

Konsep Perancangan Perangkat Lunak

REKAYASA PERANGKAT LUNAK

Semester Ganjil 2015/2016





Tujuan & Agenda Perkuliahan

Tujuan

- Memahami pengertian dan urgensi perancangan PL
- Memahami proses perancangan PL
- Memahami prinsip-prinsip perancangan PL
- Memahami parameter kualitas perancangan PL
- Memahami alat bantu dan dokumentasi yang diperlukan dalam perancangan PL

Agenda

- Pengertian, Urgensi dan Fungsi
- Proses
- Prinsip-prinsip
- Parameter Kualitas
- Alat bantu dan Dokumentasi

Perancangan Perangkat Lunak

Pengertian Perancangan Perangkat Lunak

- ❑ Perancangan Perangkat Lunak
 - ❑ Representasi rekayasa dari perangkat lunak yang akan dibangun, yang terfokus pada data, arsitektur, interface, dan komponen. Dapat dilacak terhadap kebutuhan-kebutuhan customer/perangkat lunak (Roger S. Pressman)
 - ❑ Membangun suatu solusi permasalahan yang memenuhi kebutuhan-kebutuhan perangkat lunak (Ian Sommerville)

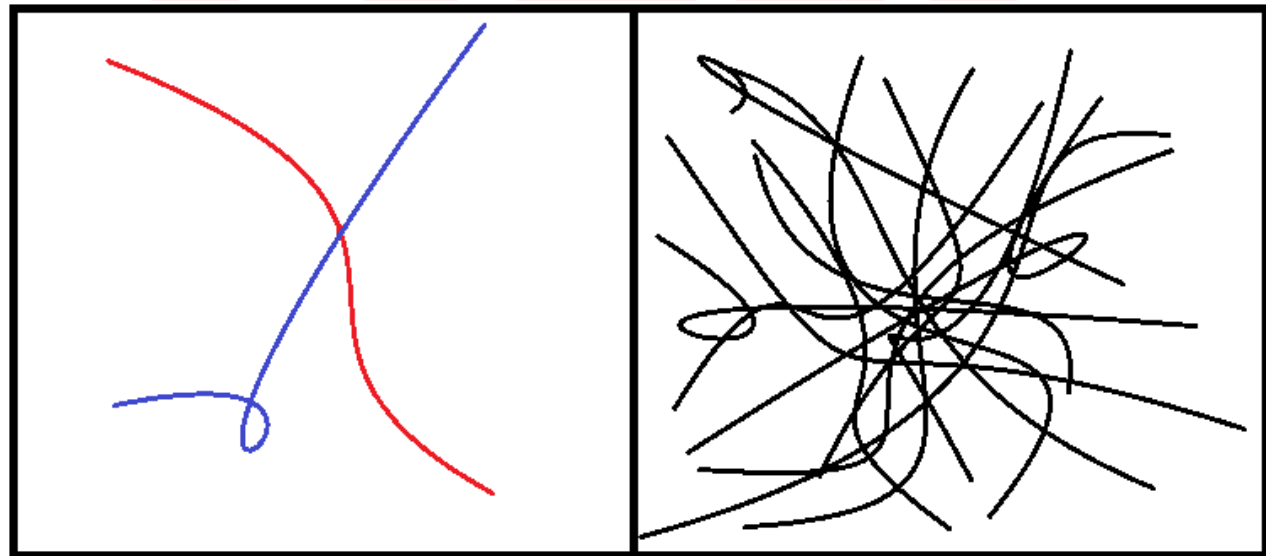




Perancangan Perangkat Lunak

Wicked Problem

- There is no definite formulation
- There is no stopping rule
- Solutions are not simply true or false
- Every wicked problem is a symptom of another problem



Traditional Problem

Wicked Problem



Urgensi dan Fungsi Perancangan Perangkat Lunak

- ❑ Mengapa → KUALITAS : perancangan adalah satu2nya jalan untuk mentransformasi secara akurat kebutuhan2 customer ke dalam sebuah produk akhir berupa perangkat lunak yang berkualitas tinggi
- ❑ Fungsi → Sebagai dasar dari semua aktifitas rekayasa perangkat lunak yang mengikutinya (implementasi/coding, pengujian dan validasi)



