

**LAPORAN TUGAS BESAR RESERVASI HOTEL  
MATA KULIAH PERANCANGAN BASIS DATA  
TAHUN 2023/2024**



**DISUSUN OLEH :**

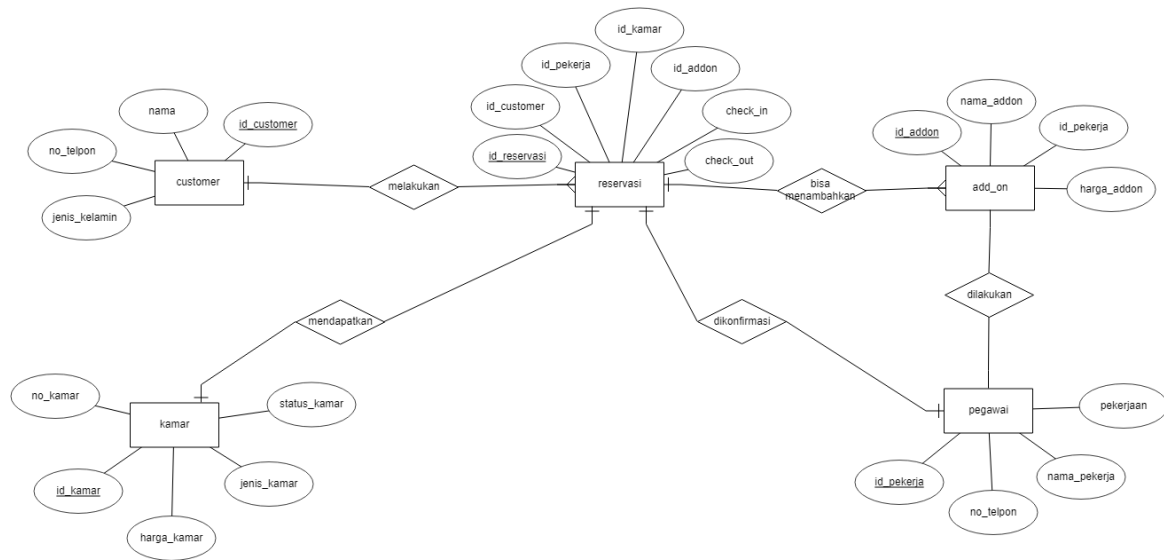
- 1. I Gede Arengga Nabakta (1203220108)**
- 2. Maria Angela Permatasari (1203220092)**
- 3. Asyam Zufar Ahmad (1203220093)**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN  
BISNIS INSTITUT TEKNOLOGI TELKOM  
SURABAYA 202**

## DAFTAR ISI

DAFTAR ISI.....	i
Entity Relationship Diagram.....	1
Skema Tabel.....	2
Query tabel .....	3
1. Query Database.....	3
2. Query Create Table .....	3
3. Insert Data.....	6
a. Insert Pegawai.....	6
c. Insert Kamar.....	7
4. Prosedurs.....	8
<b>5. View</b> .....	12
6. Event .....	13
7. Trigger.....	14
8. Call Prosedur.....	15
Lampiran.....	17
Link ERD dan Skema tabel.....	21
List Jobdesk .....	21

