Nama : Ade Hikmat Pauji Ridwan

Kelas : TIF 222K

NPM: 22552011130

**UML (Unified Modeling Language)** adalah bahasa pemodelan visual yang digunakan untuk merancang dan memodelkan sistem perangkat lunak.

**Fungsi UML:**

1. **Memodelkan Sistem Perangkat Lunak**
2. **Meningkatkan Komunikasi dan Kolaborasi**
3. **Meningkatkan Kualitas Perangkat Lunak**
4. **Meningkatkan Efisiensi Pengembangan Perangkat Lunak**
5. **Memastikan Kepuasan Klien**

**Diagram UML** adalah notasi/simbol yang digunakan untuk mempresentasikan aspek tertentu dari sistem perangkat lunak. Berikut beberapa jenis UML:

1. **Use Case Diagram**

Use case diagram menggambarkan hubungan interaksi antara sistem dan aktor. Use case menjelaskan tipe interaksi antara pengguna sistem dan sistem itu sendiri.

**Fungsi Use Case Diagram:**

* + Memperlihatkan proses aktivitas secara urut dalam sistem.
  + Menggambarkan proses bisnis, bahkan menampilkan urutan aktivitas dalam sebuah proses.
  + Berfungsi sebagai jembatan antara pembuat dan konsumen untuk mendeskripsikan sebuah sistem.

**Manfaat Use Case Diagram:**

* + Digunakan sebagai verifikasi kebutuhan.
  + Menjadi gambaran antarmuka sistem karena setiap sistem harus memiliki antarmuka.
  + Mengidentifikasi siapa yang dapat berinteraksi dengan sistem dan apa yang dapat dilakukan oleh sistem.
  + Memberikan kejelasan mengenai kebutuhan sistem, sehingga tidak membingungkan.
  + Memudahkan komunikasi antara ahli domain dan pengguna akhir.

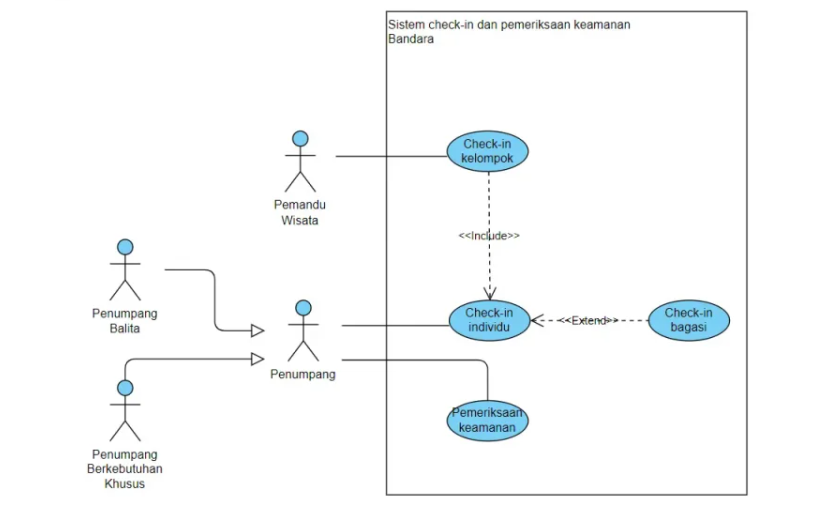
**Komponen Use Case Diagram:**

* **System:** Digambarkan dalam bentuk persegi untuk membatasi use case dari interaksi luar sistem. Umumnya diberi label yang sesuai.
* **Actor:** Menjelaskan siapa yang berinteraksi dengan sistem. Actor memberikan dan menerima informasi dari sistem tanpa memberikan kontrol terhadap sistem.
* **Use Case:** Komponen fungsional dalam sistem yang menjelaskan alur sistem yang akan dibuat.
* **Relationship:** hubungan antara actor dengan use case atau use case dengan use case.
  + - **Association:** Menunjukkan relasi antara aktor dan use case tanpa adanya interaksi yang spesifik. Misalnya, dalam konteks e-commerce, aktor "pelanggan" berasosiasi dengan use case "pencarian produk". Dengan demikian, pelanggan dapat mencari produk yang tersedia di platform e-commerce tersebut.
    - **Generalization:** Suatu aktor atau use case merupakan spesialisasi atau bentuk khusus dari aktor atau use case lain. Sebagai contoh, aktor "admin" adalah generalisasi dari "pengguna", di mana admin memiliki lebih banyak hak akses di dalam sistem.
    - **Include:** Sebuah use case merupakan bagian dari use case lainnya yang tidak dapat berdiri sendiri. Sebagai contoh, fitur "pilih kursi" merupakan bagian dari use case "pemesanan tiket kereta" saat pelanggan menggunakan aplikasi pemesanan tiket kereta.
    - **Extend:** Sebuah use case dapat diperluas ke satu atau beberapa use case lain untuk menambah fungsionalitasnya. Misalnya, ketika pelanggan menggunakan fitur "pengembalian barang", terdapat fitur tambahan yaitu "cek status pengembalian" di dalam website toko online.