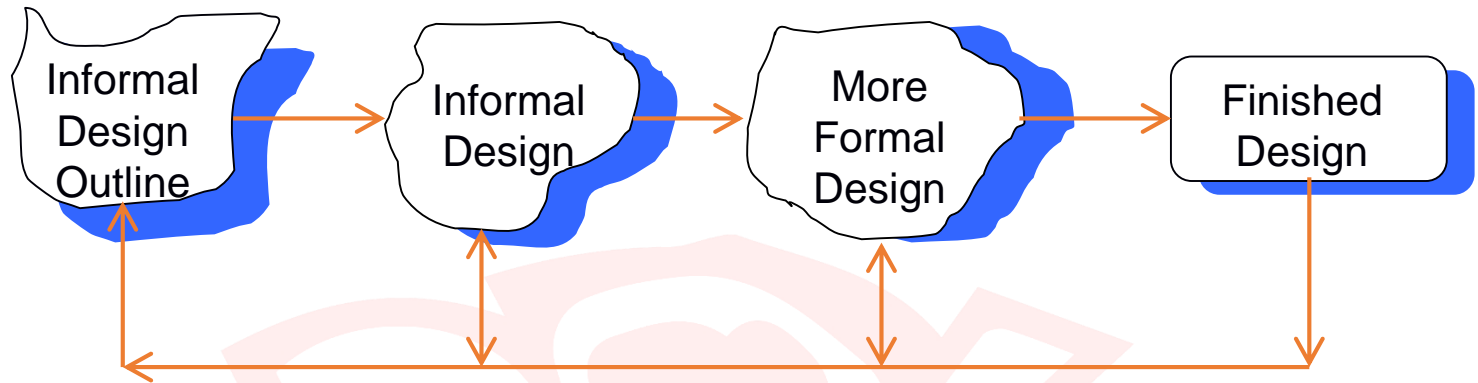
Proses - Proses Perancangan Perangkat Lunak

- Perancangan data
 - Mentransformasikan model domain informasi dari hasil analisis menjadi struktur data yang dibutuhkan dalam implementasi PL
- Perancangan arsitektural
 - Mendefinisikan elemen-elemen utama (sub-sistem) dan relasi antar elemen-elemen utama dari struktur PL
- Perancangan interface
 - Menjelaskan bagaimana PL berkomunikasi di dlm PL itu sendiri maupun dengan sistem lain yang berinteraksi atau dengan manusia sebagai pemakainya
- Perancangan komponen
 - Dekomposisi sub-sistem menjadi komponen detail



Proses - Proses Perancangan Perangkat Lunak

❑ Iteratif



- ❑ Kriteria perancangan yang baik (McGlaughlin) :
 - ❑ Harus mengimplementasikan seluruh kebutuhan eksplisit yang ada dan yang implisit yang diinginkan oleh customer
 - ❑ Harus mudah dibaca, dimengerti nantinya sebagai panduan implementasi maupun pengujian
 - ❑ Harus lengkap menggambarkan keseluruhan PL yang akan dibuat