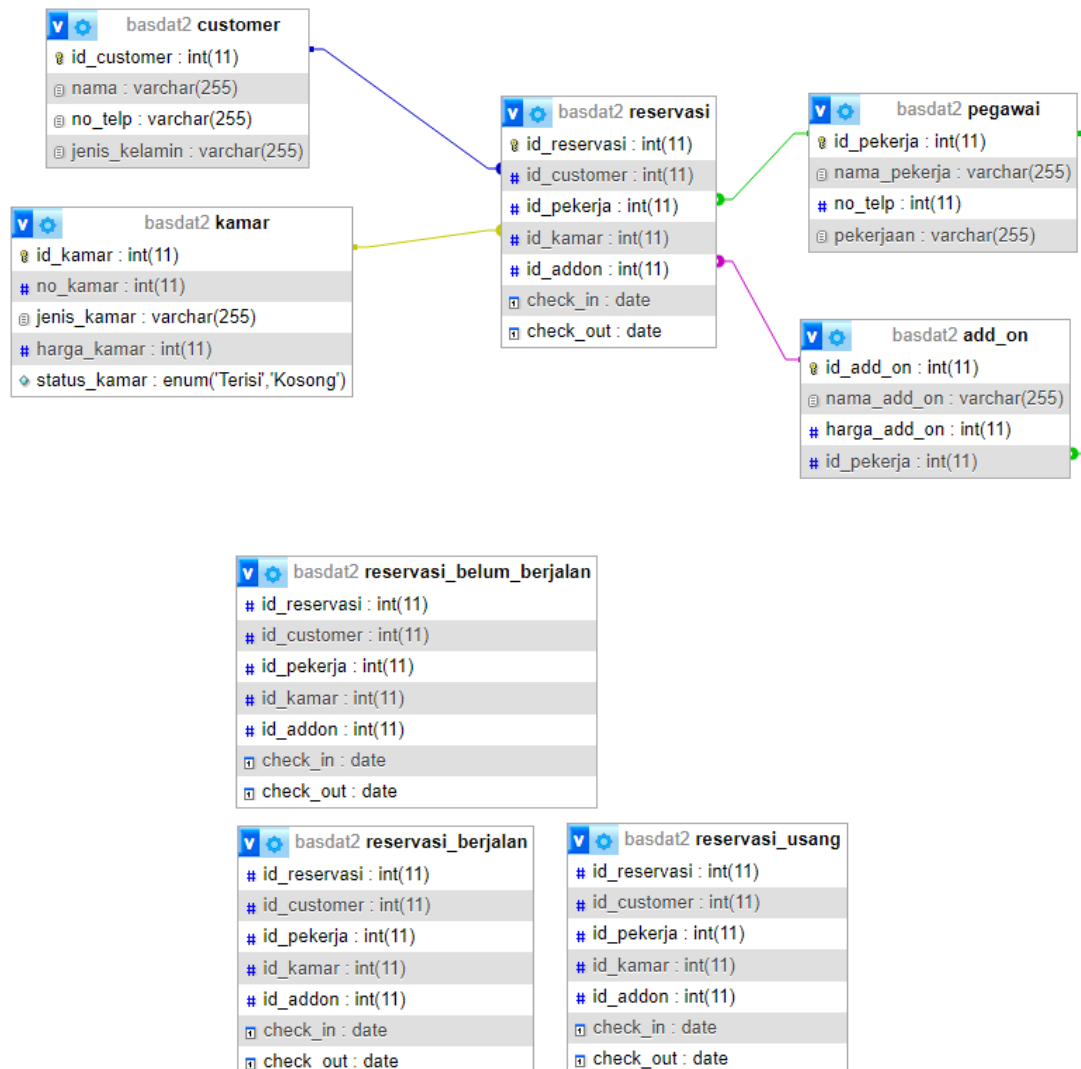
# Entity Relationship Diagram



## Penjelasan :

Tabel diatas memiliki 5 buah entitas dimana terdapat entitas kamar, entitas reservasi, entitas add\_on, entitas pegawai, dan entitas customer yang memiliki banyak atribut untuk melengkapi informasi dari sebuah tabel.

## Skema Tabel



### Penjelasan :

Berikut adalah skema yang dihasilkan pada program reservasi hotel. Terdapat Forrent key yang saling berhubungan agar data dapat bersinambungan dengan baik. Terdapat juga 3 buah tabel yang dihasilkan dari program “View” agar bisa menampilkan data pada kondisi tertentu

## Query tabel

Untuk mendapatkan program diatas dibutuhkan beberapa query untuk tempat penyimpanna data yang akan didapatkan. Berikut adalah beberapa query yang digunakan.

### 1. Query Database

```
CREATE DATABASE tubes_basdat;
```

**Penjelasan :** Query “CREATE DATABASE” digunakan untuk membuat sebuah database baru dengan nama “tubes \_basdat”.

### 2. Query Create Table

```
CREATE TABLE pegawai(  
  id_pekerja INT(11) PRIMARY KEY AUTO_INCREMENT,  
  nama_pekerja VARCHAR(255),  
  no_telp INT(11),  
  pekerjaan VARCHAR(255)  
);
```

**Penjelasan :**

Query diatas adalah query yang digunakan untuk membuat table baru dengan nama ‘pegawai’ yang memiliki beberapa atribut. Atribut yang digunakan adalah

1. id\_pekerja yang merupakan kolom dengan tipe data Integer dengan length 11. yang bekerja sebagai primary key yang akan otomatis bertambah nilainya setiap ada data baru yang dimasukan ke dalam tabel ini.
2. nama\_pekerja merupakan kolom dengan tipe data VARCHAR dengan length maskimal 255 karakter untuk menyimpan data nama pekerja.
3. no\_telp merupakan kolom dengan tipe data integer dengan maksimal 11 karakter. yang digunakan untuk menyimpan no telp dari data baru.
4. id\_pekerja merupakan kolom dengan tipe data VARCHAR dengan maksimal string 255 karakter yang digunakan untuk menyimpan pekerjaan seseorang

```
CREATE TABLE kamar(  
  id_kamar INT(11) PRIMARY KEY AUTO_INCREMENT,  
  no_kamar INT(11),  
  jenis_kamar VARCHAR(255),  
  harga_kamar INT(11),  
  status_kamar ENUM('Terisi', 'Kosong') DEFAULT 'Kosong'  
);
```

**Penjelasan :**

Query diatas digunakan untuk membuat tabel baru dengan 5 kolom yang memiliki fungsi sebagai berikut

1. id\_kamar dengan tipe data Integer dengan maks 11 char yang di set sebagai primary key yang disandingkan dengan auto increment yang digunakan untuk menambahkan nilai secara otomatis ketika ada inputan data baru.
2. no\_kamar dengan tipe data Integer dengan maks 11 char yang digunakan untuk menyimpan nomor dari sebuah kamar.
3. jenis\_kamar dengan tipe data varchar yang digunakan untuk menyimpan nama dari sebuah jenis kamar

4. harga\_kamar dengan tipe data Integer dengan maks 11 char yang digunakan untuk menyimpan harga dari sebuah jenis kamar.
5. status\_kamar dengan tipe ENUM yang memiliki nilai “Terisi” dan “Kosong” sebagai informasi tambahan untuk status dari sebuah kamar.

```
CREATE TABLE customer(
  id_customer INT(11) PRIMARY KEY AUTO_INCREMENT,
  nama VARCHAR(255),
  no_telp VARCHAR(255),
  jenis_kelamin VARCHAR(255)
);
```

**Penjelasan :** Query diatas digunakan untuk membuat tabel customer dengan 4 kolom yang berfungsi untuk menyimpan beberapa informasi dengan tipe data VARCHAR dengan maksimal 255 string seperti

1. id\_customer yang digunakan untuk menyimpan id dari seorang customer yang dimana memiliki fungsi tambahan yang memberikan nilai secara otomatis setiap data baru yang ditambahkan
2. nama yang digunakan untuk menyimpan nama dari customer untuk setiap datanya.
3. no\_telp yang digunakan untuk menyimpan nomor telephone untuk setiap datanya.
4. jenis kelamin digunakan untuk menyimpan jenis kelamin dari seseorang untuk tiap data pada tabel tersebut.

```
CREATE TABLE add_on (
  id_add_on INT(11) PRIMARY KEY AUTO_INCREMENT,
  nama_add_on VARCHAR(255),
  harga_add_on INT(11),
  id_pekerja INT(11),
  FOREIGN KEY (id_pekerja) REFERENCES pegawai(id_pekerja)
);
```

**Penjelasan :**

Tabel diatas digunakan untuk menambahkan tabel baru dengan nama “add\_on” dengan 4 attribut dengan fungsi yang berbeda beda yang ditambahkan dengan foreign key pada id\_pekerja.

