



# PANDUAN LENGKAP UML DIAGRAM

## Bekal Presentasi SIM4LON

### 1 USE CASE DIAGRAM





#### Definisi

Diagram yang menunjukkan **SIAPA** (aktor) bisa melakukan **APA** (use case) di sistem.







#### Fungsi

- Menggambarkan kebutuhan fungsional dari sudut pandang pengguna
- Menunjukkan batasan sistem (system boundary)
- Dokumentasi fitur-fitur utama

#### Komponen

| Komponen        | Simbol   | Fungsi                           |
|-----------------|--|----------------------------------|
| Actor           |  (stick figure) | Pengguna atau sistem eksternal   |
| Use Case        |  (oval)         | Fungsi/fitur yang bisa dilakukan |
| System Boundary |  (kotak)        | Batasan sistem                   |
| Association     |  (garis)        | Aktor terhubung ke use case      |

#### Relationship / Relasi

| Relasi         | Simbol   | Arti                                   | Contoh  |
|----------------|--|--|---|
| Association    |                 | Aktor bisa melakukan use case          | Admin  Kelola User |
| Include        | <br><<include>> | Use case A <b>SELALU</b> memanggil B   | Kelola Stok --include--> Catat Penerimaan   |
| Extend         | <br><<extend>>  | Use case B <b>OPSIONAL</b> dipanggil A | Update Status <--extend-- Assign Driver   |
| Generalization |                 | Actor inheritance                      | Operator  Admin    |

#### Contoh SIM4LON

- 3 Aktor: Admin, Operator, Pangkalan
- 23 Use Cases total
- Include: Kelola Stok → Catat Penerimaan

- Extend: Update Status ← Assign Driver

## 2 CLASS DIAGRAM

### Definisi

Diagram yang menunjukkan **struktur statis** sistem: class, atribut, method, dan relasi antar class.

### Fungsi

- Blueprint objek dalam sistem (OOP)
- Menggambarkan inheritance dan encapsulation
- Dasar untuk implementasi kode

### Komponen

| Komponen   | Bagian        | Contoh                           |
|------------|---------------|----------------------------------|
| Nama Class | Bagian atas   | orders                           |
| Atribut    | Bagian tengah | +code: string, -password: string |
| Method     | Bagian bawah  | +create(), +updateStatus()       |

### Visibility (Akses)

| Simbol | Nama      | Arti                                 |
|--------|-----------|--------------------------------------|
| +      | Public    | Bisa diakses dari mana saja          |
| -      | Private   | Hanya bisa diakses dalam class       |
| #      | Protected | Bisa diakses oleh class turunan      |
| ~      | Package   | Bisa diakses dalam package yang sama |

### Relationship / Relasi

| Relasi      | Simbol | Arti  | Contoh                 |
|-------------|--------|---|------------------------|
| Association | —      | Hubungan biasa                                    | orders — pangkalans    |
| Aggregation | —◇     | Kepemilikan LEMAH (child bisa exist tanpa parent) | agen ◇ — pangkalans    |
| Composition | —◆     | Kepemilikan KUAT (child hapus jika parent hapus)  | orders ◆ — order_items |
| Inheritance | —▷     | Class turunan (extends)                           | Employee —▷ Person     |

| Relasi      | Simbol | Arti                      | Contoh                          |
|-------------|--------|---------------------------|---------------------------------|
| Dependency  | - - -> | Class A bergantung pada B | OrderService ---> PrismaService |
| Realization | - - -▷ | Implementasi interface    | UserService --▷ IUserService    |

Multiplicity (Kardinalitas)

| Notasi      | Arti             |
|-------------|------------------|
| 1           | Tepat satu       |
| 0..1        | Nol atau satu    |
| * atau 0..* | Nol atau banyak  |
| 1..*        | Satu atau banyak |
| n..m        | Range n sampai m |

Contoh SIM4LON

- orders ◆—— order\_items (Composition: item hapus jika order hapus)
- agen ◇—— pangkalans (Aggregation: pangkalan bisa exist tanpa agen)

3 ERD (Entity Relationship Diagram)

Definisi

Diagram yang menunjukkan **struktur database**: entitas (tabel), atribut (kolom), dan relasi.