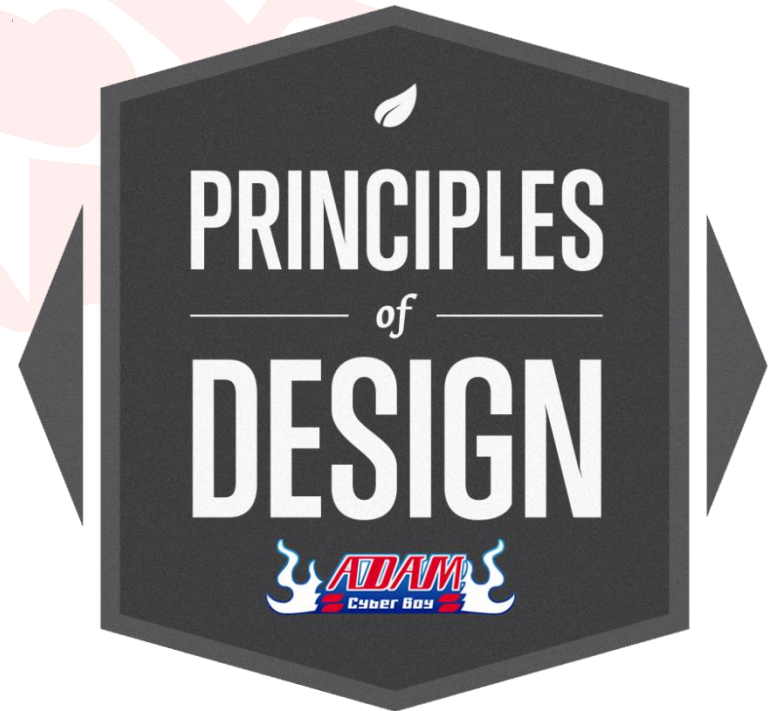
Prinsip - Prinsip Perancangan Perangkat Lunak

- Terdiri dari proses dan model
- Prinsip dasar (Davis) :
 - Proses perancangan tidak boleh menggunakan pendekatan 'kaca mata kuda'
 - Perancangan harus bisa dilacak ke model analisisnya
 - Perancangan tidak seharusnya mengulangi rancangan yang sudah ada → reusing
 - Perancangan harus sedapat mungkin mendekati kenyataan yang ada di domain permasalahannya
 - Perancangan seharusnya seragam (ada konvensi) dan integral (interface antar komponen)
 - Perancangan ≠ implementasi, implementasi ≠ perancangan



Prinsip - Prinsip Perancangan Perangkat Lunak

- ❑ Perancangan harus bisa dievaluasi pada saat berlangsung (tidak setelah selesai semuanya)
- ❑ Perancangan harus terstruktur → bisa mengakomodasi perubahan



Parameter Kualitas Perancangan Perangkat Lunak



Parameter Kualitas Perancangan Perangkat Lunak

- Cohesion
- Coupling
- Understandability
- Adaptability





Parameter Kualitas Perancangan Perangkat Lunak

- Cohesion
 - Ukuran tingkat kedekatan relasi antar elemen-elemen dalam satu komponen
 - Jika terjadi perubahan pada sebuah komponen → tingkat kohesifitas menentukan
 - Semakin tinggi semakin baik hasil rancangannya
 - Penurunan klas akan mengurangi tingkat kohesifitas
 - Untuk memahami sebuah klas/komponen secara lengkap, harus melihat klas atasnya (super-class)
 - Ada beberapa tingkat kohesifitas, tetapi cukup sulit untuk mengklasifikasikan dengan tepat

Parameter Kualitas Perancangan Perangkat Lunak

❑ Cohesion Level

❑ Logical association (weak)

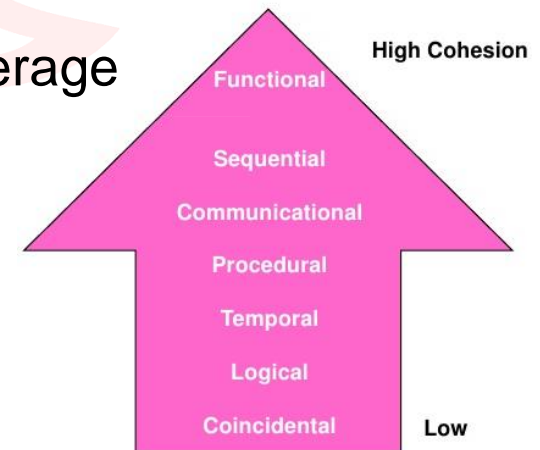
- ❑ Kesamaan fungsi/logika dari elemen2 sebuah komponen, tetapi aksinya berbeda-beda

- ❑ Contoh :
 - output text to screen
 - output line to printer
 - output record to file