1. id\_add\_on digunakan untuk menyimpan nilai dari add\_on yang bertambah secara otomatis seiring dengan bertambahnya data yang di set sebagai primary key dari tabel add\_on.
2. nama\_add\_on yang berfungsi untuk menyimpan nilai dari sebuah add\_on dengan tipe data varchar dengan maks 255 string karakter.
3. harga\_add\_on yang berfungsi untuk menyimpan data harga add on dengan tipe data Integer dengan maks karakter sebanyak 11 buah.
4. id\_pekerja adalah kolom yang menyimpan nilai dari sebuah id pekerja dengan tipe data integer dengan maks 11 char

FOREIGN KEY (id\_pekerja) REFERENCES pegawai(id\_pekerja): Ini adalah klausa yang menetapkan foreign key pada kolom id\_pekerja dalam tabel "add\_on" yang merujuk ke primary key (id\_pekerja) dalam tabel "pegawai". Dengan cara ini, hubungan antara kedua tabel ini terjalin.

```

CREATE TABLE reservasi (
  id_reservasi INT(11) PRIMARY KEY AUTO_INCREMENT,
  id_customer INT(11),
  id_pekerja INT(11),
  id_kamar INT(11),
  id_addon INT(11),
  check_in DATE,
  check_out DATE,
  FOREIGN KEY (id_customer) REFERENCES customer(id_customer),
  FOREIGN KEY (id_pekerja) REFERENCES pegawai(id_pekerja),
  FOREIGN KEY (id_kamar) REFERENCES kamar(id_kamar),
  FOREIGN KEY (id_addon) REFERENCES add_on(id_add_on)
);

```

**Penjelasan :**

merupakan perintah untuk membuat tabel bernama "reservasi" dalam sebuah sistem manajemen reservasi. Tabel ini dirancang untuk menyimpan informasi mengenai reservasi kamar hotel atau akomodasi serupa. Berikut penjelasan singkat mengenai struktur tabel:

1. id\_reservasi (INT(11) PRIMARY KEY AUTO\_INCREMENT): Kolom ini merupakan identifikasi unik untuk setiap reservasi dan diatur sebagai kunci utama (PRIMARY KEY) yang otomatis bertambah (AUTO\_INCREMENT) setiap kali ada entri baru.
2. id\_customer (INT(11)): Kolom ini menyimpan identifikasi pelanggan yang melakukan reservasi. Terhubung ke tabel "customer" melalui kunci asing (FOREIGN KEY).
3. id\_pekerja (INT(11)): Kolom ini menyimpan identifikasi pekerja atau pegawai yang menangani reservasi. Terhubung ke tabel "pegawai" melalui kunci asing (FOREIGN KEY).
4. id\_kamar (INT(11)): Kolom ini menyimpan identifikasi kamar yang dipesan. Terhubung ke tabel "kamar" melalui kunci asing (FOREIGN KEY).
5. id\_addon (INT(11)): Kolom ini menyimpan identifikasi tambahan (addon) yang mungkin dipilih oleh pelanggan selama reservasi. Terhubung ke tabel "add\_on" melalui kunci asing (FOREIGN KEY).
6. check\_in (DATE): Kolom ini menyimpan tanggal check-in, yaitu tanggal ketika pelanggan diharapkan tiba dan menggunakan kamar.
7. check\_out (DATE): Kolom ini menyimpan tanggal check-out, yaitu tanggal ketika pelanggan diharapkan meninggalkan kamar setelah selesai masa menginap.
8. FOREIGN KEY (id\_customer) REFERENCES customer(id\_customer): Kunci asing yang menghubungkan kolom "id\_customer" dengan kolom "id\_customer" dalam tabel "customer".
9. FOREIGN KEY (id\_pekerja) REFERENCES pegawai(id\_pekerja): Kunci asing yang menghubungkan kolom "id\_pekerja" dengan kolom "id\_pekerja" dalam tabel "pegawai".
10. FOREIGN KEY (id\_kamar) REFERENCES kamar(id\_kamar): Kunci asing yang menghubungkan kolom "id\_kamar" dengan kolom "id\_kamar" dalam tabel "kamar".
11. FOREIGN KEY (id\_addon) REFERENCES add\_on(id\_add\_on): Kunci asing yang menghubungkan kolom "id\_addon" dengan kolom "id\_addon" dalam tabel "add\_on".

### 3. Insert Data

#### a. Insert Pegawai

```
INSERT INTO pegawai (nama_pegawai, no_telp, pekerjaan) VALUES  
( 'tita', 1789620867, 'Resepsionis'),  
( 'ina', 2029362580, 'Resepsionis'),  
( 'sinta', 6582802421, 'Resepsionis'),  
( 'lulu', 9632158015, 'Spa Massager'),  
( 'anggie', 3580963285, 'Cleaning Service'),  
( 'frans', 2580963542, 'Cleaning Service'),  
( 'ken', 2185965846, 'Satpam'),  
( 'akbar', 1265870239, 'Driver'),  
( 'subhan', 3258965476, 'Chef'),  
( 'bagas', 1258462390, 'Manager');
```

#### Penjelasan :

Contoh baris data: ('tita', 1789620867, 'Resepsionis') menyatakan bahwa ada seorang pegawai dengan nama 'tita', nomor telepon '1789620867', dan bekerja sebagai 'Resepsionis'. Demikian pula, baris-baris lainnya mengandung informasi yang sama untuk setiap pegawai yang ditambahkan ke dalam tabel 'pegawai'.

