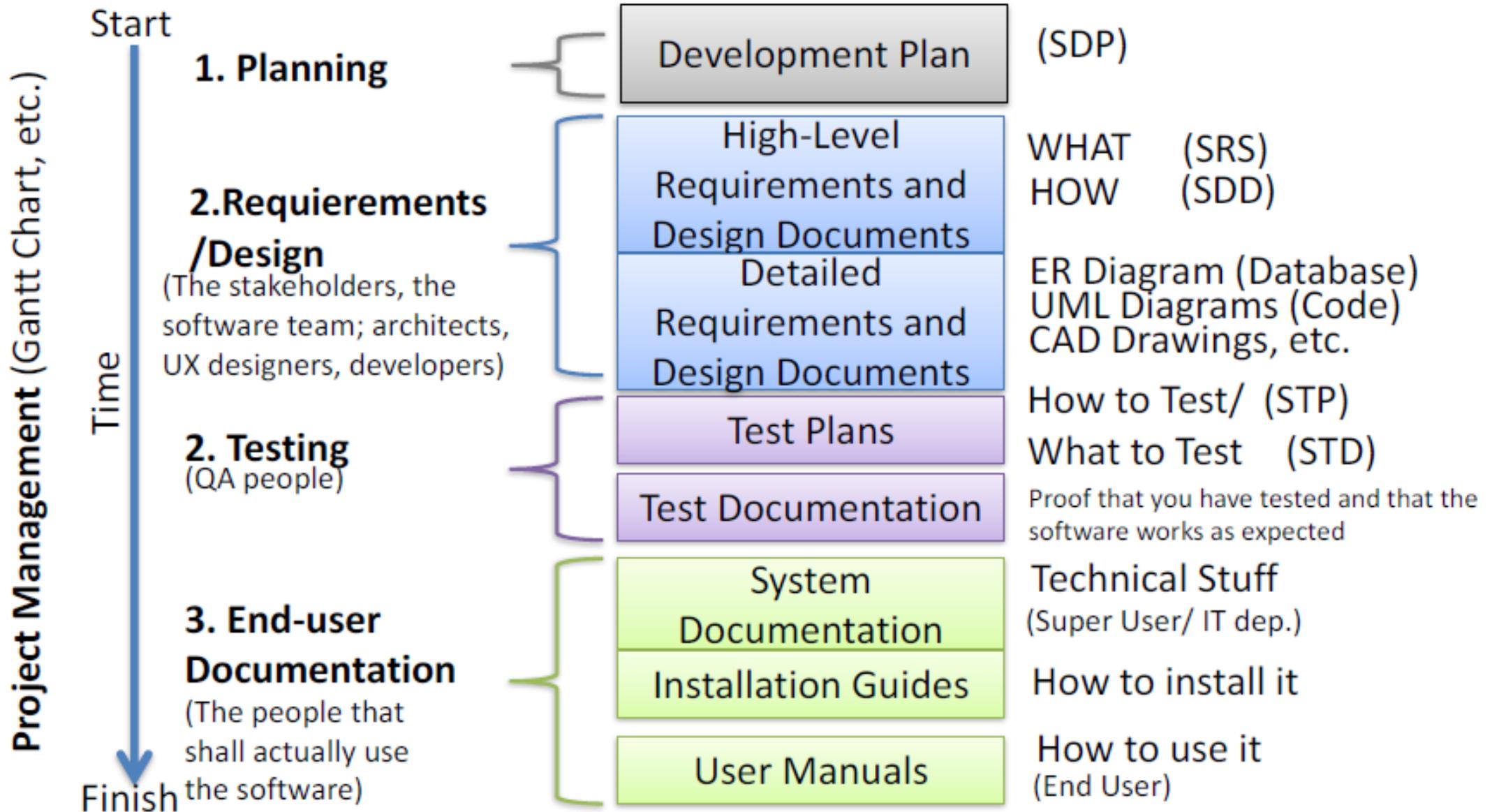


Sequence Diagram: Contoh



Dokumen SRS

Dokumentasi “Khas” Software



Dokumentasi Proyek

Dokumentasi yang dihasilkan selama proyek software dapat dibagi ke dalam 2 kategori:

- **Dokumentasi Proses**

Dokumen ini merekam proses pengembangan dan perawatan, seperti Rencana (Software Development Plan, Software Test Plan, ...), Jadwal (e.g., Gantt Charts), dll.

