

UTS Analisi Desain

“Komita Café”



Oleh:

Melkior Nuwa/23083000138

Michael Sanaty/23083000124

4F

UNIVERSITAS MERDEKA MALANG
FAKULTAS TELNOLOGI INFOMASI PRODI
SISTEM INFOMASI
TAHUN 2025

PENGEMBANGAN SISTEM INFORMASI

“Komita Café”

a. Gambaran Singkat Komita Cafe (Kopi Milik Kita)

Komita Cafe, atau dikenal juga sebagai Kopi Milik Kita, adalah sebuah kafe yang berlokasi di Jl. Rajekwesi No.17c, Gading Kasri, Kec. Klojen, Kota Malang, Jawa Timur 65115. Kafe ini menyediakan beragam menu kopi dan teh yang enak, dengan suasana yang nyaman dan santai. Komita Cafe cocok untuk berbagai kalangan, mulai dari mahasiswa, keluarga, hingga anak-anak.

Komita Cafe memiliki beberapa fasilitas dan layanan, seperti:

- Tempat duduk di area terbuka
- Layanan makan di tempat dan bawa pulang
- Cocok untuk pengunjung berkelompok
- Ramah untuk anak-anak

b. Prosedur yang Sedang Berjalan (Sistem Lama)

Berdasarkan observasi di Komita Cafe, berikut adalah deskripsi prosedur manual yang masih diterapkan hingga saat ini:

1. Pemesanan

- Pelanggan melakukan pemesanan langsung ke kasir atau pelayan secara lisan.
- Pesanan dicatat secara manual di nota kertas atau buku catatan.
- Proses ini memakan waktu, rentan terjadi kesalahan pencatatan, serta tidak tercatat secara digital untuk pelacakan ulang.

2. Pengelolaan Antrian

- Tidak ada sistem antrian yang terstruktur.
- Pelanggan menunggu berdasarkan urutan datang tanpa nomor antrian, yang seringkali menimbulkan kebingungan, terutama di jam-jam sibuk.
- Staf harus mengingat sendiri urutan pesanan, yang dapat menimbulkan ketidaktepatan penyajian.

3. Pembuatan Menu & Penyajian

- Setelah dicatat, nota fisik diserahkan ke bagian dapur atau barista untuk diproses.
- Tidak ada sistem integrasi antara bagian kasir dan dapur, sehingga komunikasi sepenuhnya mengandalkan nota manual.

4. Pembayaran

- Pembayaran dilakukan langsung ke kasir, umumnya tunai atau menggunakan QRIS/manual transfer.
- Tidak tersedia sistem POS (Point of Sale) digital, sehingga semua transaksi dicatat secara manual.
- Tidak ada bukti pembayaran digital, dan proses pelacakan transaksi cukup sulit.

5. Manajemen Stok

- Stok bahan baku dicatat di buku atau spreadsheet oleh pemilik.
- Pemantauan stok dilakukan secara manual dan tidak terintegrasi dengan transaksi penjualan.
- Risiko kehabisan stok atau kelebihan stok cukup tinggi karena tidak ada sistem real-time yang mencatat pengeluaran bahan secara otomatis saat pesanan dibuat.

ANALISIS SISTEM KOMITA

CAFE

a. Analisis Masalah Sistem Lama / Sistem yang Sedang Berjalan

1. Tidak Ada Sistem Pemesanan Digital

Pelanggan harus memesan langsung secara manual ke kasir atau pelayan, sehingga proses pemesanan menjadi lambat, rawan kesalahan, dan sulit untuk melakukan pencatatan riwayat pesanan.

2. Sistem Stok Tidak Bisa Terpantau Secara Digital

Pengelolaan stok bahan baku dilakukan secara manual, tidak ada pemantauan real-time sehingga kerap terjadi kehabisan bahan atau overstock yang menyebabkan pemborosan.

