**Pengertian Permodelan Perangkat Lunak Dan Study Kasus Toko Online**

****

Nama Kelompok :

1. Agus Deni Setiawan (1) XIRPL2
2. Hafizh Ulum (10) XIRPL2
3. MuhFaiq C.A (21) XIRPL2
4. Rafi Gusti Kurniawan (25) XIRPL2
5. Virgi Herwan Zakka S (34) XIRPL2

SMK Negeri 1 Bantul Tahun 2022

# Kata Pengantar

Puji syukur kami ucapkan kehadirat Allah SWT atas segala rahmat-Nya sehingga makalah ini dapat tersusun sampai dengan selesai. Tidak lupa kami mengucapkan terima kasih terhadap bantuan dari pihak yang telah berkontribusi dengan memberikan sumbangan baik pikiran maupun materinya. Penulis sangat berharap semoga makalah ini dapat menambah pengetahuan dan pengalaman bagi pembaca. Bahkan kami berharap lebih jauh lagi agar makalah ini bisa pembaca praktekkan dalam kehidupan sehari-hari. Bagi kami sebagai penyusun merasa bahwa masih banyak kekurangan dalam penyusunan makalah ini karena keterbatasan pengetahuan dan pengalaman Kami. Untuk itu kami sangat mengharapkan kritik dan saran yang membangun dari pembaca demi kesempurnaan makalah ini.

Bantul, 12 Mei 2022

Penyusun

# Daftar Isi

[Kata Pengantar 2](#_Toc103274250)

[Daftar Isi 3](#_Toc103274251)

[Daftar Tabel 4](#_Toc103274252)

[Daftar Gambar 5](#_Toc103274253)

[A. Pendahuluan 6](#_Toc103274254)

[1. Latar Belakang 6](#_Toc103274255)

[2. Rumusan Masalah 6](#_Toc103274256)

[3. Tujuan 6](#_Toc103274257)

[4. Manfaat 6](#_Toc103274258)

[B. Pembahasan 7](#_Toc103274259)

[1. Pengertian Permodelan Perangkat Lunak 7](#_Toc103274260)

[2. Tahapan Permodelan Perangkat Lunak 7](#_Toc103274261)

[a) Analisis 7](#_Toc103274262)

[b) Desain 8](#_Toc103274263)

[c) Implementasi 9](#_Toc103274264)

[d) Testing 10](#_Toc103274265)

[3. Desain System Desain Berorientasi Objek Mengunakan UML 10](#_Toc103274266)

[a) Definisi UML 10](#_Toc103274267)

[b) Macam-Macam UML 11](#_Toc103274268)

[c) Use Case 12](#_Toc103274269)

[d) Activity Diagram 14](#_Toc103274270)

[4. Desain System Terstuktur Dengan DFD 17](#_Toc103274271)

[a) Pengertian 18](#_Toc103274272)

[b) DFD Level 0 18](#_Toc103274273)

[1) Pengertian 18](#_Toc103274274)

[2) Tujuan/Manfaat 18](#_Toc103274275)

[3) Symbol dan Penjelasan 18](#_Toc103274276)

[4) Contoh 18](#_Toc103274277)

[c) DFD Level 1 18](#_Toc103274278)

[1) Pengertian 18](#_Toc103274279)

[2) Tujuan /Manfaat 18](#_Toc103274280)

[3) Symbol dan Penjelasan 18](#_Toc103274281)

[4) Contoh 18](#_Toc103274282)

# Daftar Tabel

# Daftar Gambar

# Pendahuluan

## Latar Belakang

## Rumusan Masalah

## Tujuan

## Manfaat

# Pembahasan

## Pengertian Permodelan Perangkat Lunak

Proses pemodelan perangkat lunak merupakan aktivitas yang saling terkait (koheren) untuk menspesifikasikan, merancang, implementasi kode program dan pengujian sistem perangkat lunak. Pada tingkat teknik, rekayasa perangkat lunak dimulai dengan serangkaian tugas pemodelan.

## Tahapan Permodelan Perangkat Lunak

### Analisis

Analisis Sistem (Systems Analysis)

Pada tahap ini, sistem akan dianalisis bagaimana akan dijalankan nantinya. Hasil analisis berupa kelebihan dan kekurangan sistem, fungsi sistem, hingga pembaharuan yang dapat diterapkan. Bagian ini termasuk dalam bagian perencanaan. Bagian lain yang termasuk dalam perencanaan ialah alokasi sumber daya, perencanaan kapasitas, penjadwalan proyek, estimasi biaya, dan penetapan. Dengan demikian, hasil dari tahap perencanaan ialah rencana proyek, jadwal, estimasi biaya, dan ketentuan. Idealnya manajer proyek dan pengembang dapat bekerja maksimal pada tahap ini.

