

SKRIP PRESENTASI SIM4LON

Alur Lengkap dari Awal sampai Akhir

BAGIAN 1: OPENING (2-3 menit)

1.1 Salam Pembuka

"Assalamualaikum warahmatullahi wabarakatuh. Selamat [pagi/siang/sore], Bapak/Ibu penguji dan hadirin sekalian."

"Perkenalkan, nama saya [NAMA LENGKAP], dengan NIM [NIM]. Pada kesempatan kali ini, saya akan mempresentasikan proyek Tugas Akhir saya yang berjudul '**SIM4LON - Sistem Informasi Manajemen LPG 4 Jalur Online**'."

1.2 Latar Belakang Masalah

"Sebelum masuk ke demo sistem, izinkan saya menjelaskan latar belakang mengapa sistem ini dibangun."

"Saat ini, proses distribusi LPG dari **Agen** ke **Pangkalan** hingga ke **Konsumen** masih banyak dilakukan secara **manual**. Pencatatan menggunakan buku tulis, komunikasi melalui WhatsApp, dan perhitungan stok masih menggunakan Excel atau bahkan ingatan."

"Hal ini menimbulkan beberapa **masalah**:

1. **Data tidak akurat** - sering terjadi selisih stok
2. **Pembayaran tercecer** - sulit melacak mana yang sudah bayar, mana yang belum
3. **Tidak ada audit trail** - kalau ada kesalahan, sulit melacak siapa yang melakukan
4. **Laporan manual** - memakan waktu berjam-jam untuk rekap bulanan"

"Oleh karena itu, saya membangun **SIM4LON** untuk **mendigitalisasi** seluruh proses ini."

1.3 Tujuan Sistem

"Tujuan utama SIM4LON adalah:

1. Mempermudah **pencatatan pesanan** LPG dari pangkalan
 2. Memantau **stok** secara real-time
 3. Mencatat **pembayaran** dengan rapi
 4. Menyediakan **laporan** yang akurat dan cepat
 5. Mendukung **multi-tenant** - setiap pangkalan hanya bisa akses data miliknya"
-

BAGIAN 2: ARSITEKTUR & TEKNOLOGI (3-4 menit)

2.1 Full Stack Overview

"Selanjutnya, izinkan saya menjelaskan arsitektur sistem yang saya gunakan."

"SIM4LON dibangun dengan arsitektur **Full Stack Modern**."

"Untuk **Frontend**, saya menggunakan:

- **React 18** sebagai library UI
- **Vite** sebagai build tool
- **Tailwind CSS** dan **Shadcn/UI** untuk styling
- **TanStack Query** untuk data fetching dan caching"

"Untuk **Backend**, saya menggunakan:

- **NestJS 10** sebagai framework
- **Prisma ORM** untuk koneksi ke database
- **JWT** dan **Passport** untuk autentikasi"

"Untuk **Database**, saya menggunakan **PostgreSQL 15** dengan total **21 tabel** dan **7 enum types**."

"Untuk **File Storage**, saya menggunakan **Supabase Storage** yang S3-compatible."

2.2 Deployment Architecture

"Untuk deployment, sistem ini di-host di cloud:

- **Vercel** untuk frontend
- **Railway** untuk backend dan database
- **Supabase** untuk file storage"

"Komunikasi antar komponen menggunakan **HTTPS** dan **RESTful API**."

BAGIAN 3: DEMO LIVE SISTEM (7-10 menit)

3.1 Halaman Login

"Sekarang, mari kita masuk ke demo sistem."

"Ini adalah halaman **Login**. Sistem SIM4LON memiliki fitur **Single-Session Login**, artinya satu user hanya bisa login dari **satu device** dalam satu waktu. Jika login dari device baru, session di device lama akan otomatis logout."

"Mari kita login sebagai **Admin** dengan email admin@agen.com..."

[Lakukan login]

3.2 Dashboard Admin

"Setelah login, kita masuk ke **Dashboard Admin**. Di sini kita bisa melihat:

- **Pesanan hari ini** - berapa total pesanan yang masuk
- **Total pendapatan** - ringkasan finansial
- **Stok LPG** - kondisi stok saat ini
- **Aktivitas terbaru** - log aktivitas sistem"

"Dashboard ini memberikan **overview** cepat tentang kondisi bisnis saat ini."