- **Dokumentasi Produk**

Dokumen ini menjelaskan produk yang dikembangkan. Dapat dibagi ke dalam 2 kategori:

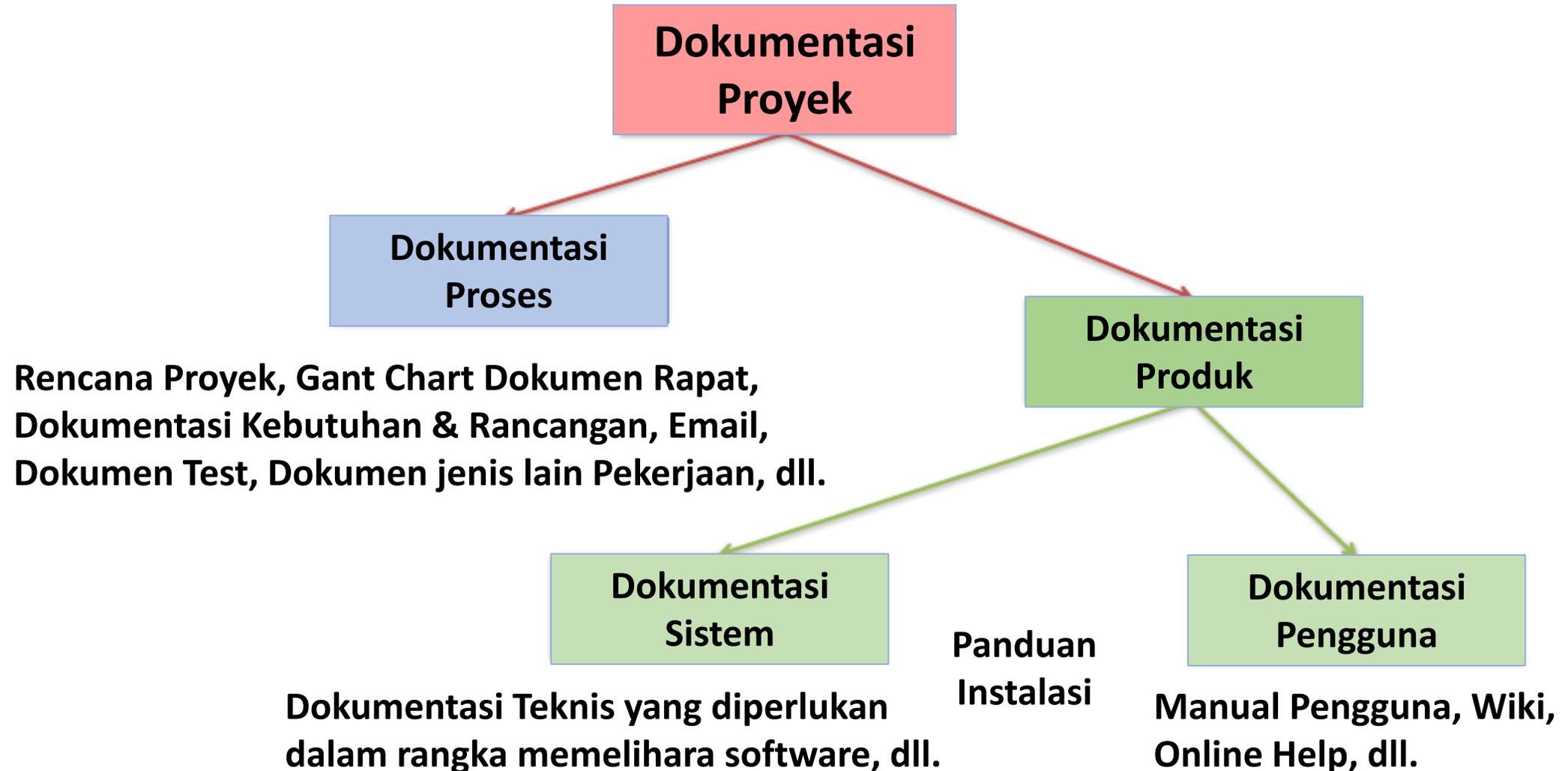
- **Dokumentasi Sistem**

Digunakan oleh insinyur dalam mengembangkan dan memelihara sistem

- **Dokumentasi Pengguna**

Digunakan oleh orang-orang yang menggunakan sistem.

