

**LAPORAN TUGAS BESAR
PROYEK BASIS DATA LANJUT**



Disusun Oleh :

Kelompok 5

Anggota Kelompok

- | | |
|-------------------------|-------------|
| 1. Habib Al Qodri | (G1A023047) |
| 2. Ricardo Gellael | (G1A023061) |
| 3. Juan Agraprana Putra | (G1A023085) |
| 4. Agyl Wendi Pratama | (G1A023087) |
| 5. M. Ryan Alhabsy | (G1A023093) |

Nama Asisten Dosen :

- | | |
|--------------------------------|-------------|
| 1. Merly Yuni Purnama | (G1A022006) |
| 2. Reksi Hendra Pratama | (G1A022032) |
| 3. Sinta Ezra Wati Gulo | (G1A022040) |
| 4. Fadlan Dwi Febrio | (G1A022051) |
| 5. Torang Four Yones Manullang | (G1A022052) |
| 6. Wahyu Ozorah Manurung | (G1A022060) |
| 7. Shalaudin Muhammad Sah | (G1A022070) |
| 8. Dian Ardiyanti Saputri | (G1A022084) |

Dosen Pengampu :

1. Dr. Endina Putri Purwandari, S. T., M.Kom
2. Ir. Tiara Eka Putri, S.T., M.Kom.

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS BENGKULU

2025

KATA PENGANTAR

Puji syukur kami ucapkan kepada Allah Swt. atas nikmat dan petunjuk-Nya yang telah memudahkan kami menyelesaikan tugas besar ini sesuai jadwal. Dokumen ini dibuat sebagai pelaksanaan tugas utama dalam kelas Proyek Basis Data Lanjut.

Tugas ini tidak hanya menguji kemampuan akademis kami, tetapi juga meningkatkan keterampilan dalam kolaborasi, penelitian, dan penyampaian. Kami berharap laporan ini mampu menyajikan pemahaman yang komprehensif dan mendalam tentang isu yang kami bahas, serta menawarkan solusi yang telah kami rancang untuk mengatasi tantangan yang ada. Laporan ini disusun secara sistematis agar pembaca dapat dengan mudah mengikuti alur pemikiran kami.

Kami juga ingin menyampaikan rasa terima kasih yang mendalam kepada asisten dosen dan rekan-rekan yang telah memberikan bimbingan, dukungan, serta masukan yang sangat berharga. Partisipasi mereka memperkaya pembicaraan dan menawarkan sudut pandang tambahan yang sangat berharga. Dengan rasa bangga, kami menyampaikan laporan ini sebagai buah dari perjalanan kami dalam mata kuliah ini yang penuh komitmen dan antusiasme.

Harapan kami, laporan ini bisa memberikan keuntungan dan motivasi untuk setiap pembaca. Kami mengucapkan terima kasih atas peluang yang diberikan, dan kami berharap laporan ini dapat memberikan kontribusi yang berarti untuk pemahaman lebih dalam dalam mata kuliah ini.

Bengkulu, 5 Mei 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	1
DAFTAR ISI.....	2
DAFTAR GAMBAR	3
DAFTAR TABEL.....	4
LANDASAN TEORI.....	5
HASIL DAN PEMBAHASAN.....	8
KESIMPULAN DAN SARAN.....	25
DAFTAR PUSTAKA	26
JOBDESK	27
DOKUMENTASI Pengerjaan.....	27

DAFTAR GAMBAR

Gambar 1. ERD	8
Gambar 4. Source Code Input Data (1).....	11
Gambar 5. Source Code Input Data	12
Gambar 5. Source Code Input Data	13
Gambar 10. Alias, Operator dan Join.....	16
Gambar 11. Alias, Operator dan Join.....	16
Gambar 11. Tabel Output Alias dan Operator	18
Gambar 12. Tabel Output Alias dan Operator	19
Gambar 13. Tabel Output Alias dan Operator	19
Gambar 14. Tabel Output Alias dan Operator	19
Gambar 15. Function.....	19
Gambar 16. Function.....	20
Gambar 17. Nested Query.....	22
Gambar 18. Tabel Output Nested Query.....	23
Gambar 19. DCL.....	23
Gambar 20. Output DCL.....	24

DAFTAR TABEL

Gambar 2. Source Code Database dan Tabel	9
Gambar 3. Source Code Tabel	10
Gambar 6. Output Table Sewa	14
Gambar 7. Output Tabel Penghuni	15
Gambar 8. Output Tabel Pembayaran	15
Gambar 9. Output Tabel Kamar	15

LANDASAN TEORI

Basis data atau database, berasal dari kata yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang, tempat berkumpul, data merupakan wujud dari dunia nyata yang mewakili suatu objek seperti manusia (siswa, mahasiswa, pegawai, pelanggan), barang, hewan, dll, yang dinyatakan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Sehingga dapat disimpulkan bahwa basis data adalah kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.

Basis data adalah kumpulan file-file yang saling berhubungan yang biasa ditunjukkan dengan kunci dari tiap file yang ada. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. Dalam satu file terdapat record-record yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan entitas yang seragam. Jadi sistem manajemen basis data dan set program pengelola yang berfungsi untuk membaca data, menambah data, menghapus data dan mengambil data

Database Management System (DBMS) atau sistem manajemen basis data adalah perangkat lunak yang digunakan untuk mengelola dan mengatur data dalam suatu basis data. Basis data merupakan kumpulan data yang terstruktur, tersimpan, dan dapat diakses dengan mudah. Database Management System (DBMS) memiliki sejumlah fungsi yang sangat penting dalam dunia teknologi dan bisnis. Fungsi-fungsi ini mendukung penyimpanan, pengelolaan, dan akses data yang efisien. Dalam dunia bisnis yang makin tergantung pada data, pemahaman tentang DBMS menjadi makin penting.

Fungsi DBMS (Database Manajemen System) adalah menyimpan dan mencari informasi basis data dengan mudah, cepat dan efisien. Sistem basis data yang dirancang untuk mengelolah banyak informasi. Data data yang diperlukan diolah melalui analisis tertentu untuk pengambilan keputusan. Basis data sangat erat dalam kehidupan sehari – hari yaitu data perusahaan, data bank, data universitas, data kemahasiswaan, data pegawai, dan lain sebagainya. Peran DBMS menghubungkan antara pengguna dengan database yaitu kumpulan data relasional yang disimpan diperangkat keras untuk memanipulasi perangkat lunak. Basis data juga merupakan

komponen penting dalam memberikan informasi kepada pengguna dan paket perangkat lunak kompleks yang digunakan untuk memanipulasi basis data.

MySQL adalah sebuah sistem manajemen basis data relasional (RDBMS) open source yang banyak digunakan untuk mengelola dan mengorganisir data dalam bentuk tabel. Setiap basis data dalam MySQL dapat terdiri dari satu atau lebih tabel, di mana setiap tabel tersusun atas sejumlah baris (record) dan kolom (field). Struktur ini memungkinkan penyimpanan data yang terorganisir dan dapat diakses dengan mudah. Sebagai salah satu database server yang populer, MySQL memiliki berbagai keunggulan yang menjadikannya pilihan utama bagi banyak pengembang dan praktisi dalam membangun berbagai jenis proyek, baik untuk keperluan pribadi, bisnis, maupun skala industri. Keunggulan tersebut meliputi kecepatan dalam pemrosesan data, kemudahan penggunaan, skalabilitas tinggi, serta kompatibilitas dengan berbagai platform dan sistem operasi.(Pratama, 2020)

Sistem manajemen basis data dirancang untuk mengatasi kelemahan sistem manajemen file. Pengembangan sistem basis data tidak terlepas dari perangkat keras dan perangkat lunak. Perkembangan teknologi jaringan komputer dan telekomunikasi merupakan salah satu pendorong implementasi database. Dalam sistem basis data, kekurangan yang dialami oleh sistem file/file handling dapat diatasi dengan memiliki semua file di satu tempat sehingga beberapa program/pengguna dapat mengaksesnya secara bersamaan. Software DBMS digunakan untuk mengelola database dan mengolah data menjadi informasi yang berguna. Contoh sistem DBMS yang banyak digunakan adalah MySQL, PostgreSQL, SQL Server, Oracle, Sybase.(Gunawan, Ningsih and Lantana, 2023)