#### b. Insert Add On

```
INSERT INTO add_on (id_add_on, nama_add_on, harga_add_on, id_pegawai) VALUES  
(1, 'Wifi', 50000, 1),  
(2, 'Breakfast', 50000, 9),  
(3, 'Spa Service', 100000, 4),  
(4, 'vallet', 50000, 7),  
(5, 'Gym Access', 80000, 3),  
(6, 'Airport Shuttle', 120000, 8),  
(7, 'Mini Bar', 60000, 3),  
(8, 'Laundry Service', 90000, 5),  
(9, 'Extra Pillow', 150000, 6),  
(10, 'Room Service', 70000, 10);
```

#### Penjelasan :

INSERT INTO add\_on: Menunjukkan bahwa data akan dimasukkan ke dalam tabel bernama add\_on. (id\_add\_on, nama\_add\_on, harga\_add\_on, id\_pegawai) VALUES: Menyatakan kolom-kolom yang akan diisi dengan nilai-nilai baru. Kemudian, setiap baris di dalam tanda kurung () menambahkan sebuah entri baru ke dalam tabel:

1. id\_add\_on: Kolom yang berisi ID untuk setiap add-on.
2. nama\_add\_on: Kolom yang berisi nama dari setiap add-on.
3. harga\_add\_on: Kolom yang berisi harga dari setiap add-on.
4. id\_pegawai: Kolom yang berisi ID pegawai yang terkait dengan setiap add-on.



### c. Insert Kamar

```
INSERT INTO kamar (id_kamar, no_kamar, jenis_kamar, harga_kamar)
VALUES
(1, '101', 'Single', 100000),
(2, '102', 'Single', 100000),
(3, '103', 'Single', 100000),
(4, '104', 'Single', 100000),
(5, '105', 'Single', 100000),
(6, '201', 'Double', 250000),
(7, '202', 'Double', 250000),
(8, '203', 'Double', 250000),
(9, '204', 'Double', 250000),
(10, '205', 'Double', 250000),
(11, '301', 'Deluxe', 400000),
(12, '302', 'Deluxe', 400000),
(13, '303', 'Deluxe', 400000),
(14, '304', 'Deluxe', 400000),
(15, '305', 'Deluxe', 400000);
```

#### **Penjelasan :**

INSERT INTO kamar: Menunjukkan bahwa data akan dimasukkan ke dalam tabel bernama kamar. (id\_kamar, no\_kamar, jenis\_kamar, harga\_kamar): Menyatakan kolom-kolom yang akan diisi dengan nilai. VALUES: Kata kunci yang mengawali baris-baris nilai yang akan dimasukkan ke dalam tabel. (1, '101', 'Single', 100000), ... (15, '305', 'Deluxe', 400000): Baris-baris ini merupakan nilai-nilai yang akan dimasukkan. Setiap baris mewakili satu baris data untuk dimasukkan ke dalam tabel kamar dengan nilai untuk masing-masing kolom yang telah dijelaskan sebelumnya.

#### 4. Prosedurs

- a. Menghitung harga berdasarkan id\_reservasi (calculateHarga)

```
DELIMITER //

CREATE PROCEDURE tampilkan_total_harga(IN id_reservasi_param INT)
BEGIN
    SELECT
        c.nama AS nama_customer,
        k.jenis_kamar,
        k.harga_kamar,
        COALESCE(SUM(a.harga_add_on), 0) AS total_add_on,
        k.harga_kamar + COALESCE(SUM(a.harga_add_on), 0) AS total_harga
    FROM
        customer c
    JOIN
        reservasi r ON c.id_customer = r.id_customer
    JOIN
        kamar k ON r.id_kamar = k.id_kamar
    LEFT JOIN
        add_on a ON r.id_addon = a.id_add_on
    WHERE
        r.id_reservasi = id_reservasi_param
    GROUP BY
        c.nama, k.jenis_kamar, k.harga_kamar;
END //

DELIMITER ;
```

#### Penjelasan :

Membuat sebuah stored procedure (prosedur tersimpan) dengan nama GetAvailableRooms. Prosedur ini menerima dua parameter input, yaitu checkin\_date dan checkout\_date. Tujuan dari prosedur ini adalah untuk mengembalikan daftar kamar yang tersedia pada rentang tanggal tertentu.

Prosedur ini menggunakan perintah SQL untuk memilih informasi kamar dari tabel kamar yang tidak memiliki konflik jadwal dengan reservasi yang sudah ada. Konflik jadwal diidentifikasi dengan membandingkan tanggal checkin\_date dan checkout\_date dengan rentang tanggal (check\_in dan check\_out) dari setiap reservasi yang sudah ada dalam tabel reservasi.

Jika suatu kamar tidak muncul dalam hasil subquery (sub-pernyataan SELECT di dalam WHERE clause), berarti kamar tersebut tersedia pada rentang tanggal yang diminta. Hasil dari prosedur ini berupa daftar kamar yang tersedia dengan informasi seperti id\_kamar, no\_kamar, jenis\_kamar, dan harga\_kamar.

b. Query Melihat kamar kosong berdasarkan tanggal

```
DELIMITER //

CREATE PROCEDURE GetAvailableRooms(
    IN checkin_date DATE,
    IN checkout_date DATE
)
BEGIN
    SELECT k.id_kamar, k.no_kamar, k.jenis_kamar, k.harga_kamar
    FROM kamar k
    WHERE k.id_kamar NOT IN (
        SELECT r.id_kamar
        FROM reservasi r
        WHERE (checkin_date BETWEEN r.check_in AND r.check_out
            OR checkout_date BETWEEN r.check_in AND r.check_out
            OR r.check_in BETWEEN checkin_date AND checkout_date
            OR r.check_out BETWEEN checkin_date AND checkout_date)
    );
END //
```

Penjelasan :

Membuat sebuah stored procedure (prosedur tersimpan) dengan nama GetAvailableRooms. Prosedur ini menerima dua parameter input, yaitu checkin\_date dan checkout\_date. Tujuan dari prosedur ini adalah untuk mengembalikan daftar kamar yang tersedia pada rentang tanggal tertentu.