❑ Temporal cohesion (weak)

- ❑ Elemen2 yang diaktifkan dalam waktu yang sama

- ❑ Contoh :
 - clear screen
 - open file
 - initialize time_average





Parameter Kualitas Perancangan Perangkat Lunak

Cohesion Level

Procedural cohesion (weak)

Sebuah komponen terdiri dari sekumpulan elemen2 yang membentuk satu rangkaian proses

Contoh : calculate quiz_score
 calculate final_score
 analyze attendance_list

Sequential cohesion (medium)

Output dari satu elemen menjadi input bagi elemen yang lain

Contoh : get temperature_value_to_convert
 convert to fahrenheit
 output conversion_result to screen



Parameter Kualitas Perancangan Perangkat Lunak

Cohesion Level

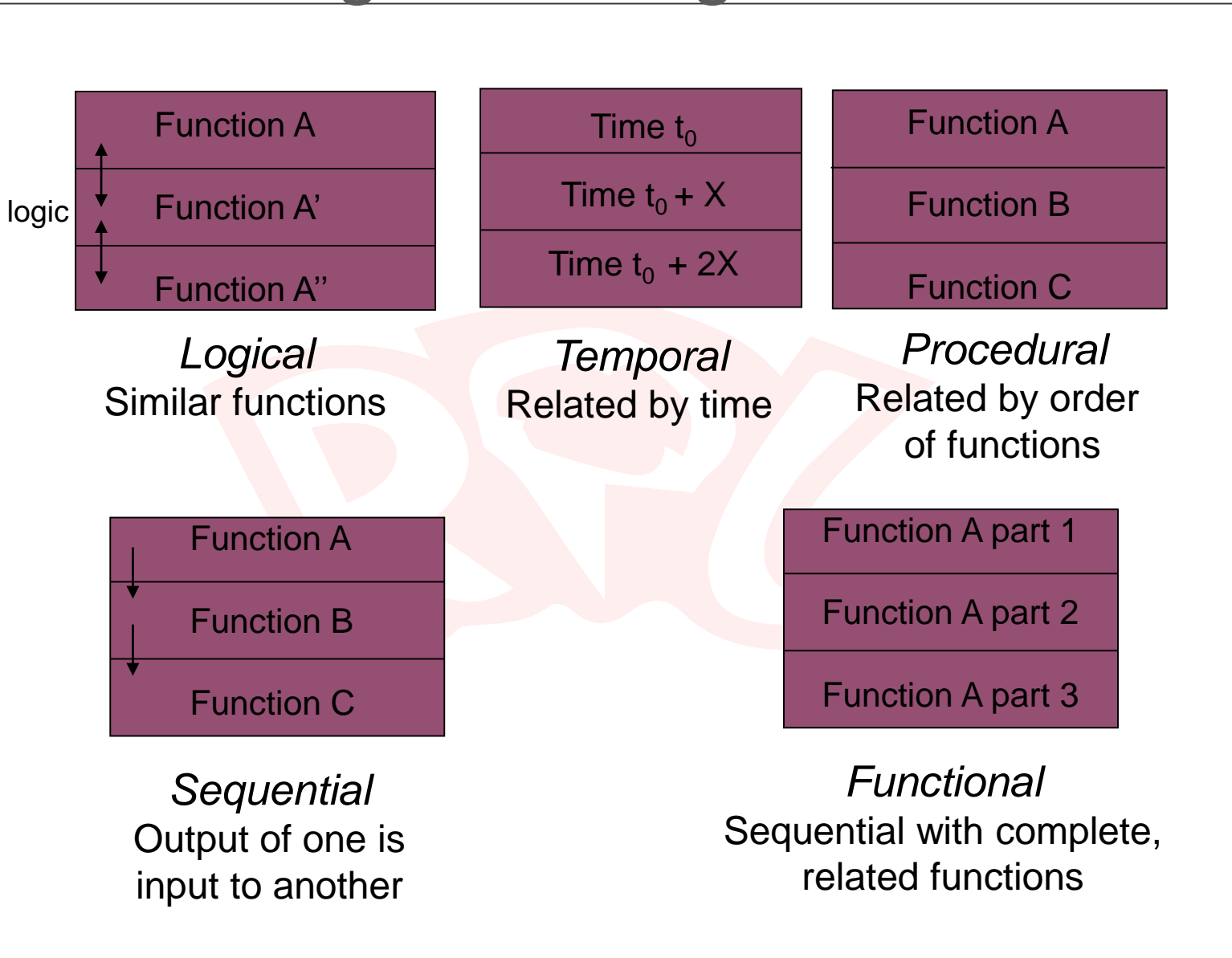
Functional cohesion (strong)