3. Sistem Antrian yang Tidak Jelas

Sistem antrian pelanggan tidak terorganisir dengan baik, menyebabkan ketidakteraturan dan ketidaknyamanan, terutama saat jam sibuk.

b. Solusi yang Ditawarkan pada Sistem yang Baru

1. Penerapan Sistem Pemesanan Digital

- Mengembangkan aplikasi atau web-based system untuk pemesanan, yang memungkinkan pelanggan memesan dari perangkat mereka atau melalui kasir dengan input digital.
- Meminimalisir kesalahan input dan mempercepat proses transaksi.
- Sistem berbasis aplikasi mobile atau web memungkinkan pelanggan memesan kapan saja dan dari mana saja, misalnya via smartphone.
- Pemesanan digital bisa dilakukan pelanggan sendiri, mengurangi beban kasir dan mempercepat proses transaksi.
- Kasir juga menggunakan aplikasi untuk memasukkan pesanan langsung ke sistem, menghilangkan pencatatan manual yang rawan kesalahan.
- Sistem mencatat semua transaksi secara otomatis, memudahkan pembuatan laporan dan analisis data.
- Fitur tambahan bisa berupa metode pembayaran digital, integrasi dengan e-wallet atau kartu kredit, mempercepat proses checkout.

2. Implementasi Sistem Manajemen Stok Otomatis

- Menggunakan modul stok terintegrasi yang otomatis mengurangi stok setiap kali terjadi transaksi penjualan.
- Menyediakan laporan stok secara real-time dan peringatan otomatis saat stok hampir habis.
- Setiap transaksi penjualan otomatis mengurangi stok bahan baku sesuai jumlah bahan yang dipakai untuk produk tersebut (misalnya 1 porsi kopi mengurangi stok 100 gr biji kopi).
- Sistem menyimpan data stok secara real-time dan menyediakan dashboard untuk monitoring kondisi stok secara visual.
- Notifikasi otomatis dikirim ke manajer/pengelola ketika stok mencapai batas minimum yang sudah ditentukan, sehingga bisa langsung dilakukan pengadaan ulang.

- Laporan stok bulanan dan histori penggunaan bahan baku memudahkan analisis tren konsumsi bahan dan efisiensi pengelolaan stok.

3. Sistem Antrian Digital Terintegrasi

- Menggunakan sistem nomor antrian digital yang dapat dilihat pelanggan melalui layar display.
- Meningkatkan efisiensi pelayanan dan mengurangi kebingungan saat antri.
- Pelanggan diberikan nomor antrian secara otomatis saat melakukan pemesanan digital atau di kasir.
- Nomor antrian ditampilkan secara real-time pada layar display di ruang tunggu agar pelanggan tahu giliran mereka.
- Sistem mengupdate status antrian secara otomatis sesuai dengan kemajuan pelayanan, mengurangi kebingungan.
- Staf dapur dan kasir juga memiliki akses ke sistem antrian untuk mengatur prioritas penyajian dan pembayaran.
- Fitur ini sangat membantu mengelola jam sibuk dengan efektif dan meningkatkan pengalaman pelanggan.

c. Analisis Kebutuhan Fungsional Sistem

- **Pemesanan Digital:** Sistem harus menyediakan user interface yang mudah diakses melalui berbagai device (mobile, desktop, tablet) agar pelanggan dapat memesan dengan cepat dan mudah.
- **Manajemen Stok Otomatis:** Harus ada sistem backend yang mengelola data stok, menghubungkan transaksi dengan database stok, dan melakukan update secara real-time.
- **Status Antrian Real-Time:** Sistem wajib menampilkan antrian yang up-to-date untuk pelanggan dan staf, dengan mekanisme update otomatis setiap kali terjadi perubahan status pesanan.
- **Laporan:** Sistem harus mampu meng-generate laporan komprehensif mengenai penjualan, stok bahan baku, serta profil dan histori pelanggan untuk kebutuhan analisis bisnis dan marketing.
- **Modul Login Multi-role:** Sistem harus mendukung multi-level akses (admin, kasir, staf dapur) untuk keamanan dan pengaturan hak akses sesuai tugas dan tanggung jawab masing-masing.
- **Notifikasi:** Harus ada fitur push notification/email otomatis sebagai pengingat penting, terutama terkait stok bahan dan status pesanan.

d. Analisis Kebutuhan Non Fungsional Sistem

1. Perangkat Keras:

- Server yang cukup kuat, baik server lokal di tempat atau cloud, untuk menangani database dan aplikasi dengan performa stabil.
- Komputer/tablet yang responsif dan tahan lama untuk kasir dan dapur agar sistem berjalan lancar.

- Monitor antrian yang jelas dan mudah dibaca oleh pelanggan.
- Infrastruktur jaringan yang handal agar koneksi antar perangkat dan server stabil tanpa gangguan.