Prosedur ini menggunakan perintah SQL untuk memilih informasi kamar dari tabel kamar yang tidak memiliki konflik jadwal dengan reservasi yang sudah ada. Konflik jadwal diidentifikasi dengan membandingkan tanggal checkin\_date dan checkout\_date dengan rentang tanggal (check\_in dan check\_out) dari setiap reservasi yang sudah ada dalam tabel reservasi.

Jika suatu kamar tidak muncul dalam hasil subquery (sub-pernyataan SELECT di dalam WHERE clause), berarti kamar tersebut tersedia pada rentang tanggal yang diminta. Hasil dari prosedur ini berupa daftar kamar yang tersedia dengan informasi seperti id\_kamar, no\_kamar, jenis\_kamar, dan harga\_kamar.

c. Query Input Customer

```
DELIMITER //

CREATE PROCEDURE input_customer_baru(
    IN nama_customer VARCHAR(255),
    IN no_telp_customer VARCHAR(255),
    IN jenis_kelamin_customer VARCHAR(255)
)
BEGIN
    INSERT INTO customer(nama, no_telp, jenis_kelamin)
    VALUES (nama_customer, no_telp_customer, jenis_kelamin_customer);
END //

DELIMITER ;
```

### Penjelasan :

procedure ini bernama "input\_customer\_baru" dan menerima tiga parameter input, yaitu nama\_customer (string), no\_telp\_customer (string), dan jenis\_kelamin\_customer (string). Tujuan dari stored procedure ini adalah untuk menyisipkan data pelanggan baru ke dalam tabel "customer" dengan nilai yang diteruskan melalui parameter input. Bagian "INSERT INTO" digunakan untuk menambahkan data baru ke dalam tabel dengan nilai yang sesuai. Penggunaan DELIMITER "/" dan "DELIMITER ;" mengatur delimiter yang digunakan dalam prosedur tersebut. Overall, stored procedure ini mempermudah penggunaan SQL dengan memisahkan logika pengelolaan data ke dalam satu blok yang dapat dipanggil dengan mudah.

#### d. Query input reservasi baru

```
DELIMITER //

CREATE PROCEDURE inputReservasi(
    IN id_customer_param INT,
    IN id_kamar_param INT,
    IN id_addon_param INT,
    IN check_in_param DATE,
    IN check_out_param DATE
)
BEGIN
    DECLARE id_pekerja_value INT;

    SET id_pekerja_value = FLOOR(1 + (RAND() * 3));

    IF (SELECT status_kamar FROM kamar WHERE id_kamar = id_kamar_param) = 'Terisi'
THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Maaf kamar sudah di Booking';
    ELSE
        -- Tambahkan reservasi jika kamar belum di-booking
        INSERT INTO reservasi (
            id_customer,
            id_pekerja,
            id_kamar,
            id_addon,
            check_in,
            check_out
        ) VALUES (
            id_customer_param,
            id_pekerja_value,
            id_kamar_param,
            id_addon_param,
            check_in_param,
            check_out_param
        );
    END IF;
END //
DELIMITER ;
```

**Penjelasan :**

Prosedur ini memiliki lima parameter input: `id_customer_param`, `id_kamar_param`, `id_addon_param`, `check_in_param`, dan `check_out_param`, yang masing-masing merepresentasikan informasi yang diperlukan untuk membuat reservasi (seperti ID pelanggan, ID kamar, ID tambahan, tanggal check-in, dan tanggal check-out).

Di dalamnya, terdapat deklarasi variabel `id_pekerja_value` yang menampung nilai ID pekerja yang dipilih secara acak dari 1 hingga 3 dengan fungsi `RAND()`.

Terdapat penggunaan kondisi IF untuk memeriksa status kamar dengan ID yang diberikan. Jika status kamar tersebut sudah 'Terisi', akan dihasilkan sebuah pesan kesalahan menggunakan `SIGNAL SQLSTATE` yang mengindikasikan bahwa kamar sudah di-booking.

Jika status kamar tidak 'Terisi', maka data reservasi baru akan dimasukkan ke dalam tabel reservasi menggunakan pernyataan `INSERT INTO`

## 5. View

### a. Query reservasi yang akan berjalan( reservasi\_belum\_berjalan ).

```
CREATE VIEW reservasi_belum_berjalan AS  
SELECT * FROM reservasi WHERE CURRENT_DATE < check_in;
```

#### Penjelasan :

Reservasi\_belum\_berjalan. View ini akan menampilkan semua data dari tabel reservasi dimana tanggal saat ini (CURRENT\_DATE) lebih kecil dari tanggal check\_in.

### b. Query reservasi yang sedang berlangsung( reservasi\_berjalan ).

```
CREATE VIEW reservasi_berjalan AS  
SELECT * FROM reservasi WHERE CURRENT_DATE BETWEEN check_in AND check_out;
```

#### Penjelasan :

Reservasi\_berjalan. View ini dibuat berdasarkan data dari tabel reservasi. View tersebut akan menampilkan semua kolom yang ada di tabel reservasi untuk baris-baris di mana tanggal saat ini (CURRENT\_DATE) berada di antara tanggal check\_in dan check\_out. Dengan kata lain, view ini menyajikan informasi reservasi yang sedang berlangsung pada hari ini.

