

Nama : Theresia Agatha Napitupulu  
NIM : 12030123120010  
Mata Kuliah : Analisis dan Desain Sistem (D)

# PLANTUML

## A. Pengertian

PlantUML adalah perangkat lunak sumber terbuka yang memungkinkan pembuatan diagram UML (Unified Modeling Language) secara otomatis menggunakan kode teks. Dengan PlantUML, pengembang atau tim pengembang dapat membuat berbagai diagram sistem yang diperlukan dalam proses analisis dan desain. Alat ini sangat bermanfaat karena memudahkan pembuatan dan pembaruan diagram yang langsung terintegrasi dengan dokumentasi teknis.

PlantUML mendukung berbagai jenis diagram, seperti diagram kelas, use case, urutan, aktivitas, dan lainnya. Keunggulan utama PlantUML adalah kemampuannya untuk menghasilkan diagram hanya dengan menulis kode teks, yang menjadikannya fleksibel dan mudah dikelola dalam repositori kode sumber.

## B. Jenis-Jenis Diagram dalam PlantUML

Berikut adalah penjelasan lebih lanjut tentang jenis diagram yang dapat dibuat menggunakan PlantUML:

### 1. Diagram Use Case (Use Case Diagram)

- **Fungsi:** Digunakan untuk memodelkan interaksi antara aktor (pengguna atau sistem lain) dan sistem. Diagram ini menggambarkan fungsionalitas yang ditawarkan oleh sistem.
- **Contoh:** Pada sistem pemesanan makanan online, aktor seperti pelanggan dan admin berinteraksi dengan sistem melalui aksi-aksi seperti registrasi, login, memilih menu, dan melakukan pembayaran.

### 2. Diagram Kelas (Class Diagram)

- **Fungsi:** Digunakan untuk menggambarkan struktur kelas dalam sistem dan hubungan antar kelas. Diagram kelas menunjukkan atribut (data) dan metode (fungsi) yang ada pada suatu kelas, serta hubungan antar kelas seperti pewarisan dan asosiasi.
- **Contoh:** Dalam sistem pemesanan makanan, diagram kelas bisa menggambarkan kelas **User**, **Menu**, **Order**, dan **Payment**, serta hubungan antar kelas tersebut.

### 3. Diagram Urutan (Sequence Diagram)

- **Fungsi:** Menunjukkan bagaimana objek berinteraksi satu sama lain dalam urutan waktu tertentu. Diagram ini sering digunakan untuk menggambarkan alur proses dalam suatu skenario.

- **Contoh:** Diagram urutan untuk proses pemesanan makanan, mulai dari login, memilih menu, hingga pembayaran.
- 4. **Diagram Aktivitas (Activity Diagram)**
  - **Fungsi:** Menggambarkan alur kerja atau aktivitas dalam sistem. Diagram aktivitas sering digunakan untuk memodelkan proses bisnis atau alur dalam suatu sistem.
  - **Contoh:** Pada sistem pemesanan makanan, diagram aktivitas bisa menggambarkan proses dari login, pemilihan menu, checkout, hingga pembayaran.
- 5. **Diagram Komponen (Component Diagram)**
  - **Fungsi:** Menunjukkan komponen-komponen sistem dan hubungan antar komponen. Berguna untuk memodelkan struktur sistem yang lebih besar dan lebih kompleks.
  - **Contoh:** Diagram komponen dapat digunakan untuk menunjukkan bagaimana komponen aplikasi, server, dan database saling berinteraksi dalam sistem pemesanan makanan.
- 6. **Diagram Paket (Package Diagram)**
  - **Fungsi:** Memodelkan pengelompokan kelas-kelas atau komponen-komponen dalam paket-paket yang lebih besar. Biasanya digunakan untuk menggambarkan struktur hierarkis atau pengelompokan modul dalam sistem.
  - **Contoh:** Mengelompokkan modul-modul seperti **Pengguna**, **Menu**, dan **Pembayaran** dalam paket-paket tertentu di sistem pemesanan makanan.

### C. Kelebihan PlantUML

- **Integrasi dengan Dokumen Teks:** PlantUML memungkinkan pembuatan diagram secara otomatis hanya dengan menulis kode teks. Hal ini memudahkan integrasi diagram dalam dokumentasi berbasis teks seperti Markdown atau dokumen LaTeX.
- **Kemudahan Pembaruan:** Setiap kali ada perubahan pada sistem, diagram dapat diperbarui dengan mudah hanya dengan mengedit kode teksnya. Ini sangat berguna untuk tim yang bekerja dalam proyek perangkat lunak yang terus berkembang.
- **Berguna untuk Kolaborasi:** Kode PlantUML dapat disimpan di repositori (misalnya GitHub), memudahkan kolaborasi antar pengembang dalam tim.
- **Berbagai Jenis Diagram:** PlantUML mendukung banyak jenis diagram yang berbeda, memungkinkan pengembangan sistem secara menyeluruh, dari pemodelan konsep hingga implementasi teknis.

