

Halo.
Assalaamu'alaykum.

- Pembahasan :
Flowchart dan UML

- Team Presentation



Alamanda Ardana



Fatcurrahman



M. Syahdan Nur Karim

1

Flowchart

Gas Mulai?

● Definisi Flowchart

- Flowchart atau bagan alur 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 sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus. Selain, itu dengan menggunakan bagan alur, proses dari sebuah program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran. Penggunaan flowchart dalam dunia pemrograman juga merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan non-teknis.



Fungsi utama dari flowchart adalah memberi gambaran jalannya sebuah program dari satu proses ke proses lainnya sehingga, alur program menjadi mudah dipahami oleh semua orang. Selain itu, fungsi lain dari flowchart adalah untuk menyederhanakan rangkaian prosedur agar memudahkan pemahaman terhadap informasi tersebut.

Jenis Flowchart

1. Flowchart Dokumen

Flowchart dokumen berfungsi untuk menelusuri alur from dari satu bagian ke bagian yang lain, termasuk bagaimana laporan diproses.

2. Flowchart Program

Flowchart ini menggambarkan secara rinci prosedur dari proses program.

3. Flowchart Proses

Flowchart proses adalah cara penggambaran rekayasa industrial dengan cara merinci dan menganalisis langkah-langkah selanjutnya dalam suatu prosedur atau sistem.

4. Flowchart Sistem













Flowchart sistem ialah flowchart yang menampilkan tahapan atau proses kerja yang sedang berlangsung di dalam sistem secara menyeluruh.

5. Flowchart Skematik

Flowchart skematik ialah menggunakan gambar-gambar pada komputer serta peralatan lainnya untuk mempermudah dalam pembacaan flowchart untuk orang awam

Simbol Flowchart

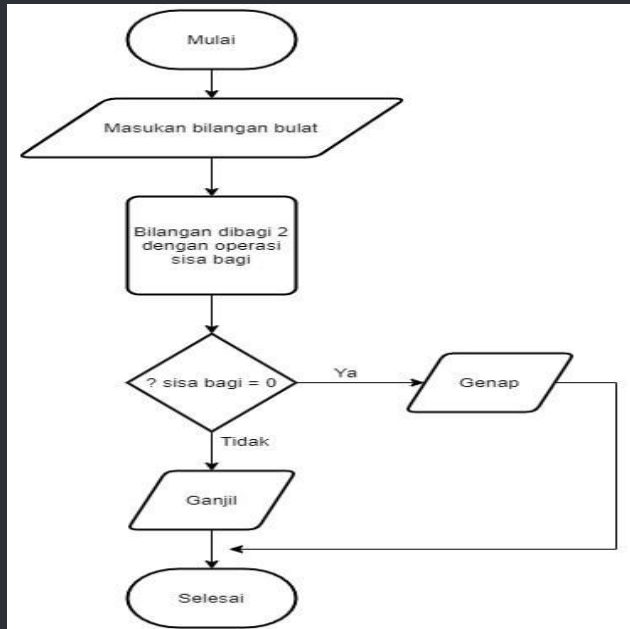
Pada dasarnya, simbol-simbol dalam flowchart memiliki arti yang berbeda-beda. Berikut adalah simbol-simbol yang sering digunakan dalam proses pembuatan flowchart.

| | | | |
|---|---|--|--|
|  | Flow Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line. |  | Input/output Simbol yang menyatakan proses input atau output tanpa tergantung peralatan. |
|  | On-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama. |  | Manual Operation Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer. |
|  | Off-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda. |  | Document Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak. |
|  | Terminator Simbol yang menyatakan awal atau akhir suatu program. |  | Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur. |
|  | Process Simbol yang menyatakan suatu proses yang dilakukan komputer. |  | Display Simbol yang menyatakan peralatan output yang digunakan. |
|  | Decision Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak. |  | Preparation Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal. |

Simbol-simbol di atas memiliki jenis dan fungsi yang berbeda-beda. Ada yang berfungsi untuk menghubungkan satu simbol dengan simbol lainnya seperti simbol flow, on-page dan off-page reference. Selain itu, ada juga simbol yang berfungsi untuk menunjukan suatu proses yang sedang berjalan, dan yang terakhir terdapat simbol yang berfungsi untuk memasukan input dan menampilkan output.