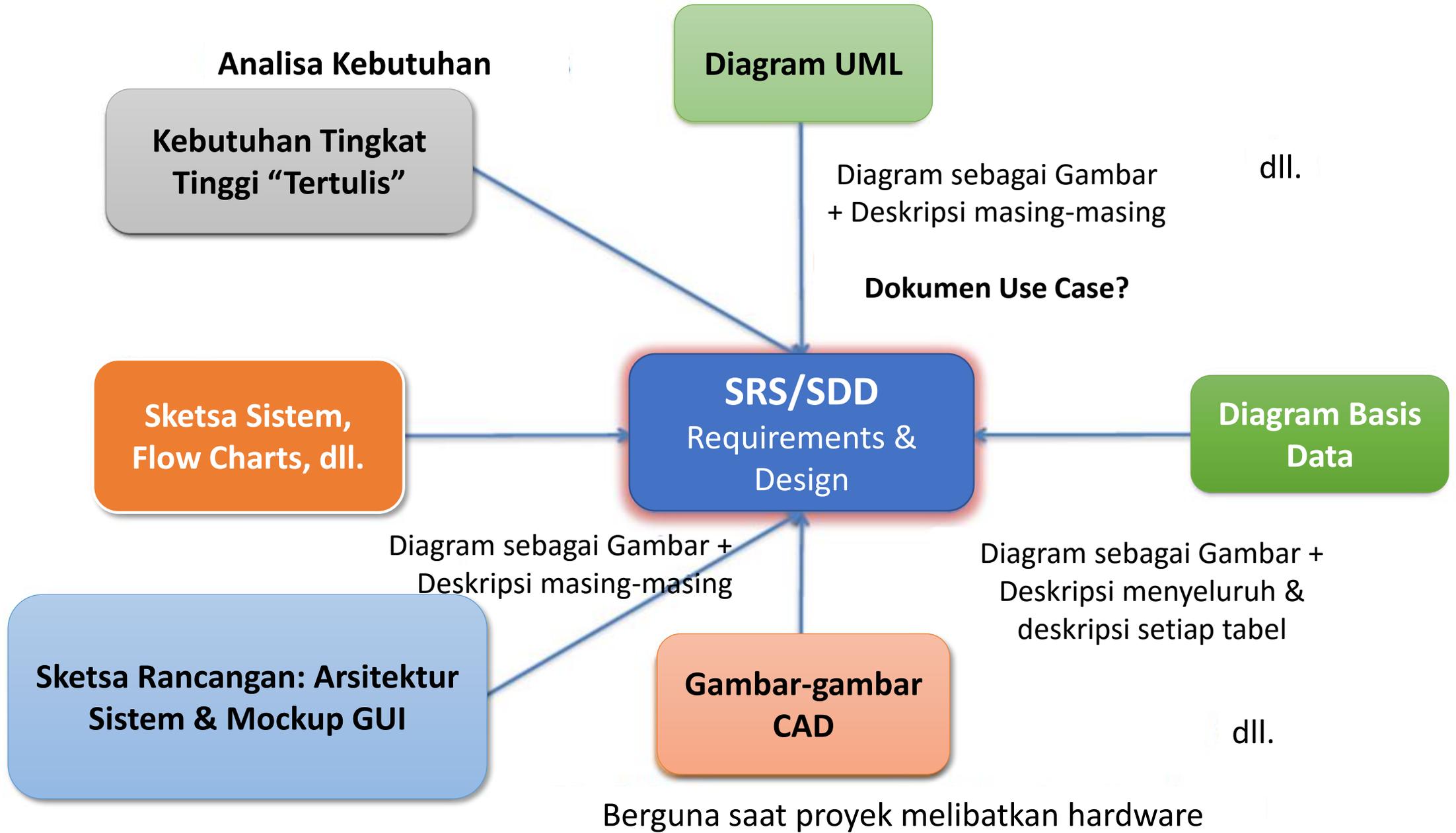
Kategori Dokumentasi



Dokumen Persyaratan Software

Software Requirements Specifications (SRS)

- Dokumen spesifikasi kebutuhan perangkat lunak (*software requirements specifications*, SRS) merupakan pernyataan resmi mengenai apa yang diperlukan oleh pengembang sistem.
- Harus memasukkan definisi dari kebutuhan pengguna dan spesifikasi dari kebutuhan sistem.
- BUKAN dokumen rancangan (desain). Se jauh mungkin, merupakan APA yang sistem akan kerjakan, bukan BAGAIMANA itu dilakukan.