- Tipe kohesi yang paling optimal
- Satu fungsi dan satu data tertentu untuk setiap komponen
- Elemen2 dalam satu komponen untuk memenuhi fungsi yang terdefinisi tsb.
- Contoh : calculate average

Object cohesion (strong)

- Setiap operasi memiliki kemampuan untuk memanipulasi atribut yang dibutuhkan dari sebuah klas
- Contoh : display temperature
 log temperature
 analyze temperature

Parameter Kualitas Perancangan Perangkat Lunak





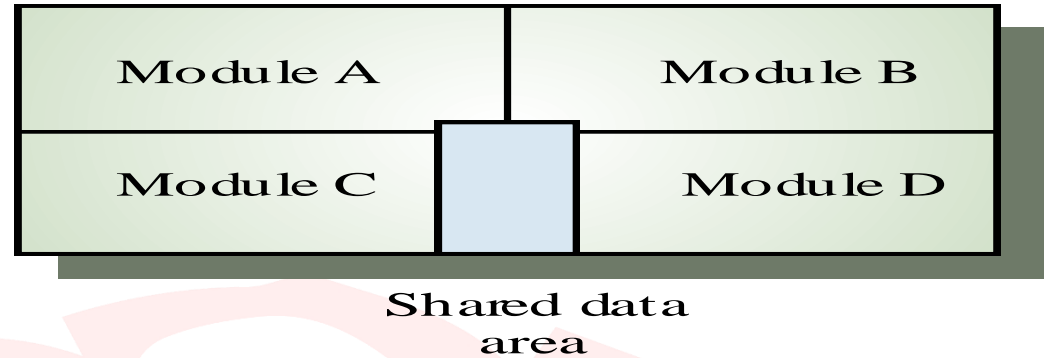
Parameter Kualitas Perancangan Perangkat Lunak

- Coupling
 - Ukuran tingkat kedekatan relasi antar komponen
 - Semakin rendah semakin baik hasil rancangannya
 - Jika terjadi perubahan pada sebuah komponen, maka tingkat kopling menentukan
 - Penurunan klas akan menaikkan tingkat kopling
 - OO systems memberikan tingkat kopling yang rendah : tidak ada shared data