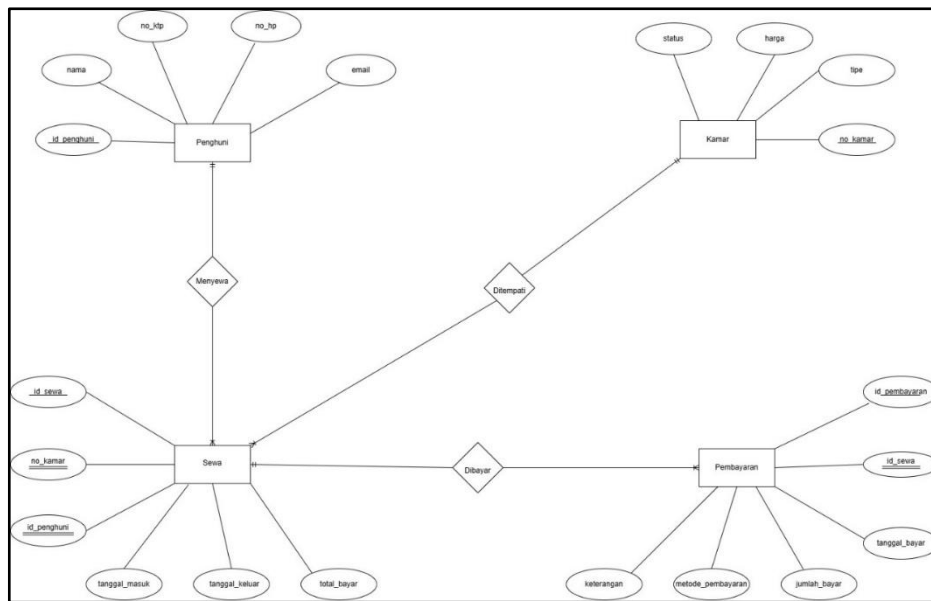
Basis data adalah kumpulan terpadu dari elemen data logis yang saling berhubungan. Basis data mengonsolidasi banyak catatan yang sebelumnya disimpan dalam file terpisah. Merupakan suatu kumpulan data yang berhubungan secara logis dan deskripsi data tersebut, yang dirancang untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi. Artinya basis data, tempat penyimpanan data yang besar dimana dapat digunakan oleh banyak pengguna. Seluruh item basis data tidak lagi dimiliki oleh satu departemen, melainkan menjadi sumber daya perusahaan yang dapat digunakan bersama-sama. Basis data (database) merupakan kumpulan dari suatu data yang saling berhubungan satu dengan yang lainnya,

tersimpan disimpan luar komputer (external memory) dan digunakan perangkat lunak tertentu untuk memanipulasinya. Database merupakan salah satu komponen yang penting di SI, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Sistem basis data (database system) ini adalah suatu SI yang mengintegrasikan kumpulan data yang saling berhubungan (Sudarso, 2022).

Basis data merupakan kumpulan terpadu dari elemen data logis yang saling berhubungan dan terorganisir secara sistematis untuk memudahkan pengelolaan, penyimpanan, serta pengambilan data. Dalam implementasinya, basis data mengonsolidasi banyak catatan yang sebelumnya tersimpan dalam file terpisah, sehingga mengurangi redundansi dan meningkatkan efisiensi dalam pengelolaan informasi. Sebagai suatu sistem yang terstruktur, basis data terdiri dari kumpulan file yang saling berelasi, di mana setiap file menyimpan data dengan format yang telah distandarisi. Struktur ini memungkinkan integrasi data yang lebih baik, memastikan konsistensi, serta mempermudah proses pencarian, manipulasi, dan analisis informasi. Selain itu, basis data dirancang agar dapat diakses oleh banyak pengguna secara bersamaan dengan mekanisme pengendalian akses yang ketat untuk menjaga keamanan dan integritas data. Dengan adanya sistem basis data, organisasi atau individu dapat mengelola informasi dengan lebih efektif, mendukung pengambilan keputusan, serta meningkatkan produktivitas dalam berbagai bidang seperti bisnis, pendidikan, pemerintahan, dan teknologi. (Maulana, 2022)

DDL (Data Definition Language) merupakan perintah dasar dalam SQL yang digunakan untuk membuat basis data dan mendefinisikan struktur tabel dalam basis data tersebut. Struktur tabel yang dimaksud terdiri dari nama kolom, lebar kolom, penentuan key, dan relasi antar tabel. Statement query DDL adalah kumpulan perintah terdiri dari Create, Alter, Drop untuk mendefinisikan tipe data dari objek-objek basis data. DML (Data Manipulation Language) merupakan perintah dalam SQL yang memungkinkan pengguna untuk memanipulasi data dalam tabel. Yang dimaksud memanipulasi data adalah, menambah data baru pada suatu table, merubah data yang sudah ada pada suatu table, menghapus data pada suatu table, menampilkan informasi-informasi yang disimpan di dalam tabel pada sebuah basis data. (Ummah, 2019)

HASIL DAN PEMBAHASAN



Gambar 1. ERD

Penjelasan :

Entity Relationship Diagram (ERD) pada gambar tersebut menggambarkan sistem manajemen penyewaan kamar kost. Terdapat lima entitas utama yaitu Penghuni, Kamar, Sewa, Pembayaran, dan tiga relasi yang menghubungkannya, yaitu Menyewa, Ditempati, dan Dibayar.

Entitas Penghuni memiliki atribut **id_penghuni** sebagai primary key, serta atribut lain seperti nama, **no_ktp**, **no_hp**, dan **email**. Penghuni dapat menyewa kamar, yang direpresentasikan melalui entitas Sewa dengan atribut **id_sewa**, **id_penghuni**, **no_kamar**, **tanggal_masuk**, **tanggal_keluar**, dan **total_bayar**. Relasi Menyewa menghubungkan entitas Penghuni dan Sewa.

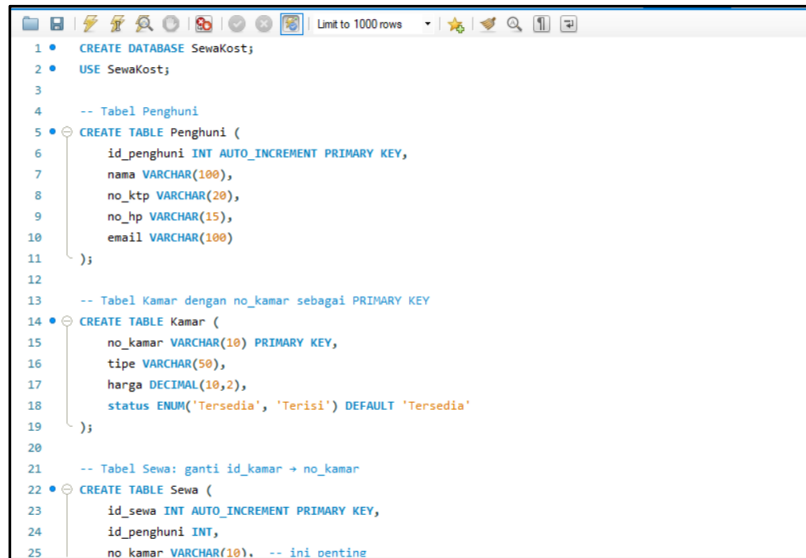
Entitas Kamar menyimpan informasi tentang kamar kost, seperti **no_kamar** sebagai primary key, tipe, harga, dan status. Relasi Ditempati menunjukkan bahwa sebuah kamar bisa ditempati melalui aktivitas sewa yang dilakukan penghuni.

Selanjutnya, entitas Pembayaran memiliki atribut seperti **id_pembayaran** (primary key), **id_sewa** (foreign key), **tanggal_bayar**, **jumlah_bayar**, **metode_pembayaran**, dan **keterangan**. Relasi Dibayar menghubungkan entitas Sewa dengan entitas Pembayaran, yang menunjukkan bahwa setiap penyewaan dapat memiliki satu atau lebih transaksi pembayaran. Secara keseluruhan, ERD ini menunjukkan alur informasi dari proses penyewaan kamar kost, mulai dari data

penghuni dan kamar, pencatatan sewa, hingga transaksi pembayaran yang terjadi. Diagram ini mendukung pengelolaan data kost secara terstruktur dan relasional.