Contoh Flowchart

Untuk membuat sebuah flowchart sederhana, kita diharuskan untuk mengetahui setiap simbol dan juga fungsinya. Nah, di bawah ini adalah sebuah contoh flowchart sederhana untuk menentukan apakah bilangan yang dimasukkan ganjil atau genap. Berikut adalah contohnya:



Pembahasan Gambar di Samping

- Pertama, pengguna menginput data yang berupa nilai dari bilangan bulat.
- Kemudian nilai yang dimasukan diproses dengan cara dibagi dengan angka 2.
- Jika sisa bagi sama dengan 0 berarti bilangan yang dimasukan adalah bilangan genap.
- Jika sisa bagi tidak sama dengan 0 berarti bilangan yang dimasukan adalah bilangan ganjil.
- Selesai.

2

UML

Gas Mulai?

● Definisi UML

- UML merupakan akronim dari *Unified Modeling Language*, sebuah standar untuk memodelkan bahasa yang membantu pengembang untuk menspesifikasikan, memvisualisasikan, membuat konstruksi dan dokumentasi dari sebuah sistem.
- Tujuan utama dari UML adalah memvisualisasikan bagaimana sebuah sistem didesain, sehingga UML akan membantu Anda memberikan gambaran sebuah sistem dalam bentuk visual yang lebih mudah untuk dibayangkan.
- UML bukanlah bahasa pemrograman, melainkan sebuah bahasa visual.

● Sejarah Singkat UML

- UML dimulai secara resmi pada Oktober 1994, ketika James Rumbaugh menggabungkan kekuatan dengan Grady Booch. Mereka berdua lalu bekerja bersama di Relational Software Cooperation. Proyek ini memfokuskan pada penyatuan metode Booch dan Rumbaugh (OMT). Pada bulan October 1995, UML merilis versi 0.8 dan pada waktu yang sama juga, Ivan Jacobson bergabung dengan Relational. Cakupan dari UML pun semakin meluas. Kemudian dibangunlah persatuan untuk UML dengan beberapa organisasi yang akan menyumbangkan sumber dayanya untuk bekerja, mengembangkan, dan melengkapi UML.
- Banyak *partner* yang berkontribusi pada UML 1.0, diantaranya Digital Equipment Corporation, Hawlett-Packard, I-Logix, IBM, ICON Computing, MCI systemhouse, Microsoft, Oracle, Relation, Texas Instruments dan Unisys. Dari kolaborasi ini, dihasilkan UML 1.0 yang merupakan bahasa pemodelan yang ditetapkan secara baik, *expressive*, kuat, dan cocok untuk lingkungan masalah yang luas.
- Pada Januari 1997, UML dijadikan sebagai standar bahasa pemodelan.

● Tujuan dan Fungsi UML

1. Dapat memberikan bahasa pemodelan visual atau gambar kepada para pengguna dari berbagai macam pemrograman maupun proses umum rekayasa.
2. Menyatukan informasi-informasi terbaik yang ada dalam pemodelan.
3. Memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem.
4. Tidak hanya menggambarkan model sistem *software* saja, namun dapat memodelkan sistem berorientasi objek.
5. Mempermudah pengguna untuk membaca suatu sistem.
6. Berguna sebagai *blueprint*, jelas ini nantinya menjelaskan informasi yang lebih detail dalam perancangan berupa *coding* suatu program.

● Tipe Diagram UML

1. Structure Diagrams:

- a. Class Diagram
- b. Object Diagram
- c. Component Diagram
- d. Composite Structure Diagram
- e. Package Diagram
- f. Deployment Diagram

2. Behaviour Diagrams:

- a. Use Case Diagram
- b. Activity Diagram
- c. State Machine Diagram

3. Interaction Diagrams :

- a. Sequence Diagram
- b. Communication Diagram
- c. Timing Diagram
- d. Interaction Overview Diagram

● Tipe Diagram UML (Structure Diagrams)

