Desain Perangkat Lunak

Team Teaching Mata Kuliah Rekayasa Perangkat Lunak
Jurusan Teknologi Informasi

Politeknik Negeri Malang

Tujuan

- Memahami desain perangkat lunak

Preface

Requirement	Analisis
Luaran dari tahap requirement menjadi masukan pada tahapan analisis. Bagian ini menjadi penting karena merupakan tahapan pemahaman terhadap domain masalah yang akan diselesaikan oleh perangkat lunak.	luaran dari tahapan analisis digunakan pada tahap desain. Hasil dari proses ini digunakan sebagai acuan untuk membuat implementasi perangkat lunak.

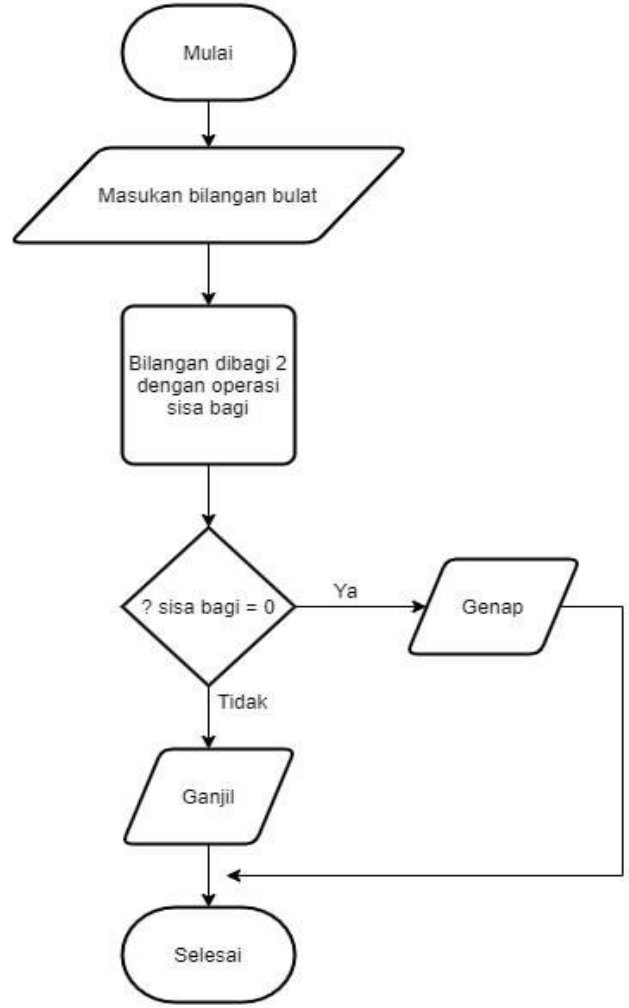
Tahapan desain menerjemahkan kebutuhan perangkat lunak ke dalam model yang dapat dipahami oleh pengembang perangkat lunak

Preface

	Paradigm	Diagrams
1	Process-oriented Paradigm	Flowchart
2	Data-oriented Paradigm	DFD
3	Object-oriented Paradigm (data + process)	UML

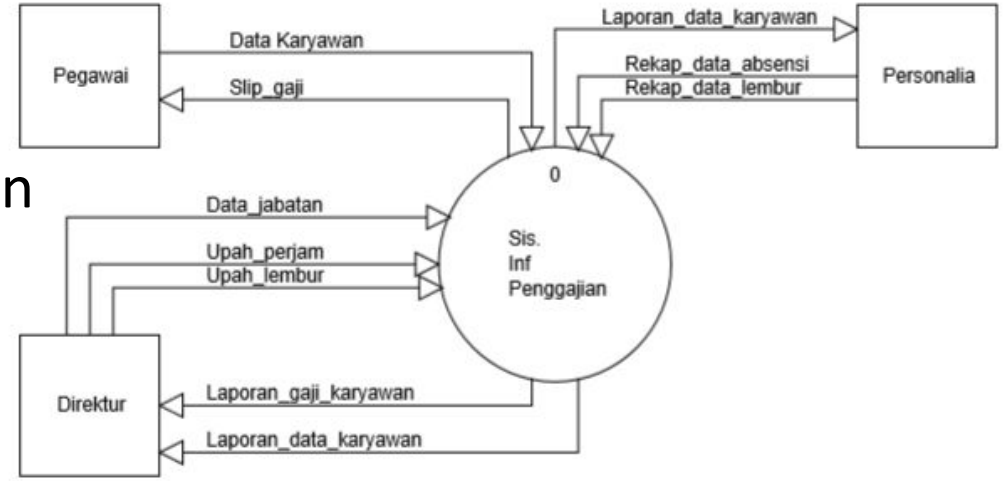
Flowchart

- **Flowchart** adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah.
- Flowchart berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari pembuatan program yang melibatkan banyak orang sekaligus.
- Penggunaan flowchart dalam dunia pemrograman juga merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan non-teknis, serta program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran.



