

# **LAPORAN PRAKTIKUM DATABASE ANTRIAN PASIEN (KLINIK)**



DISUSUN OLEH :

**Hansen Julio**

312210523

**Nicky Pascal Tambunan**

312210474

**Serius Ndruru**

312210508

**Stephen Pratama Kurnia**

312210635

Dosen Pengajar :

**Agung Nugroho, S.Kom., M.Kom.**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PELITA BANGSA**

**2023**

## KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan yang Maha Esa atas berkahnya tersusunlah laporan ini dengan harapan dapat menjadi petunjuk mengenai praktikum yang kami lakukan.

Laporan praktikum ini dibuat untuk merekam dan menganalisis implementasi sebuah database antrian pasien. Tujuan dari pembuatan database ini adalah untuk mengatur dan memantau antrian pasien dalam suatu unit pelayanan kesehatan.

Database antrian pasien memiliki beberapa entitas utama, antara lain tabel Pasien yang mencatat informasi tentang pasien seperti ID, nama, alamat, dan tanggal lahir. Selain itu, terdapat tabel Jenis\_Pelayanan yang mencatat jenis-jenis pelayanan yang disediakan, serta tabel Loker\_Pelayanan yang menghubungkan antara jenis pelayanan dengan loket pelayanan yang tersedia.

Untuk mengatur antrian pasien, terdapat tabel Antrian yang mencatat informasi tentang nomor antrian, tanggal antrian, pasien yang sedang antri, serta loket pelayanan yang dituju oleh pasien tersebut. Tabel Laporan\_Transaksi digunakan untuk mencatat transaksi pembayaran yang terkait dengan antrian pasien.

Dalam laporan praktikum ini, kami akan memperkenalkan struktur database antrian pasien, beserta contoh pengisian data ke dalam tabel-tabel yang telah dibuat. Selanjutnya, kami juga akan memberikan beberapa contoh query SQL untuk mengambil informasi spesifik dari database, seperti menampilkan jenis pelayanan yang digunakan oleh pasien tertentu atau menampilkan tagihan dari pasien-pasien dengan kriteria tertentu.

Semoga laporan praktikum ini dapat memberikan pemahaman yang lebih baik tentang implementasi database antrian pasien, serta memberikan manfaat dalam pengelolaan dan analisis data di bidang pelayanan kesehatan.

## DAFTAR ISI

KATA PENGANTAR .....	i
DAFTAR ISI .....	ii
BAB I PENDAHULUAN .....	1
1.1. LATAR BELAKANG .....	1
1.2. TUJUAN .....	1
1.3. MANFAAT .....	1
BAB II PEMBAHASAN .....	2
2.1. MEMBUAT DATABASE BARU .....	2
2.2. MEMBUAT TABEL DAN FIELD .....	3
2.3. MENGISI TABEL DENGAN QUERY SQL CRUD .....	5
2.4. CONTOH OPERASI JOIN TABEL .....	8
BAB III PENUTUP .....	10
3.1. KESIMPULAN .....	10
3.2. SARAN .....	10

# **BAB I**

## **PENDAHULUAN**

### **1.1. LATAR BELAKANG**

Pembuatan database antrian pasien bertujuan untuk memberikan solusi yang terorganisir dalam mengelola data pasien, memantau antrian, dan melacak transaksi pembayaran yang terkait. Dengan menggunakan database, informasi pasien dapat diakses dengan mudah dan cepat, meminimalkan risiko kesalahan dalam proses administrasi, serta mempercepat proses pelayanan di loket-loket yang berbeda.

Database antrian pasien juga memungkinkan untuk dilakukannya analisis data yang lebih mendalam, seperti pemantauan rata-rata waktu tunggu pasien, identifikasi pola antrian yang sering terjadi, serta evaluasi efisiensi loket pelayanan. Informasi ini dapat digunakan untuk pengambilan keputusan yang lebih baik dalam mengoptimalkan pelayanan dan meningkatkan kepuasan pasien.

