

KASUS: SISTEM PEMESANAN HOTEL

Deskripsi Kasus

Sebuah hotel ingin mengembangkan sistem pemesanan yang memungkinkan pelanggan untuk memesan kamar, melihat ketersediaan kamar, dan mengelola pemesanan mereka secara online. Sistem ini juga memungkinkan admin hotel untuk mengelola data kamar, pemesanan, dan pembayaran.

2. Penyelesaian:

A. Data Flow Diagram (DFD)

Level 0 DFD:

- **Entitas Eksternal:**
 - **Pelanggan:** Mencari kamar, memesan kamar, membayar tagihan.
 - **Admin Hotel:** Mengelola data kamar, pemesanan, dan pembayaran.

Aliran Data Utama:

1. Pelanggan → Sistem Pemesanan Hotel (memasukkan informasi pencarian & pemesanan kamar).
2. Admin Hotel → Sistem Pemesanan Hotel (mengelola data kamar & melihat pemesanan).
3. Pelanggan → Sistem Pemesanan Hotel (melakukan pembayaran).

Level 1 DFD (Detail):

- **Proses utama:**
 1. **Pencarian Kamar:** Menampilkan kamar yang tersedia berdasarkan tanggal yang diinginkan.
 2. **Pemesanan Kamar:** Pelanggan memilih kamar dan memasukkan detail pemesanan.
 3. **Pembayaran:** Memproses pembayaran dari pelanggan.
 4. **Manajemen Kamar:** Admin mengelola detail kamar (misalnya, ketersediaan, harga).
 5. **Manajemen Pemesanan:** Admin mengelola pemesanan yang masuk.

B. Entity Relationship Diagram (ERD)

Entitas:

- **Pelanggan (Customer):** customer_id, name, contact_info.
- **Kamar (Room):** room_id, room_type, price, availability.
- **Pemesanan (Booking):** booking_id, customer_id, room_id, check_in_date, check_out_date.
- **Pembayaran (Payment):** payment_id, booking_id, payment_date, amount_paid.

Relasi:

1. Satu pelanggan dapat membuat banyak pemesanan.
2. Setiap pemesanan terkait dengan satu kamar.

3. Setiap pemesanan memiliki satu pembayaran.

C. UML Diagram

1. **Class Diagram:** Menggambarkan hubungan antar kelas:

- **Class Customer:** customer_id, name, contact_info.
- **Class Room:** room_id, room_type, price, availability.
- **Class Booking:** booking_id, check_in_date, check_out_date.
- **Class Payment:** payment_id, payment_date, amount_paid.

Hubungan:

- Pelanggan dapat melakukan banyak pemesanan (one-to-many).
- Setiap pemesanan terkait dengan satu kamar (many-to-one).
- Setiap pemesanan memiliki satu pembayaran (one-to-one).

2. **Use Case Diagram:** Menggambarkan aktor dan interaksi mereka dengan sistem:

Aktor:

- **Pelanggan:** Melakukan pencarian kamar, memesan kamar, dan membayar.
- **Admin Hotel:** Mengelola data kamar dan pemesanan.

Use Case:

- **Pelanggan:** Pencarian kamar, pemesanan kamar, melakukan pembayaran.
- **Admin Hotel:** Mengelola kamar, mengelola pemesanan, memverifikasi pembayaran.

3. Visualisasi di Draw.io:

1. **DFD:**

- Gunakan elemen **Process**, **Data Store**, dan **External Entity** untuk menggambarkan aliran data antara pelanggan, sistem, dan admin hotel.

2. **ERD:**

- Gunakan elemen **Entity** dengan atribut-atribut utama untuk menggambarkan entitas pelanggan, kamar, pemesanan, dan pembayaran.

3. **UML Diagram:**

- Gunakan elemen **Class**, **Use Case**, dan **Actor** untuk menggambarkan struktur dan interaksi dalam sistem.

Anda dapat mengembangkan diagram ini lebih lanjut di **Draw.io** atau meminta bantuan tambahan untuk menggambarannya lebih detail.