- A. **Class Diagram**, adalah jenis diagram pada UML yang digunakan untuk menggambarkan paket-paket atau kelas-kelas yang ada pada sistem dan akan digunakan pada suatu sistem.
- B. **Object Diagram**, adalah diagram yang menunjukkan gambaran dari struktur model yang ada di dalam sistem untuk jangka waktu tertentu. Diagram ini akan bergantung pada class diagram yang dibuat untuk suatu sistem.
- C. **Component Diagram**, merupakan suatu diagram yang dibuat untuk menunjukkan struktur dan ketergantungan antara komponen yang ada di dalam suatu sistem. Komponen tersebut adalah hal fisik dari program yang dimodelkan dan akan terlihat saat sistem dijalankan.
- D. **Composite Structure Diagram**, adalah diagram yang menjelaskan struktur internal dari *component*, *class*, dan *use case* termasuk klasifikasi hubungan yang dimiliki dalam sebuah program.
- E. **Package Diagram**, merupakan diagram yang memiliki fungsi untuk mengumpulkan kelas dan memperlihatkan penyusunan dari elemen model dalam sistem.
- F. **Deployment Diagram**, adalah tata letak program secara fisik atau menampilkan bagian aplikasi yang ada pada perangkat keras dan digunakan untuk menerapkan sistem.

● Tipe Diagram UML (Behaviour Diagrams)

- A. **Use Case Diagram**, adalah urutan interaksi yang memiliki keterkaitan antara sistem dan aktor. *Use case* diagram dijalankan dengan cara menggambarkan tipe interaksi yang terjadi diantara *user* yang terlibat di dalam sistem.
- B. **Activity Diagram**, merupakan diagram yang menunjukkan model dari metode yang ada di dalam sistem. Diagram dapat menggambarkan perancangan sistem, proses awal sistem berjalan, keputusan yang dapat terjadi dalam sistem, dan akhir dari sistem.
- C. **State Machine Diagram**, merupakan diagram dalam UML yang memberikan gambaran tentang perubahan dan transisi kondisi dari satu status ke status lainnya dari objek yang terdapat pada program yang terjadi karena transisi dari satu status ke status lainnya.

● Tipe Diagram UML (Interaction Diagrams)

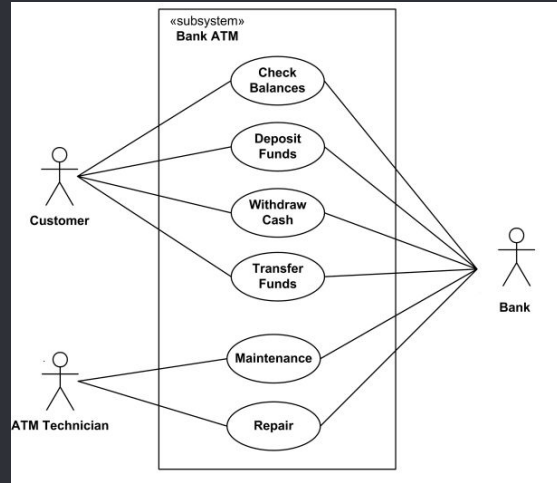
- A. **Sequence Diagram**, adalah diagram UML yang menggambarkan hubungan di objek yang didasarkan pada urutan waktu.
- B. **Communication Diagram**, merupakan jenis diagram yang terdapat pada UML untuk menjelaskan proses terjadinya aktivitas dan interaksi pada suatu sistem.
- C. **Timing Diagram**, merupakan jenis diagram yang memiliki fokus utama pada waktu dan menjadi bentuk lain dari interaksi diagram.
- D. **Interaction Overview Diagram**, merupakan salah satu jenis diagram yang berfungsi untuk menunjukkan visualisasi dari hubungan kerjasama antara activity dengan sequence diagram.

● Tipe Diagram UML yang Sering Digunakan

1. Use Case Diagram

(Yang Tadi) Use Case Diagram, adalah urutan interaksi yang memiliki keterkaitan antara sistem dan aktor. *Use case* diagram dijalankan dengan cara menggambarkan tipe interaksi yang terjadi di antara *user* yang terlibat di dalam sistem.

- Gambar di samping adalah contoh Use Case Diagram, dengan studi kasus berupa Sistem Pengoperasian ATM.