### D. Mermaid: Alternatif untuk PlantUML

**Mermaid** adalah alat lain yang digunakan untuk membuat diagram berbasis teks, sangat mirip dengan PlantUML, tetapi lebih sering digunakan dalam dokumen web atau markdown, terutama pada platform seperti GitHub. Mermaid menawarkan cara mudah untuk membuat diagram alur, diagram urutan, dan diagram lainnya.

Perbedaan utama antara **PlantUML** dan **Mermaid** adalah:

- **PlantUML** lebih banyak digunakan di dalam lingkungan pengembangan perangkat lunak dan mendukung lebih banyak jenis diagram UML, sedangkan **Mermaid** lebih sering digunakan untuk menggambarkan diagram alur atau struktur yang lebih sederhana dalam dokumentasi berbasis web.
- **PlantUML** menawarkan lebih banyak kontrol dan kustomisasi dalam desain diagram, sedangkan **Mermaid** lebih fokus pada kemudahan penggunaan dalam konteks berbasis web.

## E. Studi Kasus dan Diagram

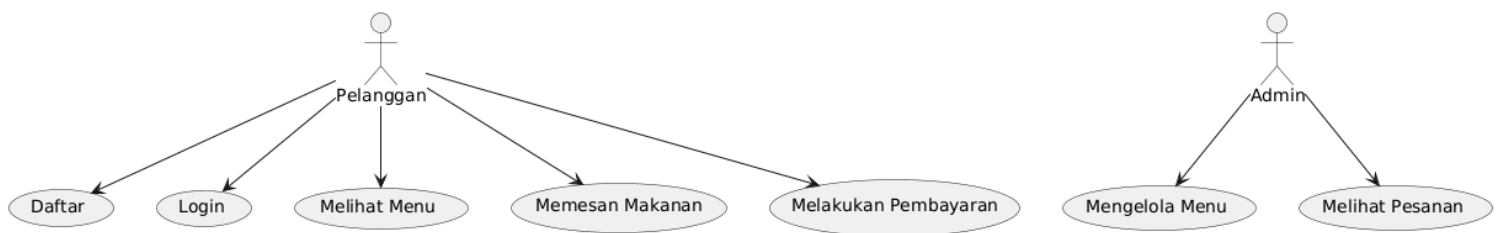
### Studi Kasus: Sistem Pemesanan Makanan Online

Pada sistem ini, terdapat beberapa aktor utama:

- **Pelanggan:** Berinteraksi dengan sistem untuk mendaftar, login, memilih menu, memesan makanan, dan melakukan pembayaran.
- **Admin:** Mengelola menu dan memantau pesanan yang masuk.

#### 1. Use Case Diagram

Diagram use case menggambarkan interaksi antara aktor dan sistem. Dalam kasus ini, kita memiliki dua aktor utama: pelanggan dan admin. Pelanggan bisa mendaftar, login, melihat menu, memesan makanan, dan membayar. Admin bisa mengelola menu dan melihat pesanan.

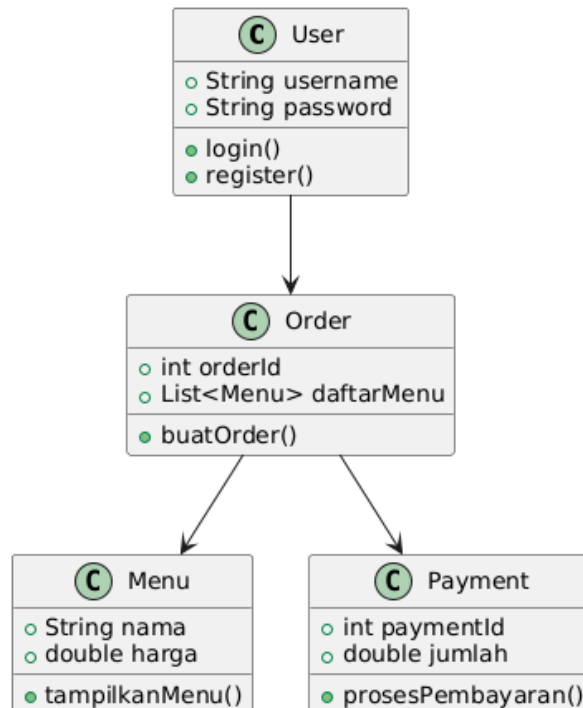


#### 2. Diagram Kelas

Diagram **kelas** menggambarkan struktur kelas dalam sistem. Kelas-kelas yang terlibat dalam sistem ini antara lain:

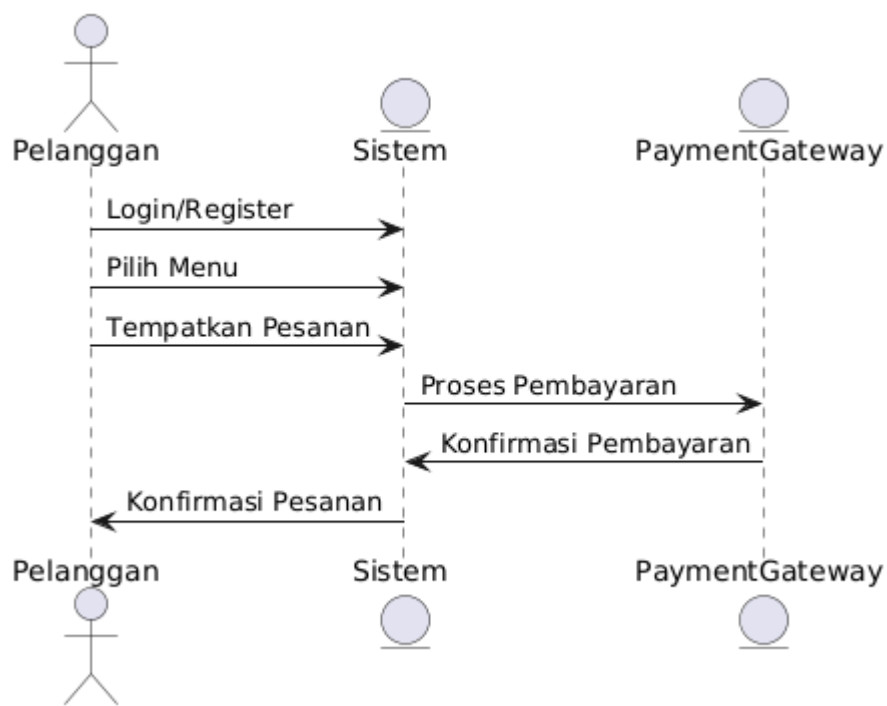
- **User:** Menangani pengguna (Pelanggan) dengan atribut seperti **username** dan **password**.

- **Menu:** Mengelola menu makanan dengan atribut seperti **nama** dan **harga**.
- **Order:** Mengelola pemesanan dengan atribut seperti **orderId** dan **daftarMenu**.
- **Payment:** Menangani pembayaran dengan atribut seperti **paymentId** dan **jumlah**.



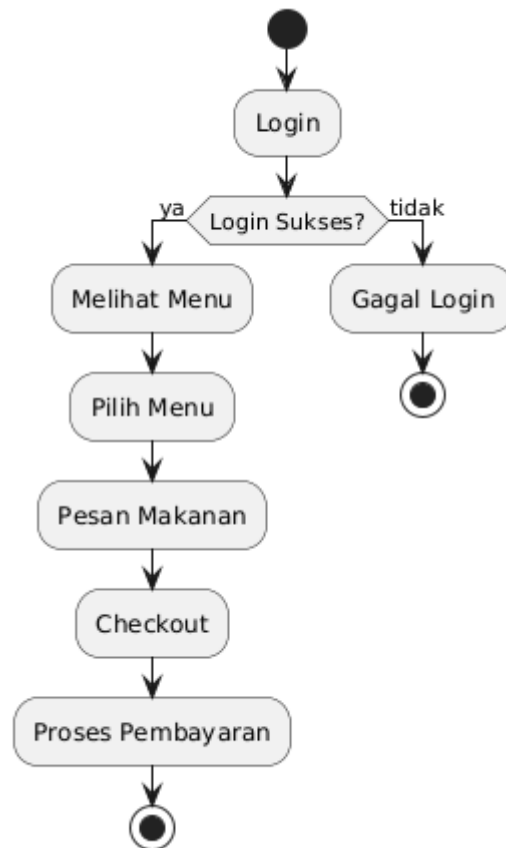
### 3. Diagram Urutan

Diagram urutan menggambarkan urutan interaksi antar objek dalam sistem berdasarkan waktu. Pada diagram ini, kita akan menunjukkan proses dari pelanggan yang login, memilih menu, melakukan pemesanan, hingga pembayaran.



#### 4. Diagram Aktivitas

Diagram aktivitas menggambarkan alur kegiatan yang dilakukan oleh pelanggan dari mulai login, memilih menu, hingga selesai melakukan pembayaran.



#### F. Kesimpulan

PlantUML dan Mermaid adalah alat yang sangat berguna dalam analisis dan desain sistem. Dengan menggunakan kode teks, pengembang dapat menggambarkan diagram UML yang kompleks dengan cara yang efisien, terutama dalam pengembangan perangkat lunak yang melibatkan tim. Keduanya memungkinkan untuk membuat diagram yang dapat dengan mudah disertakan dalam dokumentasi dan diperbarui sesuai kebutuhan tanpa perlu menggambar ulang. PlantUML lebih banyak digunakan untuk proyek perangkat lunak besar, sementara Mermaid sering digunakan untuk proyek yang lebih sederhana atau dokumentasi berbasis web.