1. Membuat Database dan Tabel

Printscreen Source Code :

A screenshot of a SQL IDE window. The window title is "Limit to 1000 rows". The code is as follows:

```
1 • CREATE DATABASE SewaKost;
2 • USE SewaKost;
3
4 -- Tabel Penghuni
5 • CREATE TABLE Penghuni (
6   id_penghuni INT AUTO_INCREMENT PRIMARY KEY,
7   nama VARCHAR(100),
8   no_ktp VARCHAR(20),
9   no_hp VARCHAR(15),
10  email VARCHAR(100)
11 );
12
13 -- Tabel Kamar dengan no_kamar sebagai PRIMARY KEY
14 • CREATE TABLE Kamar (
15   no_kamar VARCHAR(10) PRIMARY KEY,
16   tipe VARCHAR(50),
17   harga DECIMAL(10,2),
18   status ENUM('Tersedia', 'Terisi') DEFAULT 'Tersedia'
19 );
20
21 -- Tabel Sewa: ganti id_kamar → no_kamar
22 • CREATE TABLE Sewa (
23   id_sewa INT AUTO_INCREMENT PRIMARY KEY,
24   id_penghuni INT,
25   no_kamar VARCHAR(10), -- ini penting
```

Gambar 2. Source Code Database dan Tabel

Source Code :

CREATE DATABASE SewaKost;

USE SewaKost;

```
CREATE TABLE Penghuni (
    id_penghuni INT AUTO_INCREMENT PRIMARY KEY,
    nama VARCHAR(100),
    no_ktp VARCHAR(20),
    no_hp VARCHAR(15),
    email VARCHAR(100)
);
```

```
CREATE TABLE Kamar (
    no_kamar VARCHAR(10) PRIMARY KEY,
    tipe VARCHAR(50),
    harga DECIMAL(10,2),
    status ENUM('Tersedia', 'Terisi') DEFAULT 'Tersedia'
);
```

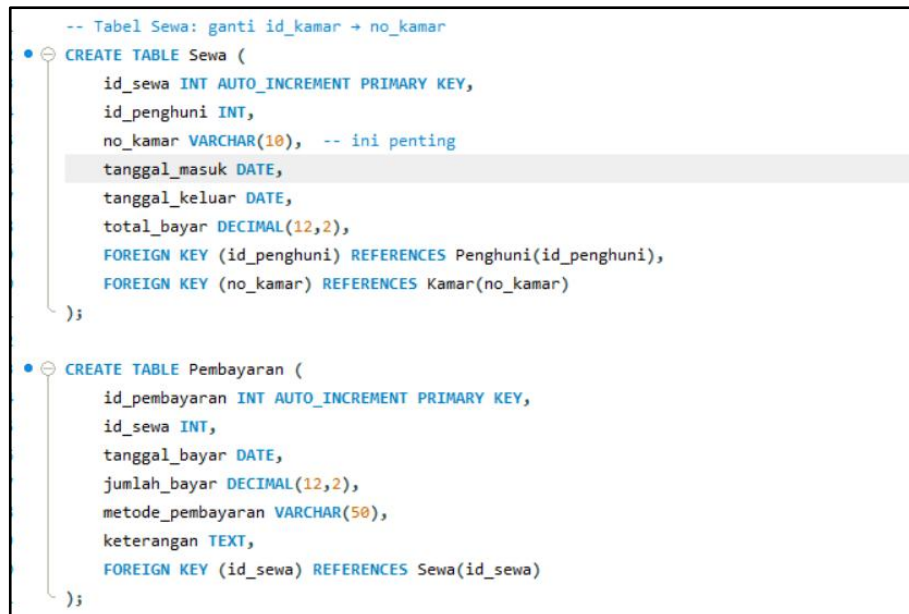
```
CREATE TABLE Sewa (
```

```

id_sewa INT AUTO_INCREMENT PRIMARY KEY,
id_penghuni INT,
no_kamar VARCHAR(10), -- ini penting
tanggal_masuk DATE,
tanggal_keluar DATE,
total_bayar DECIMAL(12,2),
FOREIGN KEY (id_penghuni) REFERENCES Penghuni(id_penghuni),
FOREIGN KEY (no_kamar) REFERENCES Kamar(no_kamar)
);

```

Printscreen Source Code :



```

-- Tabel Sewa: ganti id_kamar → no_kamar
CREATE TABLE Sewa (
    id_sewa INT AUTO_INCREMENT PRIMARY KEY,
    id_penghuni INT,
    no_kamar VARCHAR(10), -- ini penting
    tanggal_masuk DATE,
    tanggal_keluar DATE,
    total_bayar DECIMAL(12,2),
    FOREIGN KEY (id_penghuni) REFERENCES Penghuni(id_penghuni),
    FOREIGN KEY (no_kamar) REFERENCES Kamar(no_kamar)
);

CREATE TABLE Pembayaran (
    id_pembayaran INT AUTO_INCREMENT PRIMARY KEY,
    id_sewa INT,
    tanggal_bayar DATE,
    jumlah_bayar DECIMAL(12,2),
    metode_pembayaran VARCHAR(50),
    keterangan TEXT,
    FOREIGN KEY (id_sewa) REFERENCES Sewa(id_sewa)
);

```

Gambar 3. Source Code Tabel

Source Code :

```

CREATE TABLE Pembayaran (
    id_pembayaran INT AUTO_INCREMENT PRIMARY KEY,
    id_sewa INT,
    tanggal_bayar DATE,
    jumlah_bayar DECIMAL(12,2),
    metode_pembayaran VARCHAR(50),
    keterangan TEXT,
    FOREIGN KEY (id_sewa) REFERENCES Sewa(id_sewa)
);

```

Penjelasan Source Code :

Source code SQL di atas digunakan untuk membangun sebuah sistem database bernama SewaKost yang mengelola data penyewaan kamar kos. Pertama-tama, database bernama SewaKost dibuat dan diaktifkan untuk digunakan. Selanjutnya, dibuat tabel Penghuni untuk menyimpan data pribadi penghuni kos, seperti nama, nomor KTP, nomor HP, dan email, dengan id_penghuni sebagai primary key yang akan bertambah otomatis. Kemudian, tabel Kamar dibuat untuk mencatat informasi kamar yang tersedia, seperti nomor kamar (no_kamar) sebagai primary key, tipe kamar, harga sewa, dan status kamar yang hanya bisa bernilai 'Tersedia' atau 'Terisi'.

Tabel ketiga, yaitu Sewa, mencatat transaksi penyewaan kamar oleh penghuni. Tabel ini menghubungkan penghuni dan kamar melalui **id_penghuni** dan **no_kamar**, keduanya sebagai foreign key yang mengacu ke tabel Penghuni dan Kamar. Selain itu, terdapat kolom tanggal masuk, tanggal keluar, dan total bayar. Terakhir, tabel Pembayaran digunakan untuk mencatat rincian pembayaran dari setiap transaksi sewa. Setiap entri di tabel ini memiliki id_pembayaran sebagai primary key, serta mengacu pada **id_sewa** sebagai foreign key ke tabel Sewa. Tabel ini juga mencatat tanggal pembayaran, jumlah yang dibayar, metode pembayaran, dan keterangan tambahan. Hubungan antar tabel membentuk struktur relasional yang memungkinkan pengelolaan data penyewaan kamar, penghuni, dan pembayaran secara efektif dan terintegrasi.

Printscreen Source Code :

```
42
43 -- Data penghuni tetap
44 • INSERT INTO Penghuni (nama, no_ktp, no_hp, email) VALUES
45 ('Agyl Mendi Pratama', '3201010100010001', '081234567891', 'agyl.pratama@gmail.com'),
46 ('Habib Al Qodri', '3201010100010002', '081234567892', 'habib.qodri@gmail.com'),
47 ('Ricardo Gellael', '3201010100010003', '081234567893', 'ricardo.gellael@gmail.com'),
48 ('Juan Argapra Putra', '3201010100010004', '081234567894', 'juan.arga@gmail.com'),
49 ('Muhammad Ryan Al-Habsy', '3201010100010005', '081234567895', 'ryan.alhabsy@gmail.com');
50
51 -- Data kamar dengan no_kamar sebagai PK
52 • INSERT INTO Kamar (no_kamar, tipe, harga, status) VALUES
53 ('A101', 'Standar', 750000.00, 'Terisi'), -- Agyl
54 ('A102', 'Standar', 750000.00, 'Terisi'), -- Habib
55 ('B201', 'Deluxe', 1000000.00, 'Terisi'), -- Ricardo
56 ('B202', 'Deluxe', 1000000.00, 'Terisi'), -- Juan
57 ('C301', 'Premium', 1500000.00, 'Terisi'), -- Ryan
58 ('C302', 'Premium', 1500000.00, 'Tersedia'),
59 ('D101', 'Standar', 750000.00, 'Tersedia'),
60 ('D102', 'Standar', 750000.00, 'Tersedia'),
61 ('E201', 'Deluxe', 1000000.00, 'Tersedia'),
62 ('E202', 'Deluxe', 1000000.00, 'Tersedia');
```