Struktur Dokumen SRS

Bab	Deskripsi
Kata Pengantar	Mendefinisikan para pembaca yang diharapkan dari dokumen dan menjelaskan sejarah versinya, termasuk alasan pembuatan versi baru dan suatu rangkuman perubahan yang dibuat pada setiap versi.
Pendahuluan	Menjelaskan kebutuhan bagi sistem. Sebaiknya dengan ringkas mendeskripsikan fungsi-fungsi sistem dan menyebutkan bagaimana sistem akan bekerja sistem lain. Ini juga harus menjelaskan bagaimana sistem cocok dengan bisnis atau tujuan strategis organisasi secara menyeluruh dengan adanya perangkat lunak.
Daftar Istilah	Dengan jelas mendefinisikan istilah-istilah teknis yang digunakan dalam dokumen. Sebaiknya tidak membuat asumsi mengenai pengalaman atau kepakaran pembaca.
Definisi Kebutuhan Pengguna	Di sini, kita menjelaskan layanan-layanan yang tersedia bagi Pengguna. Kebutuhan sistem non-fungsional juga digambarkan dalam bagian ini. Deskripsi dapat menggunakan bahasa alami, diagram atau notasi lain yang dapat dipahami oleh Customer. Standar proses dan produk yang harus diikuti harus ditetapkan.
Arsitektur Sistem	Bab ini menyajikan suatu tinjauan level tinggi dari arsitektur sistem yang diharapkan, memperlihatkan distribusi fungsi lintas modul-modul sistem. Komponen-komponen arsitektural yang digunakan-ulang perlu disoroti.