Fungsi

- Desain database
- Menunjukkan primary key, foreign key
- Dokumentasi skema database

Komponen

| Komponen     | Simbol      | Fungsi                  |
|--------------|-------------|-------------------------|
| Entity       | □ (kotak)   | Tabel database          |
| Attribute    | Dalam kotak | Kolom tabel             |
| Primary Key  | 🔑 PK        | Identifier unik         |
| Foreign Key  | 🔗 FK        | Referensi ke tabel lain |
| Relationship | Garis       | Relasi antar tabel      |

## Crow's Foot Notation (PENTING!)

| Simbol | Nama            | Arti                     |
|--------|-----------------|--------------------------|
|        | One (mandatory) | Tepat satu, wajib ada    |
|        | Zero or One     | Nol atau satu (optional) |
|        | Many            | Banyak                   |
|        | Zero or One     | Nol atau satu            |

## Cardinality (Kardinalitas)

| Relasi              | Simbol | Arti                        | Contoh                               |
|---------------------|--------|-----------------------------|--------------------------------------|
| <b>One-to-One</b>   |        | 1 berhubungan dengan 1      | orders —  —<br>order_payment_details |
| <b>One-to-Many</b>  |        | 1 berhubungan dengan banyak | pangkalans   —o{ orders              |
| <b>Many-to-Many</b> |        | Banyak ke banyak            | (butuh junction table)               |

## Beda ERD vs Class Diagram

| Aspek   | ERD              | Class Diagram               |
|---------|------------------|-----------------------------|
| Fokus   | Database (tabel) | OOP (class)                 |
| Atribut | Kolom + tipe SQL | Property + tipe programming |
| Method  | ✗ Tidak ada      | ☑ Ada                       |
| Relasi  | FK, PK           | Association, Inheritance    |

## Contoh SIM4LON

- 21 Tables, 7 Enums
- pangkalans ||—o{ orders (1 pangkalan punya banyak order)
- orders ||—|| order\_payment\_details (1 order punya 1 payment detail)

## 4 ACTIVITY DIAGRAM

### Definisi

Diagram yang menunjukkan **alur proses/workflow** dari awal sampai akhir.

### Fungsi

- Menggambarkan langkah-langkah bisnis proses
- Menunjukkan decision/percabangan
- Dokumentasi SOP

## Komponen

| Komponen     | Simbol                | Fungsi                |
|--------------|-----------------------|-----------------------|
| Initial Node | ● (bulat hitam)       | Titik awal            |
| Final Node   | ⦿ (bulat dalam bulat) | Titik akhir           |
| Action       | ▭ (rounded rectangle) | Aktivitas/langkah     |
| Decision     | ◇ (diamond)           | Percabangan (if-else) |
| Fork         | ■ (bar horizontal)    | Parallel split        |
| Join         | ■ (bar horizontal)    | Parallel merge        |
| Swimlane     | Kolom vertikal        | Pembagian per aktor   |

## Control Flow

| Simbol      | Arti             |
|-------------|------------------|
| →           | Urutan eksekusi  |
| ◇ [kondisi] | Guard condition  |
| ■ (fork)    | Mulai parallel   |
| ■ (join)    | Selesai parallel |

## Contoh SIM4LON

- 25 Activity Diagrams
- Swimlane: User | Sistem
- Decision: [Valid?] → Yes/No

# 5 SEQUENCE DIAGRAM

## Definisi

Diagram yang menunjukkan **interaksi antar objek** dalam urutan waktu (vertikal).

## Fungsi

- Menggambarkan flow request-response
- Detail teknis komunikasi komponen
- Dokumentasi API flow

## Komponen

| Komponen    | Simbol                 | Fungsi                    |
|-------------|------------------------|---------------------------|
| Participant | □ (kotak)              | Objek yang berinteraksi   |
| Lifeline    | (garis vertikal putus) | Waktu hidup objek         |
| Activation  | ■ (bar di lifeline)    | Objek sedang aktif/proses |
| Message     | →                      | Request/pemanggilan       |
| Return      | - - ->                 | Response/return value     |

## Stereotype Participant

| Stereotype   | Simbol | Contoh                            |
|--------------|--------|-----------------------------------|
| <<actor>>    |        | User, Admin                       |
| <<boundary>> | □      | LoginPage, OrderPage (UI)         |
| <<control>>  | ⊙      | AuthService, OrderService (Logic) |
| <<entity>>   | □      | UserModel, OrderModel (Data)      |
| <<database>> | ⬡      | users, orders (Table)             |

## Message Types

| Tipe         | Simbol | Arti                               |
|--------------|--------|------------------------------------|
| Synchronous  | —▶     | Request blocking (tunggu response) |
| Asynchronous | —>     | Request non-blocking               |
| Return       | - - -> | Response/nilai balik               |
| Self-call    | ↺      | Objek panggil dirinya sendiri      |

## Fragments (Combined Fragments)

| Fragment    | Keyword | Fungsi           |
|-------------|---------|------------------|
| Alternative | alt     | If-else          |
| Option      | opt     | If tanpa else    |
| Loop        | loop    | Perulangan       |
| Break       | break   | Keluar dari loop |
| Parallel    | par     | Eksekusi paralel |

## Contoh SIM4LON

- 18 Sequence Diagrams
- Numbered messages: 1.1, 1.2, 1.2.1
- Fragments: alt (login sukses/gagal), opt (assign driver)

## 6 STATE MACHINE DIAGRAM

### Definisi

Diagram yang menunjukkan **siklus hidup (lifecycle)** suatu objek melalui berbagai state.