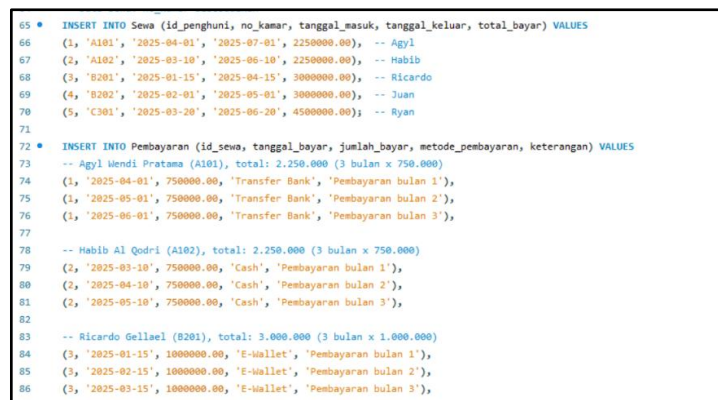
Gambar 4. Source Code Input Data (1)

Source Code :

```
INSERT INTO Penghuni (nama, no_ktp, no_hp, email) VALUES
('Agyl Wendi Pratama', '3201010100010001', '081234567891',
'agyl.pratama@gmail.com'),
('Habib Al Qodri', '3201010100010002', '081234567892',
'habib.qodri@gmail.com'),
('Ricardo Gellael', '3201010100010003', '081234567893',
'ricardo.gellael@gmail.com'),
('Juan Argaprana Putra', '3201010100010004', '081234567894',
'juan.arga@gmail.com'),
('Muhammad Ryan Al-Habsy', '3201010100010005', '081234567895',
'ryan.alhabsy@gmail.com');

INSERT INTO Kamar (no_kamar, tipe, harga, status) VALUES
('A101', 'Standar', 750000.00, 'Terisi'), -- Agyl
('A102', 'Standar', 750000.00, 'Terisi'), -- Habib
('B201', 'Deluxe', 1000000.00, 'Terisi'), -- Ricardo
('B202', 'Deluxe', 1000000.00, 'Terisi'), -- Juan
('C301', 'Premium', 1500000.00, 'Terisi'), -- Ryan
('C302', 'Premium', 1500000.00, 'Tersedia'),
('D101', 'Standar', 750000.00, 'Tersedia'),
('D102', 'Standar', 750000.00, 'Tersedia'),
('E201', 'Deluxe', 1000000.00, 'Tersedia'),
('E202', 'Deluxe', 1000000.00, 'Tersedia');
```

Printscreen Source Code :



```
65 * INSERT INTO Sewa (id_penghuni, no_kamar, tanggal_masuk, tanggal_keluar, total_bayar) VALUES
66 (1, 'A101', '2025-04-01', '2025-07-01', 2250000.00), -- Agyl
67 (2, 'A102', '2025-03-10', '2025-06-10', 2250000.00), -- Habib
68 (3, 'B201', '2025-01-15', '2025-04-15', 3000000.00), -- Ricardo
69 (4, 'B202', '2025-02-01', '2025-05-01', 3000000.00), -- Juan
70 (5, 'C301', '2025-03-20', '2025-06-20', 4500000.00); -- Ryan
71
72 * INSERT INTO Pembayaran (id_sewa, tanggal_bayar, jumlah_bayar, metode_pembayaran, keterangan) VALUES
73 -- Agyl Wendi Pratama (A101), total: 2.250.000 (3 bulan x 750.000)
74 (1, '2025-04-01', 750000.00, 'Transfer Bank', 'Pembayaran bulan 1'),
75 (1, '2025-05-01', 750000.00, 'Transfer Bank', 'Pembayaran bulan 2'),
76 (1, '2025-06-01', 750000.00, 'Transfer Bank', 'Pembayaran bulan 3'),
77
78 -- Habib Al Qodri (A102), total: 2.250.000 (3 bulan x 750.000)
79 (2, '2025-03-10', 750000.00, 'Cash', 'Pembayaran bulan 1'),
80 (2, '2025-04-10', 750000.00, 'Cash', 'Pembayaran bulan 2'),
81 (2, '2025-05-10', 750000.00, 'Cash', 'Pembayaran bulan 3'),
82
83 -- Ricardo Gellael (B201), total: 3.000.000 (3 bulan x 1.000.000)
84 (3, '2025-01-15', 1000000.00, 'E-wallet', 'Pembayaran bulan 1'),
85 (3, '2025-02-15', 1000000.00, 'E-wallet', 'Pembayaran bulan 2'),
86 (3, '2025-03-15', 1000000.00, 'E-wallet', 'Pembayaran bulan 3'),
87
```

Gambar 5. Source Code Input Data

Source Code :

```
INSERT INTO Sewa (id_penghuni, no_kamar, tanggal_masuk, tanggal_keluar, total_bayar) VALUES
```

```
(1, 'A101', '2025-04-01', '2025-07-01', 2250000.00), -- Agyl  
(2, 'A102', '2025-03-10', '2025-06-10', 2250000.00), -- Habib  
(3, 'B201', '2025-01-15', '2025-04-15', 3000000.00), -- Ricardo  
(4, 'B202', '2025-02-01', '2025-05-01', 3000000.00), -- Juan  
(5, 'C301', '2025-03-20', '2025-06-20', 4500000.00); -- Ryan
```

```
INSERT INTO Pembayaran (id_sewa, tanggal_bayar, jumlah_bayar, metode_pembayaran, keterangan) VALUES
```

```
-- Agyl Wendi Pratama (A101), total: 2.250.000 (3 bulan x 750.000)
```

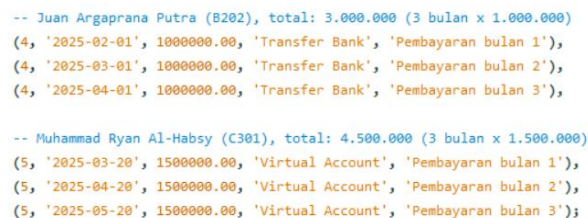
```
(1, '2025-04-01', 750000.00, 'Transfer Bank', 'Pembayaran bulan 1'),  
(1, '2025-05-01', 750000.00, 'Transfer Bank', 'Pembayaran bulan 2'),  
(1, '2025-06-01', 750000.00, 'Transfer Bank', 'Pembayaran bulan 3'),
```

```
-- Habib Al Qodri (A102), total: 2.250.000 (3 bulan x 750.000)
```

```
(2, '2025-03-10', 750000.00, 'Cash', 'Pembayaran bulan 1'),  
(2, '2025-04-10', 750000.00, 'Cash', 'Pembayaran bulan 2'),  
(2, '2025-05-10', 750000.00, 'Cash', 'Pembayaran bulan 3'),
```

```
-- Ricardo Gellael (B201), total: 3.000.000 (3 bulan x 1.000.000)
```

```
(3, '2025-01-15', 1000000.00, 'E-Wallet', 'Pembayaran bulan 1'),  
(3, '2025-02-15', 1000000.00, 'E-Wallet', 'Pembayaran bulan 2'),  
(3, '2025-03-15', 1000000.00, 'E-Wallet', 'Pembayaran bulan 3'),
```

Printscreen Source Code :

```
-- Juan Argaprana Putra (B202), total: 3.000.000 (3 bulan x 1.000.000)  
(4, '2025-02-01', 1000000.00, 'Transfer Bank', 'Pembayaran bulan 1'),  
(4, '2025-03-01', 1000000.00, 'Transfer Bank', 'Pembayaran bulan 2'),  
(4, '2025-04-01', 1000000.00, 'Transfer Bank', 'Pembayaran bulan 3'),  
  
-- Muhammad Ryan Al-Habsy (C301), total: 4.500.000 (3 bulan x 1.500.000)  
(5, '2025-03-20', 1500000.00, 'Virtual Account', 'Pembayaran bulan 1'),  
(5, '2025-04-20', 1500000.00, 'Virtual Account', 'Pembayaran bulan 2'),  
(5, '2025-05-20', 1500000.00, 'Virtual Account', 'Pembayaran bulan 3');
```

Gambar 5. Source Code Input Data

Source Code :

```
-- Juan Argaprana Putra (B202), total: 3.000.000 (3 bulan x 1.000.000)  
(4, '2025-02-01', 1000000.00, 'Transfer Bank', 'Pembayaran bulan 1'),
```

(4, '2025-03-01', 1000000.00, 'Transfer Bank', 'Pembayaran bulan 2'),
 (4, '2025-04-01', 1000000.00, 'Transfer Bank', 'Pembayaran bulan 3'),
 -- Muhammad Ryan Al-Habsy (C301), total: 4.500.000 (3 bulan x 1.500.000)
 (5, '2025-03-20', 1500000.00, 'Virtual Account', 'Pembayaran bulan 1'),
 (5, '2025-04-20', 1500000.00, 'Virtual Account', 'Pembayaran bulan 2'),
 (5, '2025-05-20', 1500000.00, 'Virtual Account', 'Pembayaran bulan 3');

Penjelasan Source Code :

Pada sebuah sistem manajemen kos yang terdiri dari empat tabel utama: Penghuni, Kamar, Sewa, dan Pembayaran. Pertama, tabel Penghuni diisi dengan data lima orang penyewa kost, masing-masing memiliki informasi nama, nomor KTP, nomor HP, dan email. Kemudian, tabel Kamar menyimpan data 10 kamar kos yang tersedia, dengan rincian nomor kamar, tipe kamar (Standar, Deluxe, Premium), harga bulanan, dan status apakah kamar tersebut sudah “Terisi” atau masih “Tersedia”. Selanjutnya, tabel Sewa mencatat relasi antara penghuni dengan kamar yang mereka sewa, termasuk tanggal masuk, tanggal keluar, dan total pembayaran untuk masa sewa (misalnya Agyl menyewa kamar A101 selama tiga bulan dengan total Rp2.250.000).