3.3 Kelola Pesanan

"Selanjutnya, mari kita lihat menu **Pesanan**."

"Di sini admin bisa melihat semua pesanan dari pangkalan. Pesanan memiliki **7 status**:

1. DRAFT
2. MENUNGGU_PEMBAYARAN
3. DIPROSES
4. SIAP_KIRIM
5. DIKIRIM
6. SELESAI
7. BATAL"

Membuat Pesanan Baru

"Mari kita coba **buat pesanan baru**..."

[Klik tombol Buat Pesanan]

"Kita pilih pangkalan, masukkan jenis LPG dan jumlahnya. Perhatikan bahwa untuk LPG **Non-Subsidi**, sistem akan otomatis menghitung **PPN 12%**."

[Isi form dan submit]

"Pesanan berhasil dibuat dengan status **DRAFT**."

3.4 Update Status Pesanan

"Sekarang mari kita update status pesanan ini."

[Klik pesanan, update status]

"Kita ubah dari DRAFT ke MENUNGGU_PEMBAYARAN, lalu ke DIPROSES setelah pembayaran diterima."

"Perhatikan bahwa setiap perubahan status, sistem akan mencatat di **Timeline** pesanan. Ini adalah **audit trail** yang berguna untuk tracking."

3.5 Fitur Auto-Sync Stok

"Ini adalah fitur penting: ketika status diubah menjadi **SELESAI**, sistem akan **otomatis menambah stok** di pangkalan."

[Update status ke SELESAI]

"Sekarang kalau kita cek stok pangkalan tersebut, stoknya sudah bertambah sesuai jumlah pesanan. Ini mengurangi **human error** karena tidak perlu input manual."

3.6 Kelola Pembayaran

"Di menu **Pembayaran**, admin bisa mencatat pembayaran dari pangkalan."

"Ada dua metode: **TUNAI** atau **TRANSFER**. Sistem juga mendukung **DP (Down Payment)**, jadi pembayaran bisa dilakukan bertahap."

"Ketika pembayaran sudah **LUNAS**, status pesanan otomatis berubah ke **DIPROSES**."

3.7 Kelola Stok

"Menu **Stok** terdiri dari dua bagian:

1. **Penerimaan** - catat stok masuk dari SPBE (Stasiun Pengisian Bulk Elpiji)
2. **Penyaluran** - catat stok keluar ke pangkalan"

"Setiap pergerakan stok tercatat di **Stock Histories** untuk audit."

3.8 Role Pangkalan (Multi-Tenant)

"Sekarang saya akan logout dan login sebagai **Pangkalan** untuk menunjukkan fitur **multi-tenant**."

[Logout, login sebagai pangkalan]

"Perhatikan bahwa tampilan berbeda. Pangkalan hanya bisa melihat:

- Stok **miliknya sendiri**
- Penjualan **miliknya sendiri**
- Konsumen **miliknya sendiri**

"Ini adalah implementasi **Row-Level Security** yang memastikan setiap pangkalan **terisolasi** datanya."

BAGIAN 4: UML DIAGRAM (5-7 menit)

4.1 Use Case Diagram

"Sekarang mari kita bahas dokumentasi sistem. Dimulai dari **Use Case Diagram**."

"Sistem ini memiliki **3 aktor**:

1. **Admin** - full access termasuk master data
2. **Operator** - akses operasional (pesanan, stok, pembayaran)
3. **Pangkalan** - hanya akses data milik sendiri"

"Total ada **23 use case** yang terbagi sesuai hak akses masing-masing aktor."

"Relasi yang digunakan:

- **Include** untuk fungsionalitas wajib, contoh: Kelola Stok include Catat Penerimaan
- **Extend** untuk fungsionalitas opsional, contoh: Assign Driver extend dari Update Status"

4.2 ERD (Entity Relationship Diagram)

"Selanjutnya **ERD**. Database terdiri dari **21 tabel** dengan **7 enum types**."

"Tabel-tabel utama:

- **users** - data pengguna
- **pangkalans** - data pangkalan
- **orders + order_items** - pesanan dan detailnya
- **payment_records** - catatan pembayaran
- **stock_histories** - riwayat stok"

"Relasi menggunakan **Crow's Foot Notation**:

- pangkalans 1:N orders (satu pangkalan banyak pesanan)
- orders 1:1 order_payment_details (satu pesanan satu summary bayar)
- orders 1:N order_items (satu pesanan banyak item)"

4.3 State Machine Diagram

"Untuk **State Machine**, yang paling penting adalah **Status Pesanan**."