2. Perangkat Lunak:

- Sistem operasi server yang reliable (Windows Server atau Linux) sesuai kebutuhan.
- Basis data relasional yang dapat menangani transaksi dalam jumlah besar dengan cepat dan aman (MySQL, PostgreSQL).
- Front-end yang interaktif dan responsif, misalnya menggunakan framework React atau Vue untuk pengalaman pengguna yang baik.
- Sistem keamanan yang meliputi firewall, antivirus, dan protokol enkripsi untuk melindungi data pengguna dan transaksi.

3. Jaringan:

- Koneksi internet berkecepatan tinggi dan stabil agar aplikasi dapat diakses tanpa lag terutama saat banyak pengguna aktif.
- LAN yang terkonfigurasi dengan baik agar komunikasi antar perangkat kasir, dapur, dan display antrian berjalan lancar dan aman.

4. Aspek Sistem Baru:

- *Kinerja*: Sistem harus mampu melayani puluhan hingga ratusan transaksi per jam tanpa delay, termasuk saat traffic puncak.
- *Keamanan*: Data pelanggan dan transaksi harus terlindungi dari ancaman eksternal maupun internal, dengan otentikasi multi-level dan enkripsi data penting.
- *Ketersediaan*: Sistem harus dirancang dengan mekanisme failover dan backup agar downtime sangat minim dan sistem selalu bisa diakses.
- *Usability*: Desain UI/UX harus intuitif dan mudah dipahami tanpa perlu pelatihan lama, sehingga staf dan pelanggan dapat menggunakan sistem dengan cepat dan nyaman.
- *Skalabilitas*: Sistem harus modular dan fleksibel agar bisa dikembangkan seiring pertumbuhan bisnis, misalnya menambah fitur baru atau menangani volume transaksi lebih besar.
- *Kompatibilitas*: Harus support berbagai perangkat dan platform (Windows, Android, iOS, browser populer) agar pengguna bebas menggunakan device favorit mereka.

e. Analisis Pengkodean

- **Bahasa Pemrograman:**
Pengembangan backend bisa menggunakan PHP (framework Laravel atau CodeIgniter), Python (Django/Flask), atau JavaScript (Node.js) untuk mengatur logika bisnis dan database.
Frontend menggunakan JavaScript dengan framework modern seperti React.js atau Vue.js untuk membuat tampilan yang interaktif dan responsif.
- **Struktur Kode Modular:**
Sistem dibuat dengan arsitektur modular, misalnya Model-View-Controller (MVC), agar

setiap komponen (pemrosesan data, tampilan, logika bisnis) terpisah dan mudah dikelola. Kode yang modular memudahkan pemeliharaan, debugging, dan penambahan fitur di masa depan tanpa mengganggu keseluruhan sistem.

- **Integrasi API:**

Sistem bisa terintegrasi dengan layanan pihak ketiga, misalnya notifikasi SMS/Email, pembayaran digital, atau layanan cloud untuk backup dan monitoring.

- **Keamanan Kode:**

Praktik coding yang aman harus diikuti, seperti validasi input, proteksi dari SQL Injection, Cross-site scripting (XSS), dan penggunaan HTTPS untuk komunikasi data.

f. Analisis Dokumen

- Dokumen Kebutuhan Sistem (SRS): Spesifikasi kebutuhan fungsional dan non-fungsional.
- Dokumen Desain Sistem: Diagram alur, ERD, wireframe tampilan.
- Dokumen Pengujian: Rencana dan hasil pengujian fungsional dan non-fungsional.
- Dokumentasi Pengguna: Manual penggunaan sistem untuk staf dan admin.
- Dokumentasi Pemeliharaan: Panduan troubleshooting dan pengembangan sistem.

g. Analisis Model Data (ERD)

Entitas dan Relasi Utama:

- Pelanggan (ID_Pelanggan, Nama, No_Telp, Email)
- Pesanan (ID_Pesanan, Tanggal, Total_Harga, ID_Pelanggan)
- Detail_Pesanan (ID_Detail, ID_Pesanan, ID_Menu, Jumlah, Harga_Satuan)
- Menu (ID_Menu, Nama_Menu, Harga, Stok)
- Stok_Bahan (ID_Bahan, Nama_Bahan, Jumlah_Stok, Satuan)
- Antrian (ID_Antrian, Nomor_Antrian, Status, Waktu_Masuk)

Penjelasan:

- Pelanggan dapat membuat banyak Pesanan.
- Setiap Pesanan terdiri dari beberapa Detail_Pesanan (item-menu yang dipesan).
- Menu memiliki stok bahan baku yang dicatat di Stok_Bahan.
- Antrian menyimpan nomor antrian dan status layanan.