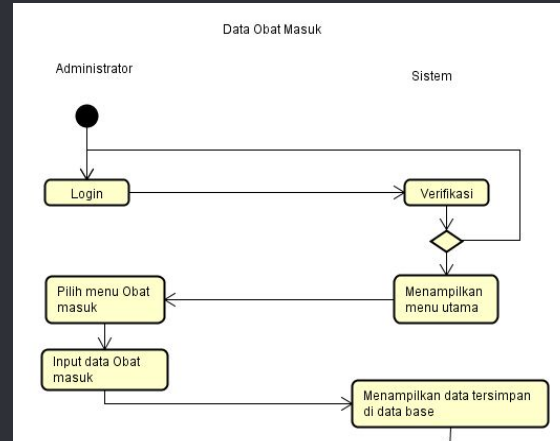
(Sumber : dicoding.com)

● Tipe Diagram UML yang Sering Digunakan

2. Activity Diagram

(Yang Tadi) Activity Diagram, merupakan diagram yang menunjukkan model dari metode yang ada di dalam sistem. Diagram dapat menggambarkan perancangan sistem, proses awal sistem berjalan, keputusan yang dapat terjadi dalam sistem, dan akhir dari sistem.

- Gambar di samping adalah contoh Activity Diagram, dengan studi kasus berupa Proses Memasukkan Data Obat.



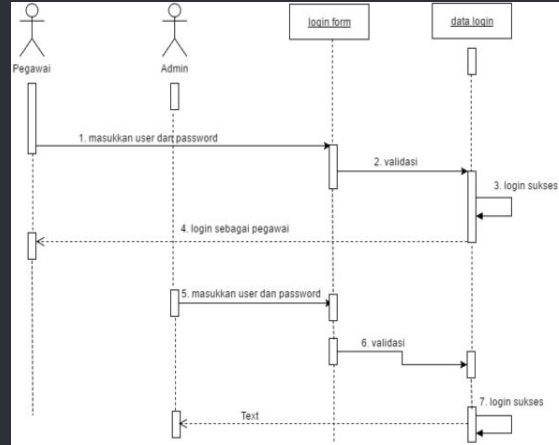
(Sumber : badoystudio.com)

● Tipe Diagram UML yang Sering Digunakan

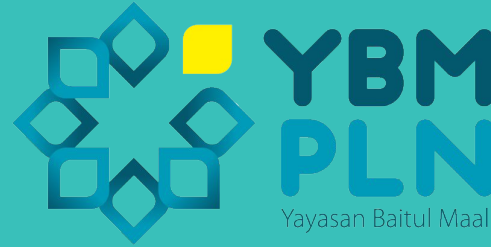
3. Sequence Diagram

(Yang Tadi) Sequence Diagram, adalah diagram UML yang menggambarkan hubungan di objek yang didasarkan pada urutan waktu.

- Gambar di samping adalah contoh Sequence Diagram, dengan studi kasus berupa Proses Login Pegawai dan Hubungannya dengan Admin.



(Sumber : badoystudio.com)



Sekian dan Terima kasih.
Wassalaamu'alaykum.

REFERENSI ISI

- <https://www.dicoding.com/blog/flowchart-adalah/>
- <https://www.uml.org/what-is-uml.htm>
- <https://www.dicoding.com/blog/apa-itu-uml/>
- [https://idcloudhost.com/panduan/mengenal-uml-pengertian-dan-sejarah
nya/](https://idcloudhost.com/panduan/mengenal-uml-pengertian-dan-sejarah-nya/)
- [https://itkampus.com/pengertian-uml/#structure diagram](https://itkampus.com/pengertian-uml/#structure_diagram)
- [https://badoustudio.com/uml/#3 Jenis UML dan Contohnya](https://badoustudio.com/uml/#3_Jenis_UML_dan_Contohnya)

● REFERENSI GAMBAR (Di Slide Salam dan Team Presentation)

- <https://petik.or.id/>
- <https://sport.detik.com/sepakbola/liga-inggris/d-6111787/mengapa-cristiano-ronaldo-bertahan-di-mu>
- https://en.wikipedia.org/wiki/Lionel_Messi
- <https://sport.detik.com/sepakbola/bola-dunia/d-6095310/psg-jual-neymar-siapa-mau>

- CREDITS

- Special thanks to all the people who made and released these awesome resources for free presentation template by [SlidesCarnival](#)