Terakhir, tabel Pembayaran menyimpan detail riwayat pembayaran dari masing-masing sewa. Setiap entri mencakup ID sewa yang terkait, tanggal pembayaran, jumlah yang dibayarkan, metode pembayaran (seperti Transfer Bank, Cash, E-Wallet, atau Virtual Account), dan keterangan bulan berapa pembayaran dilakukan. Struktur ini mencerminkan bagaimana sistem pencatatan sewa kos diorganisasi secara sistematis, dari identitas penyewa, status ketersediaan kamar, hubungan sewa, hingga detail transaksi keuangan bulanannya.

Output :

Result Grid						
Filter Rows:						
	id_sewa	id_penghuni	no_kamar	tanggal_masuk	tanggal_keluar	total_bayar
▶	1	1	A101	2025-04-01	2025-07-01	2250000.00
	2	2	A102	2025-03-10	2025-06-10	2250000.00
	3	3	B201	2025-01-15	2025-04-15	3000000.00
	4	4	B202	2025-02-01	2025-05-01	3000000.00
	5	5	C301	2025-03-20	2025-06-20	4500000.00
*	NULL	NULL	NULL	NULL	NULL	NULL

Gambar 6. Output Table Sewa

Result Grid					
Filter Rows:					
Edit: Export/Import: Wrap Cell Content:					
	id_penghuni	nama	no_ktp	no_hp	email
1	1	Agyl Wendi Pratama	3201010100010001	081234567891	agyl.pratama@gmail.com
2	2	Habib Al Qodri	3201010100010002	081234567892	habib.qodri@gmail.com
3	3	Ricardo Gellael	3201010100010003	081234567893	ricardo.gellael@gmail.com
4	4	Juan Argaprana Putra	3201010100010004	081234567894	juan.arga@gmail.com
5	5	Muhammad Ryan Al-Habsy	3201010100010005	081234567895	ryan.alhabsy@gmail.com
*	NULL	NULL	NULL	NULL	NULL

Gambar 7. Output Tabel Penghuni

Result Grid						
Filter Rows:						
Edit: Export/Import: Wrap Cell Content:						
	id_pembayaran	id_sewa	tanggal_bayar	jumlah_bayar	metode_pembayaran	keterangan
1	1	1	2025-04-01	750000.00	Transfer Bank	Pembayaran bulan 1
2	2	1	2025-05-01	750000.00	Transfer Bank	Pembayaran bulan 2
3	3	1	2025-06-01	750000.00	Transfer Bank	Pembayaran bulan 3
4	4	2	2025-03-10	750000.00	Cash	Pembayaran bulan 1
5	5	2	2025-04-10	750000.00	Cash	Pembayaran bulan 2
6	6	2	2025-05-10	750000.00	Cash	Pembayaran bulan 3
7	7	3	2025-01-15	1000000.00	E-Wallet	Pembayaran bulan 1
8	8	3	2025-02-15	1000000.00	E-Wallet	Pembayaran bulan 2
9	9	3	2025-03-15	1000000.00	E-Wallet	Pembayaran bulan 3
10	10	4	2025-02-01	1000000.00	Transfer Bank	Pembayaran bulan 1
11	11	4	2025-03-01	1000000.00	Transfer Bank	Pembayaran bulan 2
12	12	4	2025-04-01	1000000.00	Transfer Bank	Pembayaran bulan 3
13	13	5	2025-03-20	1500000.00	Virtual Account	Pembayaran bulan 1
14	14	5	2025-04-20	1500000.00	Virtual Account	Pembayaran bulan 2
15	15	5	2025-05-20	1500000.00	Virtual Account	Pembayaran bulan 3
*	NULL	NULL	NULL	NULL	NULL	NULL

Gambar 8. Output Tabel Pembayaran

Result Grid				
Filter Rows:				
Edit: Export/Import: Wrap Cell Content:				
	no_kamar	tipe	harga	status
1	A101	Standar	750000.00	Terisi
2	A102	Standar	750000.00	Terisi
3	B201	Deluxe	1000000.00	Terisi
4	B202	Deluxe	1000000.00	Terisi
5	C301	Premium	1500000.00	Terisi
6	C302	Premium	1500000.00	Tersedia
7	D101	Standar	750000.00	Tersedia
8	D102	Standar	750000.00	Tersedia
9	E201	Deluxe	1000000.00	Tersedia
10	E202	Deluxe	1000000.00	Tersedia
*	NULL	NULL	NULL	NULL

Gambar 9. Output Tabel Kamar

2. Alias, Operator, dan Join

Printscreen Source Code :

```
104 • SELECT
105     s.id_sewa,
106     p.nama AS Nama_Penghuni,
107     k.no_kamar AS Nomor_Kamar,
108     k.harga,
109     s.total_bayar,
110     IF(s.total_bayar >= k.harga, 'LUNAS', 'BELUM LUNAS') AS Status_Pembayaran
111 FROM Sewa s
112 JOIN Penghuni p ON s.id_penghuni = p.id_penghuni
113 JOIN Kamar k ON s.no_kamar = k.no_kamar;
114
115 -- Rata-rata harga kamar per tipe, diurutkan dari yang tertinggi
116 • SELECT
117     tipe,
118     COUNT(*) AS Jumlah_Kamar,
119     AVG(harga) AS Rata_Harga
120 FROM Kamar
121 GROUP BY tipe
122 ORDER BY Rata_Harga DESC;
123
```

Gambar 10. Alias, Operator dan Join

```
125 • SELECT
126     p.nama AS Nama_Penghuni,
127     s.no_kamar,
128     s.total_bayar AS Total_Tagihan,
129     COALESCE(SUM(pb.jumlah_bayar), 0) AS Total_Dibayar,
130     (s.total_bayar - COALESCE(SUM(pb.jumlah_bayar), 0)) AS Sisa_Bayar
131 FROM Sewa s
132 JOIN Penghuni p ON s.id_penghuni = p.id_penghuni
133 LEFT JOIN Pembayaran pb ON s.id_sewa = pb.id_sewa
134 GROUP BY s.id_sewa;
135
136 -- View untuk mempermudah melihat data penyewaan
137 • CREATE VIEW vw_data_sewa AS
138 SELECT
139     s.id_sewa,
140     p.nama,
141     s.no_kamar,
142     s.tanggal_masuk,
143     s.tanggal_keluar,
144     s.total_bayar
145 FROM Sewa s
146 JOIN Penghuni p ON s.id_penghuni = p.id_penghuni
147 JOIN Kamar k ON s.no_kamar = k.no_kamar;
148
149 SELECT * From vw_data_sewa
```

Gambar 11. Alias, Operator dan Join

Source Code :

SELECT

s.id_sewa,

p.nama AS Nama_Penghuni,

```

    k.no_kamar AS Nomor_Kamar,
    k.harga,
    s.total_bayar,
    IF(s.total_bayar >= k.harga, 'LUNAS', 'BELUM LUNAS') AS
Status_Pembayaran
FROM Sewa s
JOIN Penghuni p ON s.id_penghuni = p.id_penghuni
JOIN Kamar k ON s.no_kamar = k.no_kamar;

SELECT
    tipe,
    COUNT(*) AS Jumlah_Kamar,
    AVG(harga) AS Rata_Harga
FROM Kamar
GROUP BY tipe
ORDER BY Rata_Harga DESC;

SELECT
    p.nama AS Nama_Penghuni,
    s.no_kamar,
    s.total_bayar AS Total_Tagihan,
    COALESCE(SUM(pb.jumlah_bayar), 0) AS Total_Dibayar,
    (s.total_bayar - COALESCE(SUM(pb.jumlah_bayar), 0)) AS Sisa_Bayar
FROM Sewa s
JOIN Penghuni p ON s.id_penghuni = p.id_penghuni
LEFT JOIN Pembayaran pb ON s.id_sewa = pb.id_sewa
GROUP BY s.id_sewa;

CREATE VIEW vw_data_sewa AS
SELECT
    s.id_sewa,
    p.nama,