Analisa sistem adalah tahap di mana dilakukan beberapa aktivitas berikut:

Melakukan studi literatur untuk menemukan suatu kasus yang bisa ditangani oleh sistem.

Brainstorming dalam tim pengembang mengenai kasus mana yang paling tepat dimodelkan dengan sistem.

Mengklasifikasikan masalah, peluang, dan solusi yang mungkin diterapkan untuk kasus tersebut.

Analisa kebutuhan pada sistem dan membuat Batasan-batasan sistem.

Mendefinisikan kebutuhan sistem.

### Desain

Pembuatan desain dan prototipe dalam SDLC dilakukan untuk melihat bentuk awal dari sebuah software aplikasi (Sumber: Pexels)

Fase desain dalam SDLC adalah tahapan di mana kamu membuat model cara kerja aplikasi software. Ada beberapa aspek yang diperhatikan dalam tahapan desain, antara lain:

Communications. Mendefinisikan metode atau cara aplikasi berkomunikasi dengan aset lainnya seperti server pusat atau aplikasi lainnya

Programming. Tidak hanya menentukan bahasa pemrograman tapi juga termasuk metode pemecahan masalah dan tugas-tugas yang ada dalam aplikasi.

Architecture. Menentukan bahasa pemrograman, praktik dalam industri, desain keseluruhan dan penggunaan template tertentu

User Interface. Mendefinisikan bagaimana cara pelanggan berinteraksi dengan software dan bagaimana software tersebut dapat merespon input yang ada

Platforms. Mendefinisikan platform di mana software akan dijalankan. Misalnya versi android, ios, linux atau game konsol.

Security. Mendefinisikan langkah-langkah untuk mengamankan aplikasi. misalnya membuat perlindungan kata sandi, enkripsi SSL traffic atau membuat penyimpanan kredensial pengguna yang aman.

### Implementasi

Setelah fase pengujian perangkat lunak selesai dan tidak ada bug yang tersisa pada sistem, maka tahap implementasi dapat dimulai. Tahap ini biasanya juga disebut sebagai tahap deployment. Tujuan dari tahap ini adalah untuk men-deploy perangkat lunak ke lingkungan produksi sehingga users dapat mulai menggunakannya.

Fase ini melibatkan penginstalan aktual dari sistem yang baru dikembangkan. Untuk proyek sederhana, contoh deployment seperti menerapkan kode ke server web. Sedangkan untuk proyek pengembagan software berskala besar, deployment akan melibatkan proses integrasi dengan banyak sistem berbeda.

Meskipun demikian, banyak perusahaan memilih agar produk akhir dapat pertama kali dirilis dalam segmen terbatas dan diuji di lingkungan bisnis (UAT-User Acceptance Testing) sebelum benar-benar dirilis ke pasar. Hal ini juga dilakukan untuk meminimalisir adanya masalah yang ditemukan oleh users setelah produk dirilis ke pasar.

### Testing

Tahapan SDLC ini akan melibatkan para software Quality Assurance (QA) untuk melakukan pengujian pada sistem dan menilai apakah software dapat bekerja sesuai dengan fungsionalitas yang diharapkan. Tim QA akan menguji semua area software untuk memastikan bahwa sistem terbebas dari cacat, error, ataupun bug. Jika ternyata masalah ditemukan di dalam software yang dikembangkan, maka tim QA akan menginformasikannya dengan tim pengembang agar perbaikan dapat segera dilakukan. Proses ini berlanjut hingga software benar-benar terbebas dari bug, bekerja stabil, dan berfungsi sesuai harapan.

## Desain System Desain Berorientasi Objek Mengunakan UML

### Definisi UML

UML (Unified Modelling Language) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. Awal mulanya, UML diciptakan oleh Object Management Group dengan versi awal 1.0 pada bulan Januari 1997

### Macam-Macam UML

Adapun jenis diagram UML dibagi bedasarkan 2 kategori yaitu Behavioral UML diagram dan Structural UML diagram. Sesuai dengan namanya, beberapa diagram UML tersebut berguna untuk menganalisis serta menggambarkan struktur suatu sistem atau proses, sedangkan lainnya menggambarkan perilaku sistem, aktor, dan komponen bangunannya.