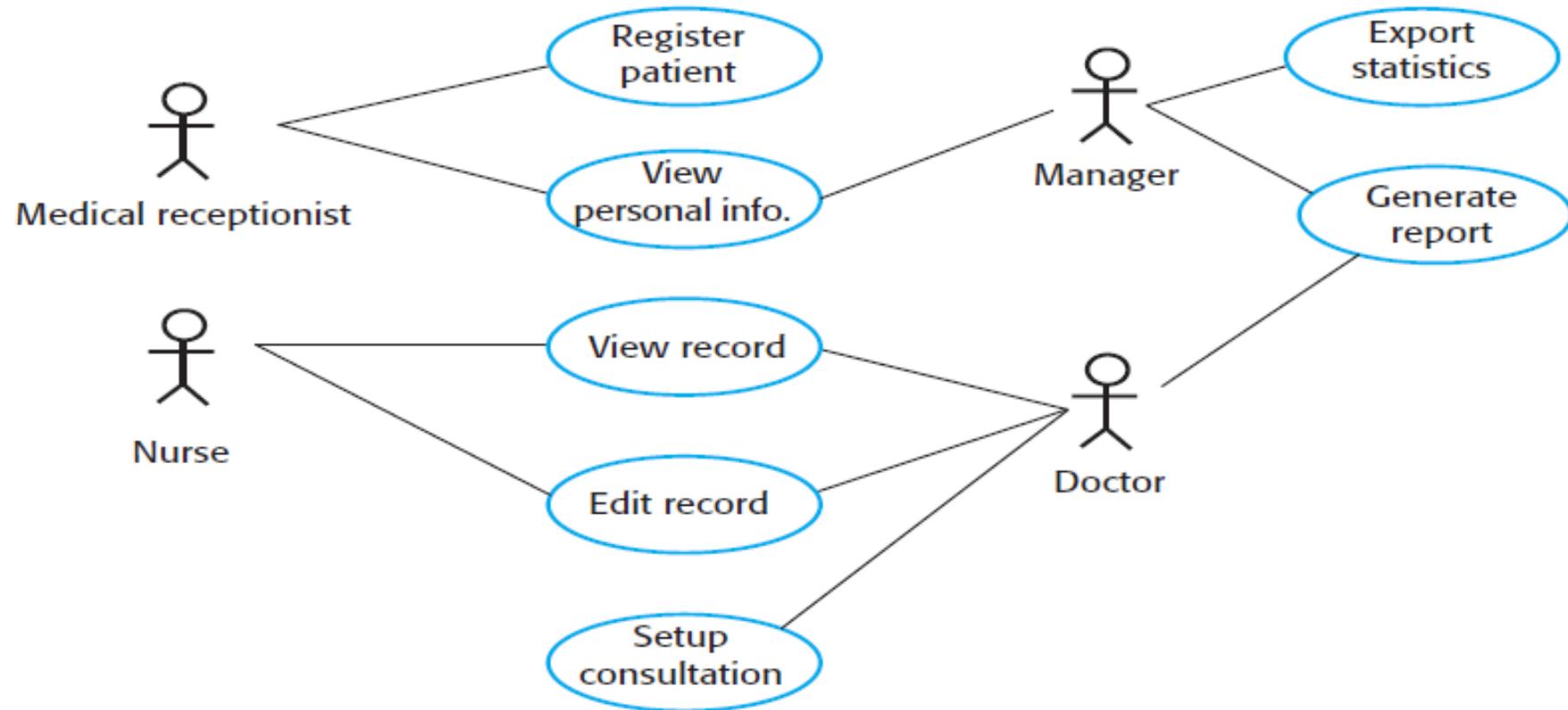
Struktur Dokumen SRS

Bab	Deskripsi
Spesifikasi Kebutuhan Sistem	Harus menjelaskan kebutuhan fungsional dan non-fungsional secara rinci. Jika diperlukan, rincian lebih lanjut dapat pula ditambahkan ke kebutuhan non-fungsional. Antarmuka ke sistem lain dapat didefinisikan.
Model-model Sistem	Ini dapat menyertakan model-model sistem grafis yang memperlihatkan hubungan antara komponen-komponen sistem dengan sistem dan sistem tersebut dengan lingkungannya. Contoh dari model yang mungkin adalah model Obyek, data-flow atau data semantik.
Evolusi Sistem	Bab ini menjelaskan asumsi-asumsi fundamental yang dijadikan basis bagi sistem, dan perubahan yang diharapkan dikarenakan evolusi hardware, kebutuhan pengguna yang berubah, dsb. Bagian ini berguna bagi Perancang sistem karena akan membantu Perancang menghindari keputusan rancangan yang akan membatasi kemungkinan perubahan di masa depan terhadap sistem.
Lampiran	Ini harus menyediakan informasi spesifik dan rinci yang terkait dengan aplikasi yang dikembangkan; misalnya deskripsi hardware dan database. Kebutuhan hardware mendefinisikan konfigurasi minimal dan optimal bagi sistem. Kebutuhan basis data mendefinisikan organisasi logis dari data yang digunakan oleh sistem dan relasi antar data.
Indeks	Beberapa indeks terhadap dokumen dapat disertakan. Selain indeks alfabet biasa, tambahan indeks diagram, fungsi, dll. tentu sangat menarik dan memudahkan pencarian informasi tertentu di dalam dokumen.

Pengguna dari SRS



UML dan Use Case



Model-model grafis, dilengkapi dengan anotasi teks digunakan untuk mendefinisikan kebutuhan fungsional bagi sistem; diagram use case dan sequence UML umum digunakan.

Referensi

- I. Sommerville, *Software Engineering*: Pearson, 2010.
- E. J. Braude and M. E. Bernstein, *Software Engineering. Modern Approaches*, 2 ed.: Wiley, 2011.
- F. Tsui, O. Karam, and B. Bernal, *Essentials of Software Engineering*, 3 ed.: Jones & Barlett Learning, 2014.
- Wikipedia. (2013). *Software Development Process*. Available: http://en.wikipedia.org/wiki/Software_process
- NTNU. (2013). *TDT4140 Systemutvikling*. Available: <http://www.ntnu.no/studier/emner/TDT4140>
- UiO. (2013). *INF1050 - Systemutvikling*. Available: <http://www.uio.no/studier/emner/matnat/ifi/INF1050/>

Pertanyaan?