**Komponen Use Case Diagram:**



**Contoh Use Case Diagram:**

****

1. **Use Case Secenario**

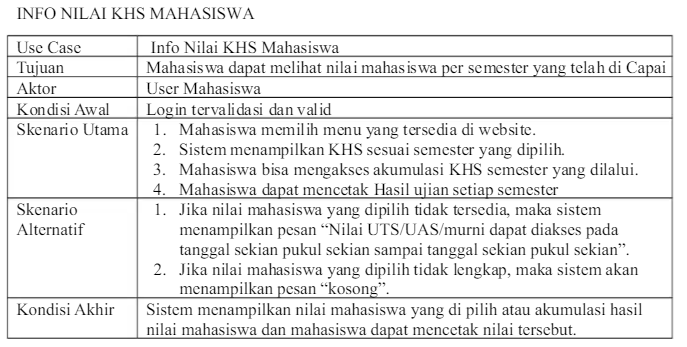
Use case scenario adalah deskripsi naratif tentang cara pengguna akan berinteraksi dengan sebuah sistem atau produk untuk mencapai tujuan tertentu. Secenario use case menjelaskan langkah-langkah spesifik yang diambil oleh pengguna atau sistem untuk mencapai suatu hasil atau memecahkan suatu masalah. Ini sering digunakan dalam pengembangan perangkat lunak untuk merinci bagaimana sebuah fitur atau fungsionalitas akan digunakan dalam situasi dunia nyata. Dengan menggunakan scenerio use case, pengembang dapat memahami secara lebih baik kebutuhan pengguna dan mendesain solusi yang sesuai.

“*Deskripsi naratif adalah penjelasan atau cerita yang disusun secara kronologis atau logis untuk menggambarkan suatu kejadian, situasi, atau konsep. Ini melibatkan penggunaan bahasa yang deskriptif untuk menyampaikan detail tentang subjek yang dibahas.*”

**Fungsi Use Case Secenario:**

* + **Merinci Fungsionalitas:** Use case scenario membantu dalam merinci fungsionalitas sistem atau aplikasi dengan menyajikan interaksi konkret antara pengguna dan sistem. Ini membantu pengembang untuk memahami secara rinci apa yang diharapkan dari sistem dan bagaimana itu harus berperilaku dalam berbagai situasi.
  + **Menjelaskan Interaksi Pengguna:** Use case scenario menggambarkan bagaimana pengguna akan berinteraksi dengan sistem atau aplikasi dalam kehidupan nyata. Ini membantu pengembang untuk memahami kebutuhan pengguna dengan lebih baik dan merancang antarmuka yang intuitif dan mudah digunakan.
  + **Mengidentifikasi Kasus Penggunaan:** Dengan merinci berbagai skenario penggunaan, use case scenario membantu dalam mengidentifikasi dan memahami kasus penggunaan yang berbeda untuk sistem atau aplikasi. Ini memungkinkan pengembang untuk fokus pada fitur dan fungsionalitas yang paling penting dan relevan untuk pengguna.
  + **Membantu dalam Pengujian:** Use case scenario menyediakan dasar yang baik untuk pengujian perangkat lunak. Dengan memiliki skenario yang jelas tentang bagaimana pengguna akan menggunakan sistem, pengembang dapat merencanakan dan melaksanakan pengujian yang komprehensif untuk memastikan bahwa sistem berperilaku sesuai dengan harapan.
  + **Komunikasi yang Efektif:** Use case scenario juga berfungsi sebagai alat komunikasi yang efektif antara pengembang, pemangku kepentingan, dan tim proyek lainnya. Mereka membantu dalam menyampaikan kebutuhan dan harapan pengguna dengan jelas dan memfasilitasi diskusi yang konstruktif tentang desain dan pengembangan sistem.