Berikut jenis-jenisnya

Behavioral UML Diagram

Activity Diagram

Use Case Diagram

Interaction Overview Diagram

Timing Diagram

State Machine Diagram

Communication Diagram

Sequence Diagram

Structural UML Diagram

Class Diagram

Object Diagram

Component Diagram

Composite Structure Diagram

Deployment Diagram

Package Diagram

Profile Diagram

### Use Case

#### Pengertian

*use case diagram* sendiri adalah proses penggambaran yang dilakukan untuk menunjukkan hubungan antara pengguna dengan sistem yang dirancang. Hasil representasi dari skema tersebut dibuat secara sederhana dan bertujuan untuk memudahkan *user* dalam membaca informasi yang diberikan.

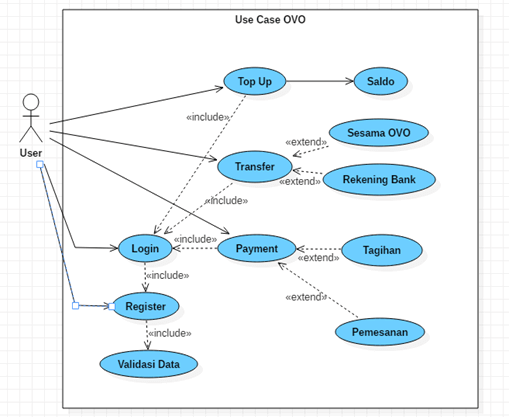
#### Manfaat

* Menggunakannya sebagai kebutuhan verifikasi.
* Menjadi gambaran *interface* dari sebuah sistem karena setiap sistem yang dibangun haruslah memiliki *interface*.
* Mengidentifikasi siapa saja orang yang dapat berinteraksi dengan sistem, serta apa yang dapat dilakukan oleh sistem.
* Memberikan kepastian mengenai kebutuhan  sistem, sehingga tidak membingungkan.
* Memudahkan proses komunikasi antara domain *expert* dan end *user*.

#### Symbol



#### Contoh UseCase



#### Scenario

Skenario Use Case mendeskripsikan aktor-aktor yang melakukan prosedur dalam sistem, serta menjelaskan respon yang ditanggapi oleh sistem tersebut terhadap prosedur yang dilakukan oleh aktor.

### Activity Diagram

#### Pengertian

*Activity diagram,* dalam bahasa Indonesia diagram aktivitas, yaitu diagram yang dapat memodelkan proses-proses yang terjadi pada sebuah sistem. Runtutan proses dari suatu sistem digambarkan secara vertikal. *Activity diagram* merupakan pengembangan dari *Use Case* yang memiliki alur aktivitas.

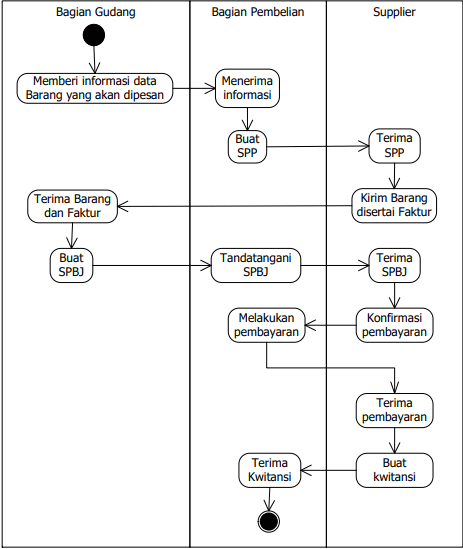
#### Manfaat

* Memperlihatkan urutan aktifitas proses pada sistem.
* Membantu memahami proses secara keseluruhan.
* **Activity Diagram** dibuat berdasarkan sebuah atau berapa use case.
* Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses.

#### Symbol dan Penjelasan



#### Contoh Activity Diagram



#### Scenario

## Desain System Terstuktur Dengan DFD

## Pengertian

## DFD Level 0

### Pengertian

### Tujuan/Manfaat

### Symbol dan Penjelasan

### Contoh

## DFD Level 1

### Pengertian

### Tujuan /Manfaat

### Symbol dan Penjelasan

### Contoh