### c. Query reservasi yang sudah lewat ( reservasi\_usang )

```
CREATE VIEW reservasi_usang AS  
SELECT * FROM reservasi WHERE CURRENT_DATE > check_out;
```

#### Penjelasan :

Reservasi usang adalah view yang digunakan untuk menampilkan seluruh Riwayat reservasi yang sudah pernah terjadi. Semua di simpan berdasarkan id\_reservasi sehingga gampang dilihat untuk informasi lainnya. Query ini menerapkan prinsip jika tanggal hari ini (CURRENT\_DATE) sudah berada/ lebih besar dari tanggal check\_out reservasi, maka reservasi tersebut tidak masuk kedalam reservasi yang sedang berjalan

## 6. Event

- a. Query mengosongkan kamar ( set\_kamar\_kosong )

```
DELIMITER //

CREATE EVENT set_kamar_kosong
ON SCHEDULE EVERY 1 DAY
DO
BEGIN
    UPDATE kamar
    SET status_kamar = 'Kosong'
    WHERE id_kamar IN (
        SELECT r.id_kamar
        FROM reservasi r
        WHERE CURRENT_DATE > r.check_out
    );
END //

DELIMITER ;
```

### Penjelasan :

Dilakukan pembaruan (UPDATE) pada tabel kamar. Baris-baris yang akan diperbarui memiliki kondisi status\_kamar akan diubah menjadi 'Kosong'. Kondisi untuk pembaruan adalah pada baris-baris di tabel kamar dimana id\_kamar terdapat dalam hasil subquery yang mengambil id\_kamar dari tabel reservasi dimana CURRENT\_DATE (tanggal saat ini) lebih besar dari tanggal check\_out pada tabel reservasi.

## 7. Trigger

```
DELIMITER //

CREATE TRIGGER set_kamar_terisi
AFTER INSERT ON reservasi
FOR EACH ROW
BEGIN
    DECLARE today DATE;
    SET today = CURRENT_DATE;

    IF (today BETWEEN NEW.check_in AND NEW.check_out) THEN
        UPDATE kamar
        SET status_kamar = 'Terisi'
        WHERE id_kamar = NEW.id_kamar;
    END IF;
END //

DELIMITER ;
```

### Penjelasan :

membuat trigger (pemicu) database yang disebut set\_kamar\_terisi. Trigger ini diaktifkan setelah suatu data baru diinsert ke dalam tabel reservasi. Trigger ini berlaku untuk setiap baris yang baru dimasukkan (FOR EACH ROW).

Pada saat trigger dijalankan, sebuah variabel lokal today dideklarasikan dan diatur nilainya sebagai tanggal saat ini (CURRENT\_DATE). Selanjutnya, trigger melakukan pengecekan kondisi dengan membandingkan tanggal saat ini dengan rentang tanggal (check\_in dan check\_out) dari data baru yang dimasukkan (NEW.check\_in dan NEW.check\_out).

Jika tanggal saat ini berada di antara rentang tanggal reservasi baru, maka trigger akan melakukan perintah UPDATE pada tabel kamar. Status kamar (status\_kamar) pada kamar yang sesuai dengan id\_kamar dari data reservasi baru akan diubah menjadi 'Terisi'.

Dengan demikian, tujuan trigger ini adalah untuk secara otomatis mengubah status kamar menjadi 'Terisi' ketika terjadi penambahan data reservasi dengan rentang tanggal yang bersinggungan dengan tanggal saat ini.



## 8. Call Prosedur

### a. Mengisi data Reservasi ( inputReservasi )

```
CALL inputReservasi(1, 1, 3, '2023-01-01', '2023-02-01');
CALL inputReservasi(2, 2, 6, '2023-01-05', '2023-02-05');
CALL inputReservasi(3, 3, 7, '2023-01-10', '2023-02-10');
CALL inputReservasi(4, 4, 9, '2023-01-15', '2023-02-15');
CALL inputReservasi(5, 5, 2, '2023-01-20', '2023-02-20');
CALL inputReservasi(6, 6, 4, '2023-01-25', '2023-02-25');
CALL inputReservasi(7, 7, 3, '2023-03-01', '2023-04-01');
CALL inputReservasi(8, 8, 4, '2023-03-05', '2023-04-05');
CALL inputReservasi(9, 9, 10, '2023-03-10', '2023-04-10');
CALL inputReservasi(10, 10, 5, '2023-03-15', '2023-04-15');
CALL inputReservasi(11, 11, 10, '2023-03-20', '2023-04-20');
CALL inputReservasi(12, 12, 5, '2023-03-25', '2023-04-25');
CALL inputReservasi(13, 13, 7, '2023-06-01', '2023-07-01');
CALL inputReservasi(14, 14, 10, '2023-06-05', '2023-07-05');
CALL inputReservasi(15, 15, 3, '2023-06-05', '2023-07-10');
CALL inputReservasi(16, 1, 9, '2023-06-10', '2023-07-15');
CALL inputReservasi(17, 2, 4, '2023-06-15', '2023-07-20');
CALL inputReservasi(18, 3, 6, '2023-06-20', '2023-07-25');
CALL inputReservasi(19, 4, 3, '2023-06-25', '2023-09-01');
CALL inputReservasi(20, 5, 8, '2023-08-01', '2023-09-05');
CALL inputReservasi(21, 6, 6, '2023-08-05', '2023-09-10');
CALL inputReservasi(22, 7, 3, '2023-08-10', '2023-09-15');
CALL inputReservasi(23, 8, 9, '2023-08-15', '2023-09-20');
CALL inputReservasi(24, 9, 2, '2023-08-20', '2023-09-25');
CALL inputReservasi(25, 10, 2, '2023-08-25', '2023-11-01');
CALL inputReservasi(26, 11, 1, '2023-10-01', '2023-11-05');
CALL inputReservasi(27, 12, 5, '2023-10-05', '2023-11-10');
CALL inputReservasi(28, 13, 10, '2023-10-10', '2023-11-15');
CALL inputReservasi(29, 14, 9, '2023-10-15', '2023-11-20');
CALL inputReservasi(30, 15, 6, '2023-10-20', '2023-11-25');
CALL inputReservasi(1, 1, 3, '2023-12-27', '2024-02-01');
CALL inputReservasi(2, 2, 6, '2024-03-05', '2024-04-05');
```