```

s.no_kamar,

s.tanggal_masuk,

s.tanggal_keluar,

s.total_bayar

FROM Sewa s

JOIN Penghuni p ON s.id_penghuni = p.id_penghuni

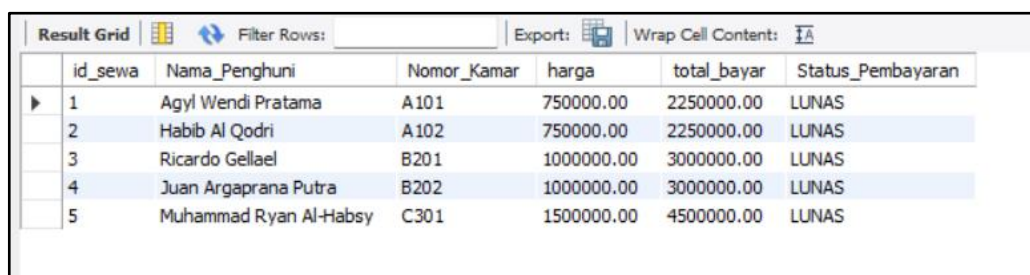
JOIN Kamar k ON s.no_kamar = k.no_kamar;

SELECT * From vw_data_sewa

Penjelasan Source Code :

Query pertama menampilkan informasi penyewaan dari tabel Sewa, Penghuni, dan Kamar, termasuk status pembayaran dengan kondisi: jika total bayar minimal sama dengan harga kamar, maka dianggap LUNAS, jika tidak maka BELUM LUNAS. Query kedua menghitung rata-rata harga kamar untuk setiap tipe kamar (Standar, Deluxe, Premium) beserta jumlah kamarnya, dan hasilnya diurutkan dari harga rata-rata tertinggi hingga terendah, sehingga berguna untuk mengetahui segmentasi harga berdasarkan tipe. Query ketiga merupakan laporan gabungan antara total tagihan dari tabel Sewa dengan total pembayaran aktual yang tercatat pada tabel Pembayaran, menggunakan **COALESCE** untuk menghindari nilai **NULL**. Ini menghasilkan informasi penting seperti sisa pembayaran yang harus dibayarkan oleh penghuni. Terakhir, dibuat sebuah view bernama vw_data_sewa untuk menyederhanakan akses terhadap data penyewaan, dengan menyatukan informasi penting seperti ID sewa, nama penghuni, nomor kamar, tanggal masuk-keluar, dan total bayar. View ini memungkinkan pengguna melakukan query data penyewaan secara lebih efisien dan terstruktur.

Output :



	id_sewa	Nama_Penghuni	Nomor_Kamar	harga	total_bayar	Status_Pembayaran
▶	1	Agyi Wendi Pratama	A101	750000.00	2250000.00	LUNAS
	2	Habib Al Qodri	A102	750000.00	2250000.00	LUNAS
	3	Ricardo Gellael	B201	1000000.00	3000000.00	LUNAS
	4	Juan Argaprana Putra	B202	1000000.00	3000000.00	LUNAS
	5	Muhammad Ryan Al-Habsy	C301	1500000.00	4500000.00	LUNAS

Gambar 11. Tabel Output Alias dan Operator

Result Grid			
	Filter Rows:		
	type	Jumlah_Kamar	Rata_Harga
▶	Premium	2	1500000.000000
	Deluxe	4	1000000.000000
	Standar	4	750000.000000

Gambar 12. Tabel Output Alias dan Operator

	Nama_Penghuni	no_kamar	Total_Tagihan	Total_Dibayar	Sisa_Bayar
▶	Agyl Wendi Pratama	A101	2250000.00	2250000.00	0.00
	Habib Al Qodri	A102	2250000.00	2250000.00	0.00
	Ricardo Gellael	B201	3000000.00	3000000.00	0.00
	Juan Argaprana Putra	B202	3000000.00	3000000.00	0.00
	Muhammad Ryan Al-Habsy	C301	4500000.00	4500000.00	0.00

Gambar 13. Tabel Output Alias dan Operator

	id_sewa	nama	no_kamar	tanggal_masuk	tanggal_keluar	total_bayar
▶	1	Agyl Wendi Pratama	A101	2025-04-01	2025-07-01	2250000.00
	2	Habib Al Qodri	A102	2025-03-10	2025-06-10	2250000.00
	3	Ricardo Gellael	B201	2025-01-15	2025-04-15	3000000.00
	4	Juan Argaprana Putra	B202	2025-02-01	2025-05-01	3000000.00
	5	Muhammad Ryan Al-Habsy	C301	2025-03-20	2025-06-20	4500000.00

Gambar 14. Tabel Output Alias dan Operator

3. Function

Printscreen Source Code :

```

152 DELIMITER //
153
154 CREATE PROCEDURE TambahSewa(
155     IN peng_id INT,
156     IN kamar_nomor VARCHAR(10),
157     IN tgl_masuk DATE,
158     IN tgl_keluar DATE,
159     IN total DECIMAL(12,2)
160 )
161 BEGIN
162     INSERT INTO Sewa (id_penghuni, no_kamar, tanggal_masuk, tanggal_keluar, total_bayar)
163     VALUES (peng_id, kamar_nomor, tgl_masuk, tgl_keluar, total);
164
165     UPDATE Kamar SET status = 'Terisi' WHERE no_kamar = kamar_nomor;
166 END //
167
168 DELIMITER ;
169

```

Gambar 15. Function

```

180 DELIMITER //
181
182 • CREATE TRIGGER after_delete_sewa
183 AFTER DELETE ON Sewa
184 FOR EACH ROW
185 BEGIN
186     UPDATE Kamar SET status = 'Tersedia' WHERE no_kamar = OLD.no_kamar;
187 END //
188
189 DELIMITER ;
190

```

Gambar 16. Function

Source Code :

DELIMITER //

CREATE PROCEDURE TambahSewa(

IN peng_id INT,

IN kamar_nomor VARCHAR(10),

IN tgl_masuk DATE,

IN tgl_keluar DATE,

IN total DECIMAL(12,2)

)

BEGIN

INSERT INTO Sewa (id_penghuni, no_kamar, tanggal_masuk,
tanggal_keluar, total_bayar)

VALUES (peng_id, kamar_nomor, tgl_masuk, tgl_keluar, total);

UPDATE Kamar SET status = 'Terisi' WHERE no_kamar = kamar_nomor;

END //

DELIMITER ;

DELIMITER //

CREATE TRIGGER after_delete_sewa

```
AFTER DELETE ON Sewa
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE Kamar SET status = 'Tersedia' WHERE no_kamar =  
    OLD.no_kamar;
```

```
END //
```

```
DELIMITER ;
```

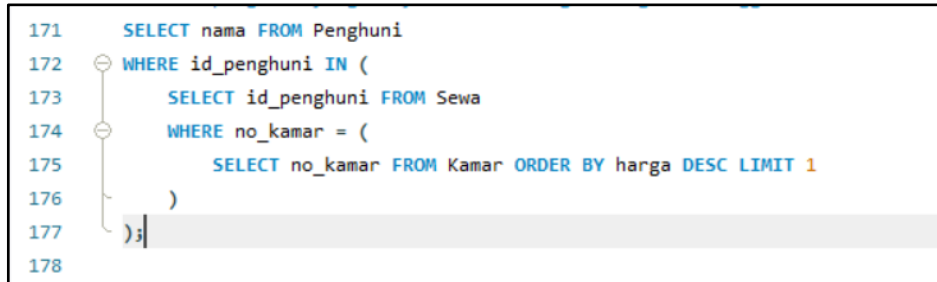
Penjelasan Source Code :

