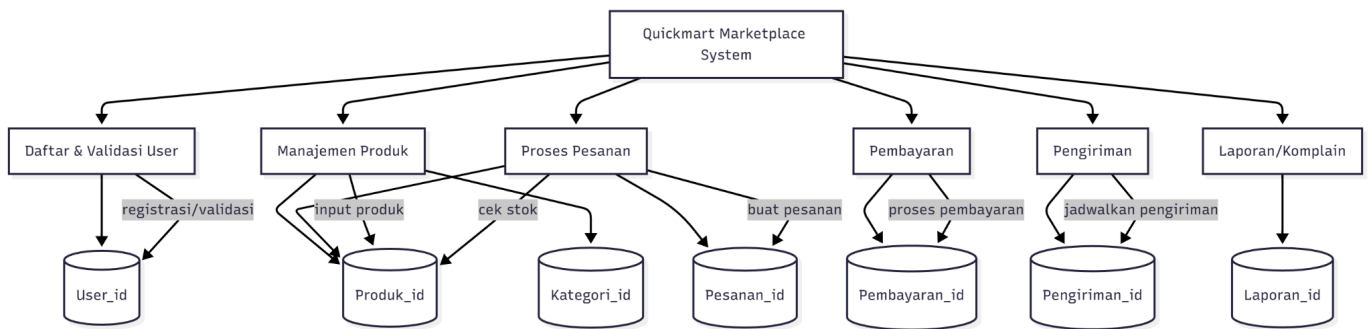


1. Diagram DFD



• Analisis Berdasarkan DFD

Berdasarkan analisis DFD (Data Flow Diagram), Quickmart memiliki alur End-to-end.

- User mendaftar dan divalidasi.
- Penjual mengelola produk.
- Pembeli membuat pesanan → sistem memproses.
- Pembayaran diverifikasi → pengiriman dijadwalkan.
- Setelah selesai, laporan/komplain bisa dibuat.

A. Proses-Proses Utama Sistem

NO.	Proses Utama	Deskripsi	Input	Output
1	Daftar dan Validasi User	Pengguna baru mendaftar dan diverifikasi oleh sistem.	- Data pendaftaran: nama, email, password, role (Penjual/Pembeli) - Permintaan login/registrasi	- Akun pengguna valid (User_id) - Status validasi (berhasil/gagal)
2	Manajemen Produk	Penjual menambahkan produk baru,	- Data produk baru dari penjual (nama produk,	- Data produk tersimpan (Produk_id)

		memperbarui stok, dan mengelola detail produk.	kategori, harga, stok) - Permintaan update stok	- Konfirmasi update stok - Informasi produk untuk ditampilkan ke pembeli
3	Proses Pesanan	Pembeli memilih produk, membuat pesanan, dan sistem mengecek stok produk.	- Permintaan pembelian produk dari pembeli - Data produk & stok dari sistem - Data user pembeli	- Data pesanan baru (Pesanan_id) - Informasi total harga - Status pesanan (“Menunggu”)
4	Pembayaran	Pembeli melakukan pembayaran dan sistem memprosesnya hingga status berubah (Menunggu → Berhasil).	- Data pesanan (Pesanan_id, total_harga) - Pilihan metode pembayaran (E-Wallet / Bank / COD) - Data user pembeli	- Bukti pembayaran (Pembayarab_id) - Status pembayaran (“Menunggu”, “Berhasil”, “Gagal”)
5	Pengiriman	Setelah pembayaran berhasil, sistem menjadwalkan pengiriman produk ke pembeli.	- Data pesanan terbayar (Pesanan_id) - Data alamat pembeli - Jadwal kirim	- Data pengiriman (Pengiriman_id) - Status pengiriman (“Dikirim”, “Selesai”)
6	Laporan/	Pembeli dapat	- Data laporan	- Catatan laporan

	Komplain	memberikan laporan, komplain, atau feedback terhadap transaksi.	atau keluhan dari user - Data transaksi terkait (Pesanan_id, Pembayaran_id)	(Laporan_id) - Status tindak lanjut laporan (“Diproses”, “Selesai”)
--	----------	---	--	--

B. Layar Input & Output yang dibutuhkan oleh Sistem

NO	Jenis Layar	Fungsi	Layar Output
1	Layar Input	Memasukkan data baru ke sistem	<ul style="list-style-type: none"> - Pesan validasi (input benar/salah) - Konfirmasi data berhasil disimpan - ID otomatis hasil input (User_id, Produk_id, Pesanan_id, Pembayaran_id, Laporan_id)
2	Layar Output	Menampilkan hasil proses/data	<ul style="list-style-type: none"> - Informasi pengguna yang login (nama, role) - Daftar produk (nama, harga, stok, kategori) - Total transaksi dan status pesanan - Status pembayaran (<i>Menunggu, Berhasil, Gagal</i>) - Ringkasan aktivitas user (penjualan/pembelian)