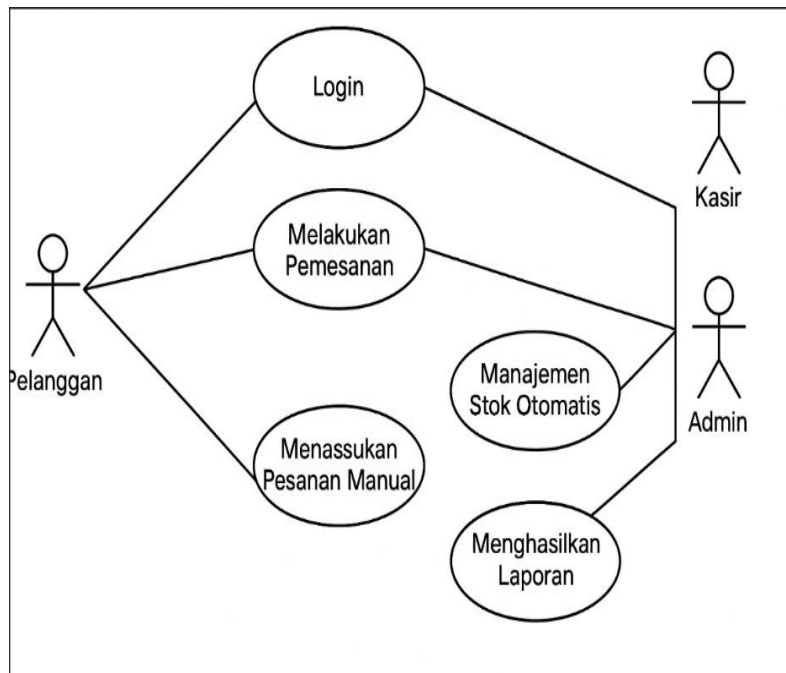
Kamus Data:

Nama Field	Type Data	Keterangan
ID_Pelanggan	INT	Primary key, unik pelanggan
Nama	VARCHAR	Nama pelanggan
No_Telp	VARCHAR	Nomor telepon pelanggan
ID_Pesanan	INT	Primary key pesanan
Tanggal	DATETIME	Waktu pesanan dibuat
Total_Harga	DECIMAL	Total harga pesanan
ID_Menu	INT	ID menu yang dipesan

Jumlah	INT	Banyak item menu
ID Pelanggan	INT	Primary key, unik pelanggan
Nama	VARCHAR	Nama pelanggan
No Telp	VARCHAR	Nomor telepon pelanggan
ID Pesanan	INT	Primary key pesanan
Tanggal	DATETIME	Waktu pesanan dibuat

Analisi Sistem

a. Use case Diagram



1. Skenario Use Case: Login

Aktor: Pelanggan, Kasir, Admin

Deskripsi: Pengguna masuk ke sistem untuk mengakses fitur sesuai peran.

Aktor	Sistem
Skenario normal	
1. Pengguna membuka aplikasi/web sistem.	
2. Pengguna memasukkan username dan password.	
	3. Sistem memverifikasi data login.

	4. Sistem memberikan akses sesuai peran (pelanggan, kasir, admin).
Scenario alternatif	
	5. Username/password salah → sistem menampilkan pesan error dan meminta login ulang.

2. Use Case: Melakukan Pemesanan

- Aktor: Pelanggan
- Tujuan: Memesan makanan/minuman secara mandiri lewat sistem.
- Pra-Kondisi: Pengguna telah login ke sistem.
- Pasca-Kondisi: Pesanan tersimpan dan muncul di antrian dapur.

Aktor	Sistem
Skenario normal	1. Pelanggan login ke sistem.
2. Pelanggan memilih menu makanan/minuman.	
3. Pelanggan memasukkan jumlah dan catatan khusus jika ada.	
4. Pelanggan menekan tombol “Pesan”.	
	5. Sistem menyimpan pesanan dan menampilkan nomor antrian serta estimasi waktu.
Alur alternatif	
	6. Menu tidak tersedia → sistem memberi notifikasi.
	7. Koneksi internet terputus → sistem menampilkan pesan kegagalan transaksi.

3. Use Case: Memasukkan Pesanan Manual

- Aktor: Kasir
- Tujuan: Memasukkan pesanan pelanggan secara manual jika tidak menggunakan sistem mandiri.
- Pra-Kondisi: Kasir telah login ke sistem.
- Pasca-Kondisi: Pesanan tercatat dan masuk ke dapur serta antrian.