Contoh Use Case Secenario:



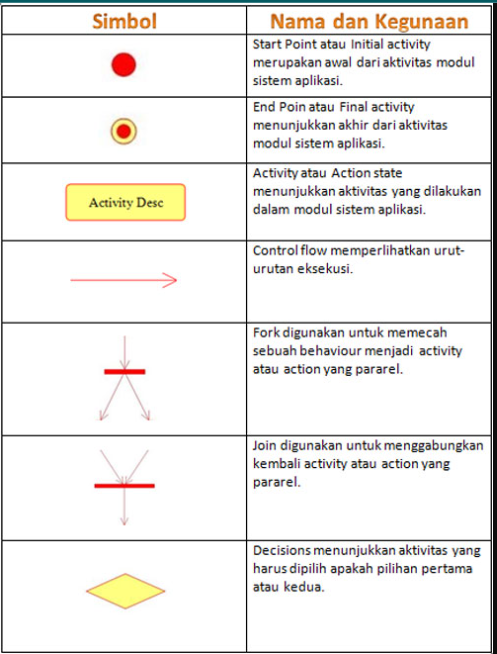
1. **Activity diagram**

Activity diagram, juga dikenal sebagai diagram aktivitas, adalah alat visual yang digunakan untuk memodelkan urutan aktivitas dalam suatu proses. Menurut Rosa A.S, "Diagram aktivitas tidak menjelaskan kelakuan aktor." Ini berarti bahwa activity diagram digunakan khusus untuk menggambarkan alur kerja atau aktivitas sistem.

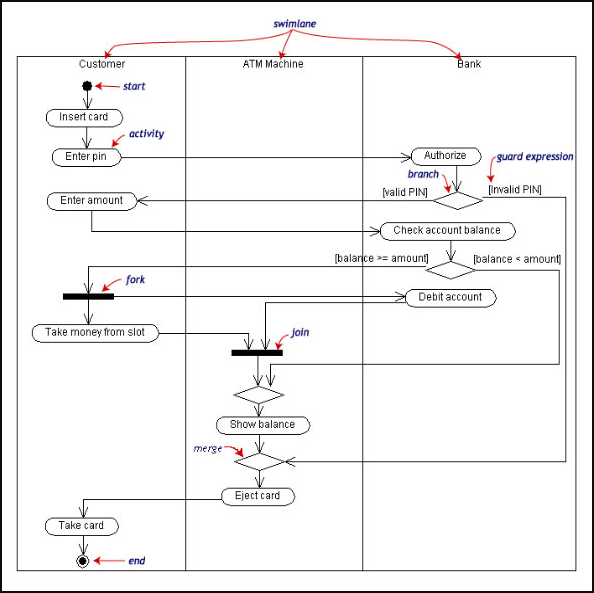
Manfaat Activity Diagram:

* + Menjelaskan Urutan Aktivitas dalam Suatu Proses: Activity diagram membantu dalam menjelaskan secara visual urutan langkah-langkah yang harus dilakukan dalam suatu proses. Ini membantu pengguna atau pengembang memahami dengan jelas bagaimana proses berjalan.
  + Modeling Proses Bisnis: Dalam dunia bisnis, activity diagram sering digunakan untuk memodelkan proses bisnis. Ini membantu dalam memvisualisasikan dan memahami alur kerja bisnis secara keseluruhan.
  + Memudahkan Pemahaman Proses: Activity diagram menyajikan proses secara visual, sehingga lebih mudah dipahami oleh semua pihak terkait. Ini memfasilitasi komunikasi antara tim pengembangan, pemangku kepentingan, dan pengguna.
  + Metode Perancangan yang Terstruktur: Activity diagram merupakan metode perancangan yang terstruktur, mirip dengan flowchart dan Data Flow Diagram (DFD). Hal ini memudahkan dalam merancang dan mengembangkan sistem dengan pendekatan yang sistematis.
  + Mengidentifikasi Aktivitas Pengguna berdasarkan Use Case: Activity diagram dapat digunakan untuk mengidentifikasi aktivitas pengguna berdasarkan use case atau diagram lainnya yang telah dibuat sebelumnya. Ini membantu dalam memahami bagaimana pengguna berinteraksi dengan sistem dan melengkapi analisis use case.

**Komponen Activity Diagram:**



**Contoh Activiity Diagram:**



1. Sequence Diagram

Class Diagram

Daftar Pusaka:

UML : <https://www.sekawanmedia.co.id/blog/apa-itu-uml/>

Use Case Diagram :

* <https://www.dicoding.com/blog/contoh-use-case-diagram/>
* <https://www.niagahoster.co.id/blog/use-case-diagram-adalah/>

Use Case Secenario :

* <https://www.scribd.com/document/432374314/Skenario-Web-Student-BSI>
* <https://chat.openai.com/c/9d7a556f-e50b-455b-aa71-3474231bf69c>

Activity Diagram :

* <https://www.dicoding.com/blog/apa-itu-activity-diagram/>
* <https://andydharmalau.com/diagram-activity/>

Sequence Diagram:

* <https://www.dicoding.com/blog/apa-itu-sequence-diagram/>