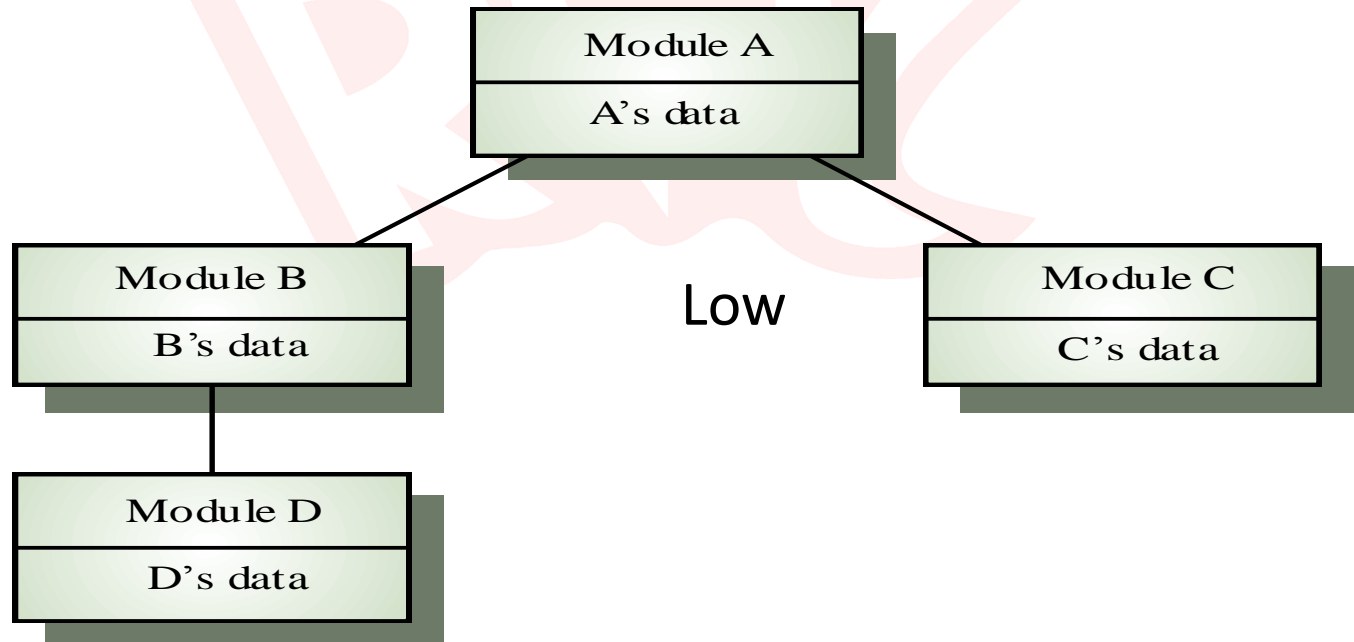
Parameter Kualitas Perancangan Perangkat Lunak

❑ Coupling

High



LOW





Parameter Kualitas Perancangan Perangkat Lunak

Understandability

- Mudah dipahami : implementasi, pengujian dan perawatan (jika terjadi perubahan)
- Berhubungan dengan beberapa karakteristik komponen :
 - Cohesion dan coupling : can the component be understood on its own ?
 - Naming : are meaningful names used ?
 - Documentation : is the design well-documented ?
 - Complexity : are complex algorithms used ?





Parameter Kualitas Perancangan Perangkat Lunak

- Adaptability
 - Seberapa mudah dilakukan perubahan perancangan
 - Perancangan yang adaptif :
 - Koping rendah
 - Dokumentasi bagus dan up to date
 - Traceability yang tinggi
 - Kohesi tinggi
 - Penurunan klas : meningkatkan adaptability



Alat Bantu Perancangan Perangkat Lunak



Alat Bantu Perancangan Perangkat Lunak

- Functional Design :
 - Aplikasi pengolah model : Visio, dll.
 - Aplikasi pengolah kata : MS Word, dll.
- Object-Oriented Design :
 - Aplikasi pengolah model : Visio, dll.
 - Aplikasi pengolah kata : MS Word, dll.
 - CASE Tool : Rational Soda for Word, Rational Rose, OODTool (Coud-Yourdon) dll.



Dokumentasi Perancangan Perangkat Lunak



Dokumentasi Perancangan Perangkat Lunak

DOD2167A (SDD)

1.Scope

- 1.1. Identification
- 1.2. System overview
- 1.3. Document overview

2.Referenced Documents

3.Preliminary Design

- 3.1. CSCI overview
- 3.2. CSCI design description

4.Detailed Design

5.CSCI Data

6.CSCI Data Files

7.Requirements Traceability

8.Notes



Penutup

- Perancangan PL memiliki peran yang sangat penting dalam pengembangan PL karena mentransformasi kebutuhan2 customer menjadi produk akhir PL yang berkualitas
- Proses dalam perancangan PL terdiri perancangan data, arsitektural, interface dan komponen
- Parameter kualitas perancangan PL terdiri dari cohesion, coupling, understandability dan adaptability

Terimakasih v^^