			id_reservasi	id_customer	id_pekerja	id_kamar	id_addon	check_in	check_out
<input type="checkbox"/>	Ubah	Salin	Hapus	1	1	1	1	3	2023-01-01 2023-02-01
<input type="checkbox"/>	Ubah	Salin	Hapus	2	2	3	2	6	2023-01-05 2023-02-05
<input type="checkbox"/>	Ubah	Salin	Hapus	3	3	3	3	7	2023-01-10 2023-02-10
<input type="checkbox"/>	Ubah	Salin	Hapus	4	4	1	4	9	2023-01-15 2023-02-15
<input type="checkbox"/>	Ubah	Salin	Hapus	5	5	2	5	2	2023-01-20 2023-02-20
<input type="checkbox"/>	Ubah	Salin	Hapus	6	6	2	6	4	2023-01-25 2023-02-25
<input type="checkbox"/>	Ubah	Salin	Hapus	7	7	2	7	3	2023-03-01 2023-04-01
<input type="checkbox"/>	Ubah	Salin	Hapus	8	8	1	8	4	2023-03-05 2023-04-05
<input type="checkbox"/>	Ubah	Salin	Hapus	9	9	1	9	10	2023-03-10 2023-04-10

b. Mengisi data Customer ( inputCustomer )

```
CALL input_customer_baru('angela', '08123413880', 'p');
CALL input_customer_baru('enji', '02111039200', 'p');
CALL input_customer_baru('kayla', '2325856933', 'p');
CALL input_customer_baru('windi', '1235698700', 'p');
CALL input_customer_baru('aisyah', '5892369528', 'p');
CALL input_customer_baru('asyam', '9632857421', 'L');
CALL input_customer_baru('rengga', '3366995582', 'L');
CALL input_customer_baru('rafael', '2111039620', 'L');
CALL input_customer_baru('zidan', '2111032712', 'L');
CALL input_customer_baru('lina', '1962582196', 'p');
CALL input_customer_baru('beni', '1962196316', 'L');
CALL input_customer_baru('niko', '2111039658', 'L');
CALL input_customer_baru('sulis', '3692580132', 'p');
CALL input_customer_baru('david', '2580369147', 'L');
CALL input_customer_baru('lutpi', '3695280361', 'L');
CALL input_customer_baru('uni', '2314569870', 'p');
CALL input_customer_baru('dani', '2308199925', 'L');
CALL input_customer_baru('tisa', '2111032065', 'p');
CALL input_customer_baru('clarisa', '9632580741', 'p');
CALL input_customer_baru('deren', '1256789034', 'L');
CALL input_customer_baru('kurnia', '2119635874', 'L');
CALL input_customer_baru('satya', '8756940231', 'L');
CALL input_customer_baru('martin', '6669993325', 'L');
CALL input_customer_baru('dharma', '8976452132', 'L');
CALL input_customer_baru('lita', '3695802147', 'P');
CALL input_customer_baru('dimas', '2111087963', 'L');
CALL input_customer_baru('damar', '1319494673', 'L');
CALL input_customer_baru('kumala', '1236579202', 'p');
CALL input_customer_baru('anggun', '3636965823', 'p');
CALL input_customer_baru('lala', '2187093456', 'p');
```

<div><div><div>←</div><div>T</div><div>→</div></div></div>				id_customer	nama	no_telp	jenis_kelamin
<input type="checkbox"/>				1	angela	08123413880	p
<input type="checkbox"/>				2	enji	02111039200	p
<input type="checkbox"/>				3	kayla	2325856933	p
<input type="checkbox"/>				4	windi	1235698700	p
<input type="checkbox"/>				5	aisyah	5892369528	p
<input type="checkbox"/>				6	asyam	9632857421	L
<input type="checkbox"/>				7	rengga	3366995582	L
<input type="checkbox"/>				8	rafael	2111039620	L
<input type="checkbox"/>				9	zidan	2111032712	L
<input type="checkbox"/>				10	lina	1962582196	p

## Lampiran

### a. Lampiran Add-on

				id_add_on	nama_add_on	harga_add_on	id_pekerja
<input type="checkbox"/>				1	Wifi	50000	1
<input type="checkbox"/>				2	Breakfast	50000	9
<input type="checkbox"/>				3	Spa Service	100000	4
<input type="checkbox"/>				4	vallet	50000	7
<input type="checkbox"/>				5	Gym Access	80000	3
<input type="checkbox"/>				6	Airport Shuttle	120000	8
<input type="checkbox"/>				7	Mini Bar	60000	3
<input type="checkbox"/>				8	Laundry Service	90000	5
<input type="checkbox"/>				9	Extra Pillow	150000	6
<input type="checkbox"/>				10	Room Service	70000	10

### b. Lampiran Customer

				id_customer	nama	no_telp	jenis_kelamin
<input type="checkbox"/>				1	angela	08123413880	p
<input type="checkbox"/>				2	enji	02111039200	p
<input type="checkbox"/>				3	kayla	2325856933	p
<input type="checkbox"/>				4	windi	1235698700	p
<input type="checkbox"/>				5	aisyah	5892369528	p
<input type="checkbox"/>				6	asyam	9632857421	L
<input type="checkbox"/>				7	rengga	3366995582	L
<input type="checkbox"/>				8	rafael	2111039620	L
<input type="checkbox"/>				9	zidan	2111032712	L
<input type="checkbox"/>				10	lina	1962582196	p
<input type="checkbox"/>				11	beni	1962196316	L
<input type="checkbox"/>				12	niko	2111039658	L
<input type="checkbox"/>				13	sulis	3692580132	p
<input type="checkbox"/>				14	david	2580389147	L
<input type="checkbox"/>				15	lutpi	3695280361	L
<input type="checkbox"/>				16	uni	2314569870	p
<input type="checkbox"/>				17	dani	2308199925	L
<input type="checkbox"/>				18	tisa	2111032065	p
<input type="checkbox"/>				19	clarisa	9632580741	p
<input type="checkbox"/>				20	deren	1256789034	L
<input type="checkbox"/>				21	kurnia	2119635874	L
<input type="checkbox"/>				22	satya	8756940231	L
<input type="checkbox"/>				23	martin	6669993325	L
<input type="checkbox"/>				24	dharm	8976452132	L
<input type="checkbox"/>				25	lita	3695802147	P

