

# DOKUMENTASI STATE MACHINE DIAGRAM

---

## Sistem Informasi Manajemen LPG (SIM4LON)

---

### 1. GAMBARAN UMUM

State Machine Diagram menggambarkan **siklus hidup (lifecycle)** suatu objek, menunjukkan:

- **States** (kondisi objek)
  - **Transitions** (perpindahan antar state)
  - **Events** (pemicu transisi)
  - **Guards** (kondisi transisi)
- 

### 2. NOTASI

Simbol	Nama	Fungsi
●	Initial State	Titik awal
○	Final State	Titik akhir
□	State	Kondisi objek
→	Transition	Perpindahan state
[guard]	Guard Condition	Syarat transisi

---

### 3. DAFTAR STATE MACHINE DIAGRAM

SM-ID	Nama	States	File
SM-01	Order Status	7	<a href="#">SM_01_OrderStatus.puml</a>
SM-02	Payment Status	3	<a href="#">SM_02_PaymentStatus.puml</a>
SM-03	Agen Order Status	4	<a href="#">SM_03_AgenOrderStatus.puml</a>
SM-04	User Session	4	<a href="#">SM_04_UserSession.puml</a>

---

### 4. DETAIL STATE MACHINE

SM-01: Order Status (Status Pesanan)

**Enum:** [status\\_pesanan](#)

State	Warna	Deskripsi
DRAFT	Default	Pesanan baru dibuat

State	Warna	Deskripsi
MENUNGGU_PEMBAYARAN	Warning	Menunggu pembayaran
DIPROSES	Active	Sedang diproses
SIAP_KIRIM	Default	Siap untuk dikirim
DIKIRIM	Active	Dalam pengiriman
SELESAI	Success	Pesanan selesai
BATAL	Danger	Dibatalkan

#### Transition Flow:

```

DRAFT → MENUNGGU_PEMBAYARAN → DIPROSES → SIAP_KIRIM → DIKIRIM → SELESAI
                               ↓
(Cancel dari state manapun) → BATAL

```

#### Business Rules:

1. **DRAFT → MENUNGGU\_PEMBAYARAN**: Saat user submit pesanan
2. **MENUNGGU\_PEMBAYARAN → DIPROSES**: Saat pembayaran diterima (lunas/DP)
3. **SIAP\_KIRIM → DIKIRIM**: Wajib assign driver terlebih dahulu
4. **DIKIRIM → SELESAI**: Auto-sync stok pangkalan
5. → **BATAL**: Bisa dari state manapun kecuali SELESAI

#### SM-02: Payment Status (Status Pembayaran)

**Field:** `order_payment_details.is_paid`, `order_payment_details.is_dp`

State	Nilai	Deskripsi
UNPAID	<code>is_paid=false, is_dp=false</code>	Belum bayar
PARTIAL	<code>is_paid=false, is_dp=true</code>	Sudah DP
PAID	<code>is_paid=true</code>	Lunas

#### Transition Flow:

```

UNPAID → PARTIAL → PAID
↓      ↑
     → PAID (bayar langsung lunas)

```

#### Guard Conditions:

- `amount < total`: Masuk ke PARTIAL

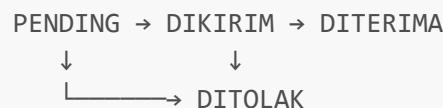
- `amount >= total`: Langsung ke PAID
- 

## SM-03: Agen Order Status

**Field:** `agen_orders.status`

State	Warna	Deskripsi
PENDING	●	Menunggu konfirmasi agen
DIKIRIM	●	Agen sudah kirim
DITERIMA	●	Pangkalan sudah terima
DITOLAK	●	Order ditolak

### Transition Flow:



### Business Rules:

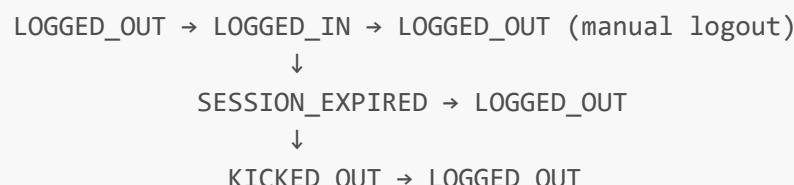
- `qty_received` bisa berbeda dari `qty_ordered` (partial delivery)
  - Saat DITERIMA: Stok pangkalan otomatis bertambah
- 

## SM-04: User Session (Single Session Login)

**Field:** `users.session_id`

State	Deskripsi
LOGGED_OUT	User belum login
LOGGED_IN	User aktif dengan token valid
SESSION_EXPIRED	Token JWT expired
KICKED_OUT	Login dari device lain (single session)

### Transition Flow:



## Single Session Feature:

- Setiap login menggenerate `session_id` baru
  - Session lama otomatis invalid
  - JWT validation memeriksa `session_id` match
- 

## 5. MATRIKS STATE & ENTITAS

State Machine	Entitas	Field	Enum/Type
SM-01	orders	current_status	status_pesanan
SM-02	order_payment_details	is_paid, is_dp	boolean
SM-03	agen_orders	status	varchar(20)
SM-04	users	session_id	varchar(100)

---

## 6. CATATAN TEKNIS

### 6.1 State Entry Actions

- **SM-01 SELESAI:** Auto-update `pangkalan_stocks`, create `pangkalan_stock_movements`
- **SM-03 DITERIMA:** Auto-update `pangkalan_stocks`, create `pangkalan_stock_movements`

### 6.2 Guard Conditions

- Guards ditulis dalam format `[condition]`
- Contoh: `[is_paid = true], [amount >= total]`

### 6.3 Composite States

- SM-01 LOGGED\_IN: Memiliki internal behaviors (Do: Monitor token)
- 

Dokumen ini menjelaskan 4 State Machine Diagram SIM4LON

Total States: 18 | Total Transitions: 25+