3	Layar Transaksi	Menghubungkan input dan output dalam satu alur	<ul style="list-style-type: none"> - Data pesanan (produk, jumlah, total harga, tanggal) - Metode pembayaran yang dipilih - Status pesanan (<i>Menunggu, Diproses, Dikirim, Selesai</i>) - Status pengiriman dan nomor resi (jika ada)
4	Layar Master Data	Mengelola data inti sistem	<ul style="list-style-type: none"> - Daftar seluruh data (produk, kategori, user) - Detail atribut tiap entitas (ID, nama, deskripsi, tanggal input) - Status aktif/nonaktif dari data
5	Layar Laporan	Menampilkan hasil evaluasi atau rekap data	<ul style="list-style-type: none"> - Ringkasan penjualan per produk / per user - Data transaksi lengkap (pesanan, pembayaran, pengiriman) - Daftar laporan atau komplain (isi laporan, status, tanggal tanggapan)

C. Kebutuhan DDL, DML, DQL

- a. DDL (Data Definition Language) digunakan untuk membuat struktur tabel dan view:

```
CREATE TABLE user (...);  
CREATE TABLE kategori (...);  
CREATE TABLE produk (...);  
CREATE TABLE pesanan (...);  
CREATE TABLE pembayaran (...);  
CREATE VIEW view_transaksi_lengkap;
```

- b. DML (Data Manipulation Language) digunakan untuk manipulasi data (Insert, Update, Delete):

INSERT:

```
user → sp_tambah_user  
produk → sp_tambah_produk  
kategori → sp_tambah_kategori  
pesanan → sp_buat_pesanan  
pembayaran → sp_tambah_pembayaran
```

UPDATE:

```
pesanan.status → sp_upstatus_pesanan  
produk.stok → sp_upstok_produk
```

DELETE:

```
produk → sp_hapus_produk  
pesanan → sp_hapus_pesanan
```

- c. DQL (Data Query Language) Digunakan untuk menampilkan laporan atau pencarian data:

```

sp_lap_penjualan(p_tanggal)
sp_riwayat_pesanan(p_id_user)
sp_cek_stok_menipis()
SELECT * FROM view_transaksi_lengkap

```

D. Daftar Stored Procedure

No.	Nama Procedure	Jenis	Parameter	Fungsi
1	sp_tambah_user	DML	p_nama, p_email, p_role	Menambah pengguna baru
2	sp_tambah_kategori	DML	p_nama_kategori, p_deskripsi	Menambah kategori baru
3	sp_tambah_produk	DML	p_user_id, p_kategori_id, p_nama_produk, p_harga, p_stok	Menambah produk baru
4	sp_buat_pesanan	DML	p_user_id, p_produk_id, p_jumlah, p_total_harga	Membuat pesanan baru
5	sp_tambah_pembayaran	DML	p_pesanan_id, p_metode, p_harga	Menyimpan data pembayaran
6	sp_upstatus_pembayaran	DML	p_pesana_id, p_status	Mengubah status pesanan
7	sp_upstok_produk	DML	p_produk_id, p_stok_baru	Memperbarui stok produk

8	sp_hapus_pesanan	DML	p_pesanan_id	Menghapus pesanan
9	sp_hapus_produk	DML	p_produk_id	Menghapus produk
10	sp_lap_penjualan	DQL	p_tanggal	Menampilkan laporan penjualan
11	sp_riwayat_pesanan	DQL	p_id_user	Menampilkan riwayat transaksi pembeli
12	sp_cek_stok	DQL	-	Menampilkan daftar dengan stok < 10

E. Hasil Implementasi & Output

No.	Nama Store Procedure	Contoh Perintah (CALL)	Contoh Hasil Output
1	sp_tambah_user	CALL sp_tambah_user('Lala', 'lala@gmail.com', 'Pembeli');	id_user_baru = 6
2	sp_tambah_kategori	CALL sp_tambah_kategori ('Make Up', 'Alat kecantikan lipstik, bedak, eyeshadow');	id_kategori_baru = 3
3	sp_tambah_produk	CALL sp_tambah_produk ('3', '4', 'Liptint Dior', '500000', '20');	id_produk_baru = 7
4	sp_buat_pesanan	CALL sp_buat_pesanan ('6', '6', '1');	id_pesanan_baru = 5

5	sp_tambah_pembayaran	CALL sp_tambah_pembayaran ('4', 'Transfer Bank', '500000');	id_pembayaran_baru = 2
6	sp_upstatus_pesanan	CALL sp_upstatus_pesanan (1, 'Dikirim');	“Status Updated”
7	sp_upstok_pesanan	CALL sp_upstok_produk (6, '19');	“Stok Updated”
8	sp_hapus_produk	CALL sp_hapus_produk(6);	“Produk berhasil dihapus (jika tidak terkait pesanan)
9	sp_hapus_pesanan	CALL sp_hapus_pesanan(4);	“Pesanan dan pembayaran berhasil dihapus”
10	sp_lap_penjualan	CALL sp_lap_penjualan(CURDAT E());	daftar produk + total_penjualan
11	sp_riwayat_pesanan	CALL sp_riwayat_pesanan(2);	daftar seluruh pesanan user id = 2
12	sp_cek_stok	CALL sp_cek_stok();	daftar produk stok < 10