### c. Lampiran Kamar

																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			<
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---

				id_reservasi	id_customer	id_pekerja	id_kamar	id_addon	check_in	check_out
<input type="checkbox"/>	Ubah	Salin	Hapus	1	1	1	1	3	2023-01-01	2023-02-01
<input type="checkbox"/>	Ubah	Salin	Hapus	2	2	3	2	6	2023-01-05	2023-02-05
<input type="checkbox"/>	Ubah	Salin	Hapus	3	3	3	3	7	2023-01-10	2023-02-10
<input type="checkbox"/>	Ubah	Salin	Hapus	4	4	1	4	9	2023-01-15	2023-02-15
<input type="checkbox"/>	Ubah	Salin	Hapus	5	5	2	5	2	2023-01-20	2023-02-20
<input type="checkbox"/>	Ubah	Salin	Hapus	6	6	2	6	4	2023-01-25	2023-02-25
<input type="checkbox"/>	Ubah	Salin	Hapus	7	7	2	7	3	2023-03-01	2023-04-01
<input type="checkbox"/>	Ubah	Salin	Hapus	8	8	1	8	4	2023-03-05	2023-04-05
<input type="checkbox"/>	Ubah	Salin	Hapus	9	9	1	9	10	2023-03-10	2023-04-10
<input type="checkbox"/>	Ubah	Salin	Hapus	10	10	3	10	5	2023-03-15	2023-04-15
<input type="checkbox"/>	Ubah	Salin	Hapus	11	11	1	11	10	2023-03-20	2023-04-20
<input type="checkbox"/>	Ubah	Salin	Hapus	12	12	2	12	5	2023-03-25	2023-04-25

g. Lampiran reservasi belum berjalan

				id_reservasi	id_customer	id_pekerja	id_kamar	id_addon	check_in	check_out
<input type="checkbox"/>	Ubah	Salin	Hapus	34	2	3	2	6	2024-03-05	2024-04-05

h. Lampiran sedang berjalan

				id_reservasi	id_customer	id_pekerja	id_kamar	id_addon	check_in	check_out
<input type="checkbox"/>	Ubah	Salin	Hapus	33	1	3	1	3	2023-12-27	2024-02-01

i. Lampiran error Input reservasi

**Galat**

Query SQL: [Salin](#)

```
CALL inputReservasi(1, 1, 3, '2023-12-27', '2024-02-01');
```

MySQL menyatakan: 

#1644 - Maaf kamar sudah di Booking

j. Lampiran tampilkan\_total\_harga

nama_customer	jenis_kamar	harga_kamar	total_add_on	total_harga
lina	Double	250000	80000	330000

- k. Lampiran melihat kamar kosong

✓ Menampilkan baris 0 - 11 (total 12, Pencarian dilakukan di: ...)

```
CALL getAvailableRooms('2023-12-1','2024-05-1');
```

☐ Tampilkan semua | Jumlah baris: 25 ▼

Extra options

id_kamar	no_kamar	jenis_kamar	harga_kamar
4	104	Single	100000
5	105	Single	100000
6	201	Double	250000
7	202	Double	250000
8	203	Double	250000
9	204	Double	250000
10	205	Double	250000
11	301	Deluxe	400000
12	302	Deluxe	400000
13	303	Deluxe	400000
14	304	Deluxe	400000
15	305	Deluxe	400000

- l. Lampiran kamar yang sudah berganti dikarenakan Trigger

				id_kamar	no_kamar	jenis_kamar	harga_kamar	status_kamar
<input type="checkbox"/>	Ubah	Salin	Hapus	1	101	Single	100000	Terisi
<input type="checkbox"/>	Ubah	Salin	Hapus	2	102	Single	100000	Kosong
<input type="checkbox"/>	Ubah	Salin	Hapus	3	103	Single	100000	Kosong
<input type="checkbox"/>	Ubah	Salin	Hapus	4	104	Single	100000	Kosong

## Link ERD dan Skema tabel

<https://github.com/Rengga29/Tubes-Basdat-2023/>

## List Jobdesk

List Tugas \ Nama		Arengga	Maria	Asyam
Menyusun ERD		x		x
Menulis ERD		x		x
Membuat data	Customer		x	
	Reservasi	x		
	Kamar			x
	Add_on			x
	Pekerja	x	x	
Membuat Query Data	Customer	x		
	Reservasi	x		
	Kamar			x
	Add_on			x
	Pekerja	x		
Membuat Prosedur	GetAvailableRoom	x		
	Input Reservasi	x		
	Input Customer	x		
	Menampilkan Harga	x		
Membuat View	Reservasi berjalan	x		
	Reservasi belum berjalan	x		
	Reservasi usang	x		
Membuat Trigger	Set kamar terisi	x		
Membuat Event	Set Kamar Kosong	x		
Membuat Laporan	Membuat template Word	x		
	Memasukan tiap Query di tempatnya	x		
Memberikan Penjelasan Query	Query database	x		
	Create table pegawai	x		
	Create table kamar	x		
	Create table Customer	x		
	create table Add_on	x		
	Create table reservasi	x		

	Query insert into	x	x	
	Prosedur Calculate harga		x	
	Prosedur getAvailableRoom			x
	Prosedur inputCustomer		x	
	Prosedur inputReservasi		x	
	View tabel reservasi belum berjalan		x	
	View tabel reservasi berjalan			x
	View tabel reservasi usang		x	
	Query Event set kamar kosong		x	
	Trigger Set kamar terisi			x
	Call Prosedur			x