Kode SQL pertama adalah sebuah stored procedure bernama `TambahSewa` yang digunakan untuk menambahkan data penyewaan baru ke dalam sistem kost. Procedure ini menerima lima parameter masukan: `peng_id` (ID penghuni), `kamar_nomor` (nomor kamar), `tgl_masuk` (tanggal masuk), `tgl_keluar` (tanggal keluar), dan `total` (jumlah total pembayaran). Ketika procedure dijalankan, pertama-tama ia akan menyisipkan data baru ke dalam tabel `Sewa`, yaitu mencatat siapa yang menyewa kamar mana, kapan mulai dan berakhir masa sewanya, serta total biayanya. Setelah itu, procedure juga melakukan update pada tabel `Kamar`, mengubah status kamar yang disewa menjadi 'Terisi' berdasarkan nomor kamar yang dimasukkan. Blok kode ini dibungkus dengan `DELIMITER //` untuk memungkinkan penggunaan tanda titik koma (;) di dalam prosedur tanpa mengakhiri perintah SQL utama sebelum waktunya. Setelah prosedur selesai didefinisikan, delimiter dikembalikan ke default (;). Dengan adanya procedure ini, proses pencatatan penyewaan baru dapat dilakukan secara otomatis dan konsisten hanya dengan satu pemanggilan perintah. Kemudian query SQL kedua merupakan definisi trigger bernama `after_delete_sewa` yang berfungsi untuk menjaga konsistensi data antara tabel `Sewa` dan tabel `Kamar`. Trigger ini dijalankan secara otomatis setelah terjadi operasi penghapusan (`DELETE`) pada tabel `Sewa`. Dalam setiap baris data yang dihapus, trigger ini akan mengeksekusi perintah `UPDATE` pada tabel `Kamar`, yaitu mengubah status kamar yang sebelumnya disewa menjadi 'Tersedia'. Informasi nomor kamar yang akan diubah diambil dari nilai `OLD.no_kamar`, yaitu data lama sebelum dihapus. Dengan cara ini, sistem secara otomatis memastikan bahwa jika sebuah penyewaan dihapus (misalnya karena

pembatalan atau selesai), maka kamar yang bersangkutan akan ditandai sebagai tersedia kembali tanpa memerlukan intervensi manual. Trigger ini dibungkus dengan **DELIMITER //** untuk memungkinkan penggunaan titik koma (;) di dalam blok kode.

4. Nested Query

Printscreen Source Code :



```
171 SELECT nama FROM Penghuni
172 WHERE id_penghuni IN (
173     SELECT id_penghuni FROM Sewa
174     WHERE no_kamar = (
175         SELECT no_kamar FROM Kamar ORDER BY harga DESC LIMIT 1
176     )
177 );
178
```

Gambar 17. Nested Query

Source Code :

```
SELECT nama FROM Penghuni
WHERE id_penghuni IN (
    SELECT id_penghuni FROM Sewa
    WHERE no_kamar = (
        SELECT no_kamar FROM Kamar ORDER BY harga DESC LIMIT 1
    )
);
```

Penjelasan Source Code :

Query SQL di atas digunakan untuk menampilkan nama penghuni yang menyewa kamar dengan harga paling mahal. Prosesnya dimulai dengan subquery terdalam, yaitu **SELECT no_kamar FROM Kamar ORDER BY harga DESC LIMIT 1**, yang mengambil nomor kamar dengan harga tertinggi dari tabel Kamar. Hasil nomor kamar ini kemudian digunakan oleh subquery di bagian tengah untuk mencari id_penghuni dari tabel Sewa yang menyewa kamar tersebut. Terakhir, **SELECT nama FROM Penghuni** mengambil nama penghuni berdasarkan id_penghuni yang ditemukan dari subquery sebelumnya. Dengan demikian, query ini secara efektif menampilkan siapa yang sedang menyewa kamar termahal di system

Output :

	nama
▶	Muhammad Ryan Al-Habsy

Gambar 18. Tabel Output Nested Query

5. DCL (Data Control Language)

Printscreen Source Code :

```
-- Buat user khusus admin kost
CREATE USER 'admin_kost'@'localhost' IDENTIFIED BY 'admin123';

-- Beri hak akses hanya untuk membaca data
GRANT SELECT ON SewaKost.* TO 'admin_kost'@'localhost';

-- Beri hak akses penuh ke user operator
CREATE USER 'operator'@'localhost' IDENTIFIED BY 'op123';
GRANT ALL PRIVILEGES ON SewaKost.* TO 'operator'@'localhost';

-- Terapkan perubahan hak akses
FLUSH PRIVILEGES;
```

Gambar 19. DCL

Source Code :

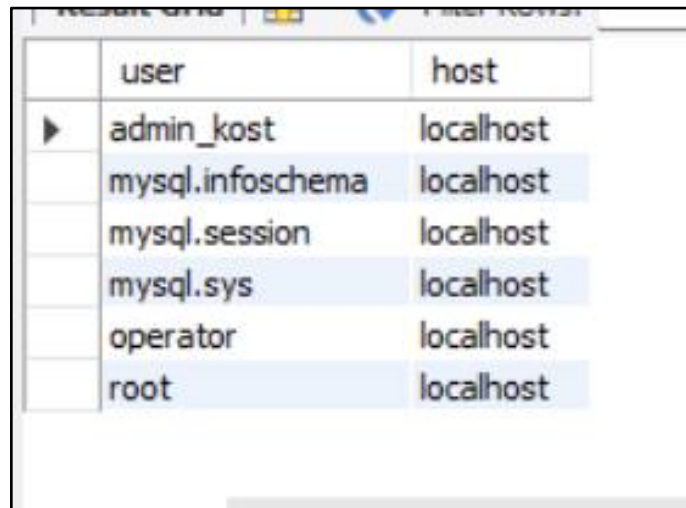
```
CREATE USER 'admin_kost'@'localhost' IDENTIFIED BY 'admin123';
GRANT SELECT ON SewaKost.* TO 'admin_kost'@'localhost';
CREATE USER 'operator'@'localhost' IDENTIFIED BY 'op123';
GRANT ALL PRIVILEGES ON SewaKost.* TO 'operator'@'localhost';
FLUSH PRIVILEGES;
```

Penjelasan Source Code :

Kode SQL ini digunakan untuk mengelola pengguna (**user**) dan hak aksesnya dalam basis data MySQL untuk database bernama **SewaKost**. Pertama, dibuat user baru dengan nama '**admin_kost**' yang hanya bisa login dari localhost, dengan kata sandi '**admin123**'. Setelah itu, user tersebut diberikan hak akses hanya untuk membaca data (perintah **SELECT**) di semua tabel dalam database **SewaKost**, yang artinya user ini tidak bisa mengubah, menambah, atau menghapus data.

Selanjutnya, dibuat user lain dengan nama 'operator', juga dari localhost, dengan password 'op123', dan diberikan hak akses penuh (**ALL PRIVILEGES**) ke seluruh database *SewaKost*, sehingga user ini dapat melakukan semua operasi (membaca, menulis, mengubah, dan menghapus data). Akhirnya, perintah **FLUSH PRIVILEGES** dijalankan untuk menerapkan atau menyegarkan perubahan hak akses yang telah dibuat, agar segera aktif dalam sistem.

Output :



	user	host
▶	admin_kost	localhost
	mysql.infoschema	localhost
	mysql.session	localhost
	mysql.sys	localhost
	operator	localhost
	root	localhost

Gambar 20. Output DCL

Penjelasan Output :

Gambar yang diatas merupakan hasil query dari DCL di database MySQL, yang menunjukkan daftar pengguna (user) yang terdaftar pada sistem basis data beserta host yang diizinkan untuk mengaksesnya. Setelah proses penambahan user baru dilakukan, terlihat bahwa dua pengguna tambahan telah berhasil dibuat, yaitu admin_kost dan operator, yang keduanya memiliki akses dari localhost, artinya mereka hanya dapat melakukan koneksi dari komputer tempat server MySQL dijalankan. Penambahan ini umumnya dilakukan menggunakan perintah **CREATE USER** diikuti dengan **GRANT** untuk memberikan hak akses tertentu. Misalnya, admin_kost dapat diberikan hak akses penuh untuk mengelola seluruh database (seperti **SELECT**, **INSERT**, **UPDATE**, **DELETE**, **CREATE**, **DROP**, dll.), sedangkan operator mungkin hanya diberikan hak terbatas seperti membaca dan memasukkan data. Dengan demikian, struktur manajemen pengguna menjadi lebih terkontrol dan aman, serta mendukung sistem berbasis peran (role-based access control) dalam pengelolaan database Sewa Kost.

KESIMPULAN DAN SARAN

A. Kesimpulan

Struktur dan fungsionalitas yang dibangun telah mencerminkan sistem penyewaan kamar kost yang efisien dan terintegrasi. Database ini terdiri dari beberapa tabel utama seperti Penghuni, Kamar, Sewa, dan Pembayaran, yang masing-masing memiliki peran penting dalam mencatat data pengguna, ketersediaan kamar, transaksi sewa, dan riwayat pembayaran. Relasi antar tabel dibangun melalui foreign key untuk menjaga integritas data. Selain itu, pembuatan stored procedure seperti TambahSewa memudahkan proses penambahan data penyewaan sekaligus memperbarui status kamar secara otomatis. Penggunaan trigger juga menambahkan lapisan otomatisasi dalam manajemen data, khususnya dalam mengubah status kamar menjadi "Tersedia" saat data sewa dihapus. Fitur tambahan seperti view mempermudah dalam penyajian data penyewaan yang sering digunakan, sedangkan pemberian hak akses kepada user admin_kost dan operator menunjukkan perhatian terhadap keamanan dan manajemen peran dalam sistem. Query-query yang digunakan juga memperlihatkan bagaimana informasi penting seperti total tagihan, sisa pembayaran, dan rata-rata harga kamar dapat diperoleh dengan mudah dari database.