"Alurnya: DRAFT → MENUNGGU_PEMBAYARAN → DIPROSES → SIAP_KIRIM → DIKIRIM → SELESAI"

"Transisi diatur dengan **guard condition**, misalnya:

- Dari MENUNGGU_PEMBAYARAN ke DIPROSES hanya bisa jika **is_paid = true**
- Dari SIAP_KIRIM ke DIKIRIM hanya bisa jika **driver sudah di-assign**"

"Ada juga state **BATAL** yang bisa dicapai dari state manapun kecuali SELESAI."

4.4 Deployment Diagram

"Terakhir, **Deployment Diagram** menunjukkan arsitektur infrastruktur."

"Alurnya:

- Browser → Vercel (Frontend React)
- Vercel → Railway (Backend NestJS)

- Railway → PostgreSQL (Database)
- Railway → Supabase (File Storage)"

"Semua komunikasi menggunakan **HTTPS** untuk keamanan."

BAGIAN 5: FITUR UNGGULAN (2-3 menit)

5.1 Highlight Fitur

"Sebelum menutup, izinkan saya me-highlight beberapa fitur unggulan SIM4LON:"

"1. Single-Session Login" "Satu user hanya bisa login dari satu device. Ini mencegah sharing password dan meningkatkan keamanan."

"2. Auto-Sync Stok" "Ketika pesanan SELESAI, stok pangkalan otomatis bertambah. Mengurangi human error."

"3. Multi-Tenant" "Setiap pangkalan hanya bisa akses data miliknya. Data privacy terjamin."

"4. PPN 12% Otomatis" "Untuk LPG Non-Subsidi, pajak dihitung otomatis. Tidak perlu kalkulasi manual."

"5. Audit Trail" "Semua aktivitas tercatat di log. Mudah untuk tracking jika ada masalah."

BAGIAN 6: CLOSING (1-2 menit)

6.1 Kesimpulan

"Sebagai kesimpulan, **SIM4LON** berhasil:

1. Mendigitalisasi proses distribusi LPG yang sebelumnya manual
2. Menyediakan dashboard real-time untuk monitoring bisnis
3. Memastikan keamanan data dengan single-session login dan multi-tenant
4. Mempermudah pembuatan laporan dengan fitur export"

6.2 Dokumentasi

"Seluruh sistem telah didokumentasikan dengan **51 diagram UML**:

- 1 Use Case Diagram
- 1 Class Diagram
- 1 ERD
- 25 Activity Diagram
- 18 Sequence Diagram
- 4 State Machine Diagram
- 1 Deployment Diagram"

6.3 Sesi Tanya Jawab

"Demikian presentasi dari saya. Apakah ada pertanyaan dari Bapak/Ibu penguji?"

[Siap menjawab pertanyaan]

6.4 Penutup

"Baik, jika tidak ada pertanyaan lagi, saya ucapkan terima kasih atas waktu dan perhatiannya."

"Wassalamualaikum warahmatullahi wabarakatuh."

LAMPIRAN: PERTANYAAN YANG MUNGKIN DITANYAKAN

Pertanyaan	Jawaban
"Kenapa pakai React?"	"React dipilih karena arsitektur component-based, ecosystem yang besar, dan performa yang baik untuk SPA."
"Kenapa NestJS?"	"NestJS dipilih karena menggunakan TypeScript, arsitektur modular, dan cocok untuk aplikasi enterprise."
"Bagaimana keamanannya?"	"Keamanan dijaga dengan JWT token, session_id untuk single-session, bcrypt untuk password hashing, dan HTTPS untuk transport."
"Apa bedanya ERD dan Class Diagram?"	"ERD fokus pada struktur database (tabel, kolom, FK), sedangkan Class Diagram fokus pada struktur OOP (class, method, inheritance)."
"Apa itu Include dan Extend?"	"Include adalah relasi wajib (selalu dipanggil), Extend adalah relasi opsional (dipanggil dalam kondisi tertentu)."
"Kenapa pakai PostgreSQL?"	"PostgreSQL dipilih karena mendukung UUID, ACID compliant, dan memiliki fitur lengkap untuk relational database."

Good luck! 🍀 Semoga presentasinya lancar dan sukses!