### Fungsi

- Menggambarkan status dan transisi
- Dokumentasi workflow status
- Validasi transisi yang valid

### Komponen

| Komponen      | Simbol      | Fungsi            |
|---------------|-------------|-------------------|
| Initial State | •           | Titik awal        |
| Final State   | ⦿           | Titik akhir       |
| State         | □ (rounded) | Kondisi objek     |
| Transition    | →           | Perpindahan state |
| Guard         | [kondisi]   | Syarat transisi   |

### Anatomy of State

STATE NAME

entry / action  
do / activity  
exit / action

← Nama state

← Aksi saat masuk state

← Aksi selama di state

← Aksi saat keluar state

### Transition Syntax

trigger [guard] / action

- **trigger**: Event yang memicu transisi
- **guard**: Kondisi yang harus terpenuhi [dalam kurung siku]

- **action:** Aksi yang dilakukan saat transisi

### Contoh SIM4LON

- SM-01: Order Status (7 states)
  - DRAFT → MENUNGGU\_PEMBAYARAN → DIPROSES → SIAP\_KIRIM → DIKIRIM → SELESAI
- SM-02: Payment Status (3 states)
  - UNPAID → PARTIAL → PAID
- SM-04: User Session (Single Session Login)

## 7 DEPLOYMENT DIAGRAM

### Definisi

Diagram yang menunjukkan **infrastruktur fisik** dan bagaimana komponen di-deploy.

### Fungsi

- Arsitektur sistem
- Protokol komunikasi
- Dokumentasi infrastruktur

### Komponen

| Komponen   | Simbol       | Fungsi                       |
|------------|--------------|------------------------------|
| Node       | □ 3D (kubus) | Server/environment           |
| Artifact   | □ dengan <>  | Aplikasi/file yang di-deploy |
| Component  | □ dengan <>  | Modul dalam artifact         |
| Connection | —            | Komunikasi antar node        |

### Stereotype

| Stereotype    | Contoh                     |
|---------------|----------------------------|
| <<device>>    | Client Browser, Mobile     |
| <<platform>>  | Vercel, Railway            |
| <<service>>   | PostgreSQL, Supabase       |
| <<artifact>>  | React App, NestJS API      |
| <<component>> | Auth Module, Order Service |

### Communication



| Protokol | Port | Contoh               |
|----------|------|----------------------|
| HTTPS    | 443  | Browser → Vercel     |
| TCP      | 5432 | Railway → PostgreSQL |
| WSS      | 443  | WebSocket            |

Contoh SIM4LON



TABEL PERBANDINGAN SEMUA DIAGRAM

| Diagram       | Menjawab           | Fokus         | Contoh Relasi            |
|---------------|--------------------|---------------|--------------------------|
| Use Case      | SIAPA bisa APA?    | Fungsional    | include, extend          |
| Class         | Struktur objek?    | OOP           | association, composition |
| ERD           | Struktur database? | Database      | one-to-many, FK          |
| Activity      | Alur proses?       | Workflow      | decision, fork           |
| Sequence      | Interaksi objek?   | Komunikasi    | message, return          |
| State Machine | Status & transisi? | Lifecycle     | transition, guard        |
| Deployment    | Dimana deploy?     | Infrastruktur | connection, protocol     |

STATISTIK SIM4LON

| Diagram       | Jumlah | File                      |
|---------------|--------|---------------------------|
| Use Case      | 1      | SIM4LON_UseCase.puml      |
| Class         | 1      | SIM4LON_ClassDiagram.puml |
| ERD           | 1      | SIM4LON_ERD.puml          |
| Activity      | 25     | AD_01 - AD_25             |
| Sequence      | 18     | SD_01 - SD_18             |
| State Machine | 4      | SM_01 - SM_04             |
| Deployment    | 1      | SIM4LON_Deployment.puml   |

| Diagram | Jumlah | File |
|---------|--------|------|
| TOTAL   | 51     |      |

Good luck presentasinya! ☺