B. Saran

Untuk pengembangan lebih lanjut, disarankan agar sistem basis data ini diintegrasikan dengan antarmuka aplikasi (seperti berbasis web atau mobile) agar pengguna non-teknis dapat lebih mudah berinteraksi dengan data. Selain itu, penting untuk menambahkan fitur validasi seperti pengecekan apakah kamar benar-benar tersedia sebelum disewa, atau mencegah duplikasi penyewaan kamar di tanggal yang sama. Dari sisi keamanan, perlu juga diterapkan pengelompokan hak akses yang lebih rinci, seperti hanya memperbolehkan user tertentu untuk melakukan perubahan pada data pembayaran atau penghapusan data sewa. Selanjutnya, audit log atau riwayat aktivitas user juga akan sangat membantu dalam menjaga transparansi dan keamanan sistem. Terakhir, backup data secara berkala harus menjadi bagian dari SOP (Standard Operating Procedure) untuk menghindari kehilangan data penting akibat kesalahan pengguna atau gangguan sistem.

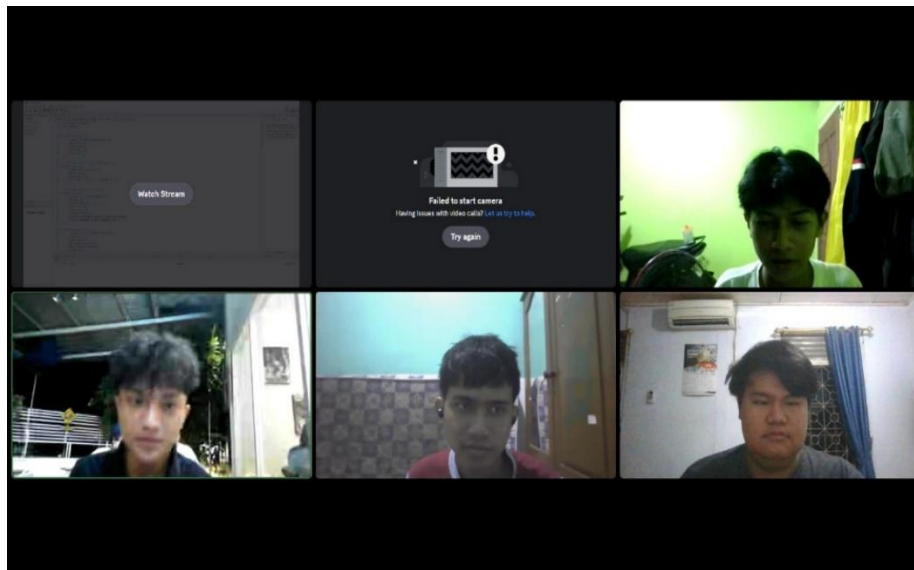
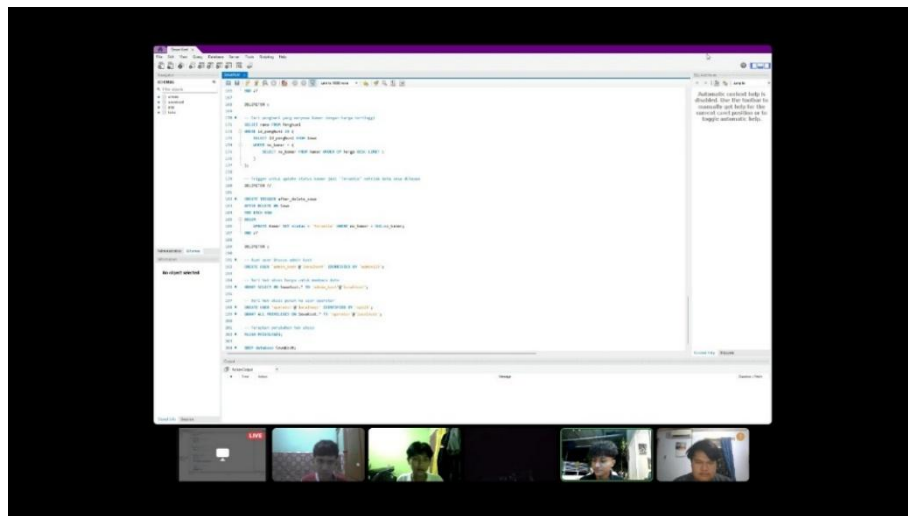
DAFTAR PUSTAKA

- Gunawan, A., Ningsih, S. and Lantana, D.A. (2023) 'Pengantar Basis Data', *Gastronomía ecuatoriana y turismo local.*, 15(2), pp. 1–23.
- Jamaluddin, Samosir, K., S, W., Devia, E., Santoso, L. W., Yuniansyah, Juanaidi, Nursari, S. R. C., Azizah, N., & Saputra, M. H. (2022). BUKU (Book Chapter)-Sistem Basis Data (Elmi Devia)_oke. In *PT. Global Eksekutif Teknologi*.
- Kalsum Siregar, U., Arbaim Sitakar, T., Haramain, S., Nur Salamah Lubis, Z., Nadhirah, U., & Sains dan Teknologi, F. (2024). Pengembangan database Management system menggunakan My SQL. *SAINTEK: Jurnal Sains, Teknologi & Komputer*, 1(1), 8–12.
- Maulana, H. (2022) 'Analisis Dan Perancangan Sistem Replikasi Database Mysql Dengan Menggunakan Vmware Pada Sistem Operasi Open Source', *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, 1(1), pp. 32–37. Available at: <https://doi.org/10.30743/infotekjar.v1i1.37>.
- Pratama, A. (2020) 'Pengenalan MySQL Workbench', *Ilmukomputer.Org*, pp. 1–5
- Sabbrina, A., Sufa, A. oktavia, Ritonga, D. putra, Siregar, E. R. sari, & . N. (2023). Pengenalan Konsep Dasar Dan Penggunaan Database Manajemen Sistem (DBMS). *Jurnal Sains Dan Teknologi (JSIT)*, 3(3), 224–232. <https://doi.org/10.47233/jsit.v3i2.907>
- Ummah, M.S. (2019) *Title, Sustainability (Switzerland)*. Available at: http://scioteca.caf.com/bitstream/handle/123456789/1091/RED2017-Eng-8ene.pdf?sequence=12&isAllowed=y%0Ahttp://dx.doi.org/10.1016/j.regs-ciurbeco.2008.06.005%0Ahttps://www.researchgate.net/publication/305320484_SISTEM_PEMBETUNGAN_TERPUSAT_STRATEGI_MELESTARI.

JOBDESK

1. Habib Al Qodri (G1A023047) : Pembuatan SQL
2. Ricardo Gellael (G1A023061) : Pembuatan Laporan
3. Agyl Wendi Pratama (G1A023087) : Pembuatan SQL
4. Juan Agraprana Putra (G1A023085) : Pembuatan Laporan
5. M. Ryan Alhabsy (G1A023093) : Pembuatan ERD

DOKUMENTASI Pengerjaan





**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN
RISET, DAN TEKNOLOGI
UNIVERSITAS BENGKULU
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA**

Jl. Wr. Supratman Kandang Limun, Bengkulu Bengkulu
38371 A Telp: (0736) 344087, 22105 - 227

**LEMBAR ASISTENSI
PROYEK BASIS DATA LANJUT**

Nama Mahasiswa : 1. Habib Al Qodri (G1A023047)
2. Ricardo Gellael (G1A023061)
3. Agyl Wendi Pratama (G1A023087)
4. Juan Agraprana Putra (G1A023085)
5. M. Ryan Alhabsy (G1A023093)

Dosen : 1. Dr. Endina Putri Purwandari, S. T., M.Kom
2. Ir. Tiara Eka Putri, S.T., M.Kom.

Asisten Dosen : 1. Merly Yuni Purnama (G1A022006)
2. Reksi Hendra Pratama (G1A022032)
3. Sinta Ezra Wati Gulo (G1A022040)
4. Fadlan Dwi Febrio (G1A022051)
5. Torang Four Yones Manullang (G1A022052)
6. Wahyu Ozorah Manurung (G1A022060)
7. Shalaudin Muhammad Sah (G1A022070)
8. Dian Ardiyanti Saputri (G1A022084)

Laporan Tugas Besar	Catatan dan Tanda Tangan
Laporan Tugas Besar	