Aktor	Sistem
-------	--------

Skenario Normal	
1. Pelanggan datang dan menyampaikan pesanan.	
	2. Kasir login ke sistem dan membuka modul input pesanan.
3. Kasir memilih menu, jumlah, dan catatan khusus.	
4. Kasir menekan tombol “Simpan”.	
	5. Sistem mencatat pesanan dan mengirimkannya ke dapur/antrian.

4. Use Case: Manajemen Stok Otomatis

- Aktor: Sistem
- Tujuan: Menjaga ketersediaan stok bahan baku dengan mengurangi secara otomatis.
- Pra-Kondisi: Terdapat data resep yang menghubungkan menu dengan bahan baku.
- Pasca-Kondisi: Stok bahan berkurang sesuai pesanan dan dashboard diperbarui.

Aktor	Sistem
Skenario normal	
1. Sistem mendeteksi pesanan baru.	
2. Sistem menghitung bahan baku yang diperlukan.	
3. Sistem mengurangi stok otomatis.	
4. Jika stok minimum tercapai, sistem mengirim notifikasi ke kasir/admin.	
	5. Data stok diperbarui secara real-time.
Alur alternatif	
	6. Notifikasi gagal terkirim → sistem tandai stok sebagai “perlu dicek manual”.

5. Use Case: Menghasilkan Laporan

Aktor: Admin atau Kasir dengan hak akses

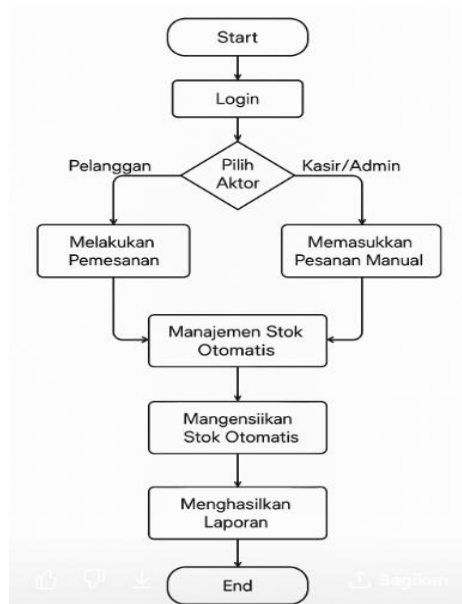
Tujuan: Menyediakan laporan penjualan, stok, atau pelanggan.

Pra-Kondisi: Pengguna telah login dan memiliki hak akses laporan.

Pasca-Kondisi: Laporan berhasil ditampilkan, diunduh, atau dicetak.

Aktor	Sistem
Skenario Normal	
	1. Admin login ke sistem
2. Admin membuka modul laporan	
3. Admin memilih jenis laporan	
	4. Sistem memproses dan menampilkan laporan
5. Admin mencetak atau mengunduh laporan	
Alur alternatif	
	6. Jika terjadi kesalahan data → sistem tampilkan log error dan tawarkan opsi perbaikan

b. Flow Map



PENJELASAN

1. Start
 - Titik awal proses dalam sistem.
2. Login
 - Pengguna (baik pelanggan maupun kasir/admin) harus melakukan login terlebih dahulu untuk mengakses sistem.
3. Pilih Aktor
 - Setelah login, sistem meminta pengguna untuk memilih peran/aktor:
Pelanggan
Kasir/Admin

4. Proses Berdasarkan Aktor

- Jika Pelanggan:
Melakukan Pemesanan (secara langsung melalui sistem).
- Jika Kasir/Admin:
Memasukkan Pesanan Manual (misalnya pelanggan pesan di tempat dan kasir input ke sistem).

5. Manajemen Stok Otomatis

- Setelah pesanan masuk (baik dari pelanggan langsung maupun dari kasir/admin), sistem otomatis mengelola stok:
- Mengurangi jumlah barang dari gudang/stok sesuai pesanan.

6. Menganalisis Stok Otomatis

- Sistem melakukan pengecekan ketersediaan barang:
- Apakah stok masih mencukupi?
- Perlu reorder atau tidak?

7. Menghasilkan Laporan

- Berdasarkan pesanan dan manajemen stok, sistem membuat laporan:
- Bisa berupa laporan penjualan, laporan stok, atau performa.

8. End

- Proses berakhir setelah laporan dihasilkan.