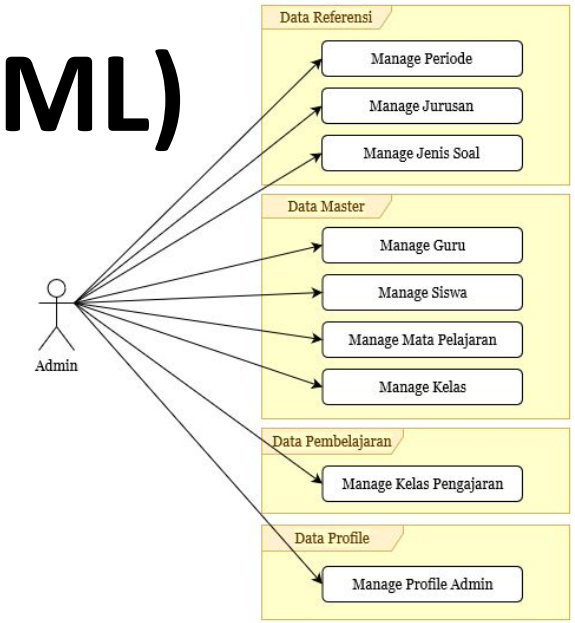
Data Flow Diagram (DFD)

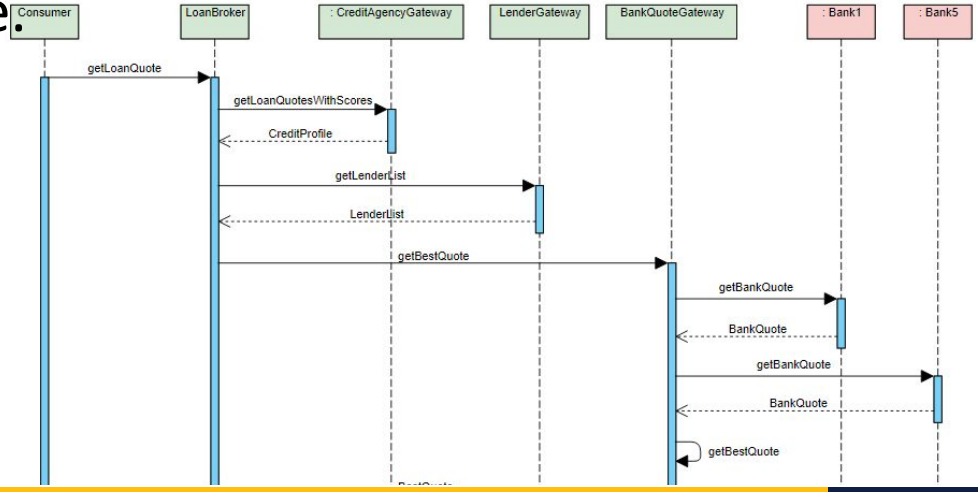
- **Data Flow Diagram (DFD)** adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai yang dihubungkan satu sama lain dengan alur data, baik secara komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.
- DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi system dan berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.



Unified Modeling Language (UML)

- **Unified Modelling Language (UML)** adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek.
- UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan blueprint sebuah software.





Pengertian

- Desain perangkat lunak merupakan tahapan pengembangan perangkat lunak yang hasilnya dapat digunakan oleh pengembang perangkat lunak untuk membuat program.

  - **Input software design**: Model analisa kebutuhan dan dokumentasi spesifikasi
  - **Output software design**: Model desain dan dokumentasi spesifikasi desain

Prinsip-prinsip Desain

Desain perangkat lunak, baik desain konvensional/prosedural maupun berorientasi objek, dilakukan dengan mengacu pada prinsip-prinsip atau pedoman-pedoman tertentu untuk mempermudah proses desain itu sendiri dan untuk menghasilkan desain berkualitas tinggi.

Prinsip-prinsip tersebut merupakan prinsip-prinsip yang umum yang dapat diterapkan pada proyek apapun.

Proses Desain

Desain bertujuan untuk membentuk sebuah model yang siap untuk diimplementasikan ke dalam program. Dalam membentuk model desain, terdapat serangkaian proses yang perlu dilakukan dengan tetap berpegang pada prinsip-prinsip desain. Semua kegiatan desain perlu untuk didokumentasikan dan proses dokumentasi perlu manajemen yang baik.

Dalam desain berorientasi objek diuraikan beberapa kegiatan yang lain pada tahapan desain

- Mendefinisikan tujuan desain.
- Mendefinisikan subsistem.
- Pemetaan subsistem ke dalam platform yang digunakan
- Pengelolaan persistent data.
- Mendefinisikan kendali akses.
- Mendefinisikan kendali aliran (control flow).
- Mendefinisikan boundary condition.

Jenis desain perangkat lunak

- **Desain data**: mentransformasi model domain informasi ke struktur data
- **Desain arsitektur**: menggambarkan relasi antar elemen struktural utama program
- **Desain interface** : mendeskripsikan bagaimana software berkomunikasi dengan pengguna
- **Desain prosedur**: mentransformasikan elemen structural arsitektur program ke sebuah deskripsi prosedur dari komponen software.

Desain Data

Desain data adalah aktivitas pertama dari empat aktivitas desain yang dilakukan selama rekayasa perangkat lunak. Proses pemilihan struktur dalam menentukan desain yang paling efisien sesuai kebutuhan.

Tujuan

- Untuk mendapatkan struktur data yang baik sehingga diperoleh program yang lebih efektif dan mengurangi kompleksitas pengembangan software

Tools

- Basisdata : ERD (Entity Relationship Diagram)

Desain Arsitektur

Sasaran utama desain arsitektur adalah untuk mengembangkan struktur program modular dan merepresentasikan hubungan kontrol antar modul. Desain arsitektur merupakan desain makro / struktur yang mencerminkan kualitas serta fungsi dari perangkat lunak. Aktivitas pembentukan arsitektur merupakan aktivitas dekomposisi, yaitu membagi perangkat lunak menjadi elemen-elemen/modul.

Tujuan

- Untuk mendapatkan arsitektur yang baik dan saling berhubungan antar modul dalam perangkat lunak

Tools

- Untuk desain perangkat lunak berbasis object, bisa menggunakan Unified Modeling Language (UML) Diagram

Desain Antarmuka (*interface*)

Desain *interface* memberikan suatu gambaran mengenai struktur program kepada perekayasa perangkat lunak (programmer), sehingga mempermudah bagaimana tampilan dan interaksi yang terjadi pada perangkat lunak.

Desain interface memfokuskan diri pada 3 area:

- Desain *interface* antar modul dalam perangkat lunak
- Desain *interface* antara perangkat lunak dan entitas eksternal
- Desain *interface* manusia dengan komputer

Tools

- Untuk desain perangkat lunak berbasis object, bisa menggunakan UML Component Diagram dan tool UI Diagram seperti Figma/Canva dll

Desain Prosedur

Desain prosedur dilakukan setelah diselesaikannya perancangan desain data, arsitektur, dan antar muka perangkat lunak. Desain ini digunakan untuk merancang bagaimana prilaku dari sebuah perangkat lunak.

Tujuan

- Untuk mendapatkan spesifikasi procedural akan prilaku yang ada da;am perangkat lunak

Tools

- Untuk desain perangkat lunak berbasis object, bisa menggunakan Unified Modeling Language (UML) Diagram

Pemodelan Sistem

- Pemodelan sistem adalah proses pengembangan model abstrak dari sebuah sistem, dengan setiap model menyajikan pandangan atau perspektif yang berbeda dari sistem tersebut.
- Pemodelan sistem secara umum berarti merepresentasikan sistem dengan menggunakan notasi grafis (diagram), seperti Unified Modeling Language (UML).
- Model digunakan selama proses rekayasa kebutuhan untuk membantu
  - ✔ mendapatkan persyaratan sistem,
  - ✔ untuk mendeskripsikan sistem kepada insinyur yang mengimplementasikan sistem
  - ✔ untuk mendokumentasikan struktur dan operasi sistem

Perspektif Pemodelan Sistem

Terdapat beberapa pandangan (perspektif) dalam pemodelan sistem

1. **Perspektif eksternal**, memodelkan konteks atau lingkungan sistem.
2. **Perspektif interaksi**, memodelkan interaksi antara sistem dan lingkungannya atau antara komponen sistem.
3. **Perspektif struktural**, memodelkan organisasi sistem atau struktur data yang diproses oleh sistem.
4. **Perspektif perilaku**, memodelkan perilaku dinamis dari sistem dan bagaimana responsnya terhadap suatu proses.

Pemodelan Sistem dengan UML

- UML secara luas digunakan dalam memodelkan sebuah system perangkat lunak.
- Dalam implementasinya, terdapat **lima jenis diagram** (dari banyak diagram ±13 jenis diagram) yang dapat mewakili esensi dari suatu system, yaitu
    1. ***Activity Diagram***, diagram yang menunjukkan aktivitas yang terlibat dalam suatu proses atau dalam pengolahan data.
    2. ***Use Case Diagram***, diagram yang menunjukkan interaksi antara sistem dan lingkungannya/aktor.
    3. ***Sequence Diagram***, diagram yang menunjukkan interaksi antara aktor dengan sistem dan antar komponen sistem.
    4. ***Class Diagram***, diagram yang menunjukkan kelas-kelas objek dalam sistem dan asosiasi antara kelas-kelas tersebut.
    5. ***State Diagram***, yang menunjukkan bagaimana sistem bereaksi terhadap kejadian internal dan eksternal.

Referensi

- Sumber : Sommerville, I., (2011) 9th. *Software Engineering*. Pearson
- Romi Satria Wahono, *Systems Analysis and Design.*