Dalam laporan praktikum ini, kami akan memaparkan implementasi database antrian pasien, termasuk struktur tabel, pengisian data, dan contoh-contoh query SQL untuk menampilkan informasi yang relevan. Diharapkan laporan ini dapat memberikan pemahaman yang lebih baik tentang manfaat dan penggunaan database antrian pasien dalam konteks pelayanan kesehatan, serta mendorong pengembangan solusi yang lebih baik di masa depan.

### **1.2. TUJUAN**

Tujuan pembuatan laporan ini yaitu memenuhi tugas praktikum pembuatan database antrian pasien beserta langkah – langkahnya. Dan tentunya agar mahasiswa dapat membuat database yang berguna.

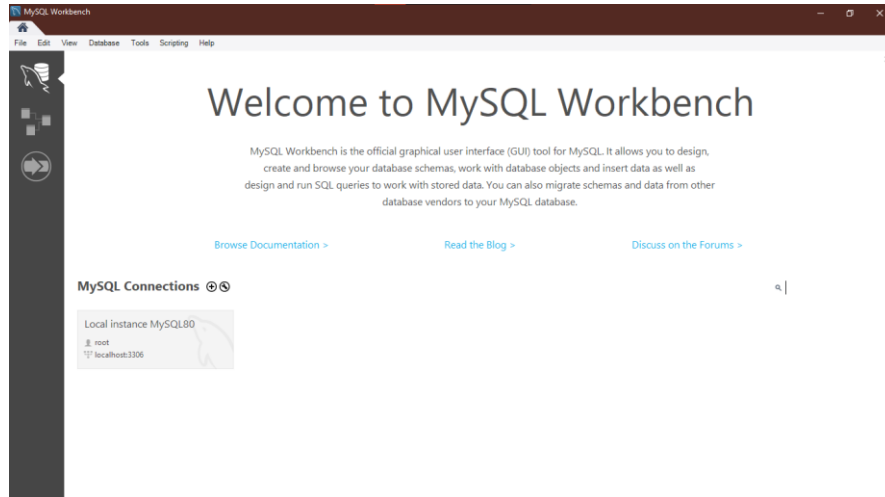
### **1.3. MANFAAT**

1. Mahasiswa dapat mengetahui cara membuat database.
2. Mahasiswa dapat mengetahui cara membuat tabel dalam database.
3. Mahasiswa dapat menggunakan query dalam database.
4. Mahasiswa dapat melakukan CRUD dan JOIN pada tabel database.

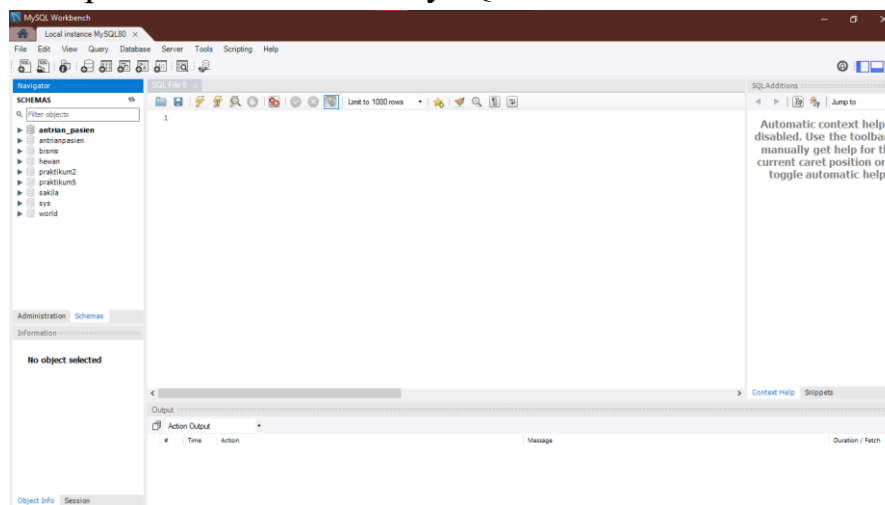
## BAB II PEMBAHASAN

### 2.1. MEMBUAT DATABASE BARU

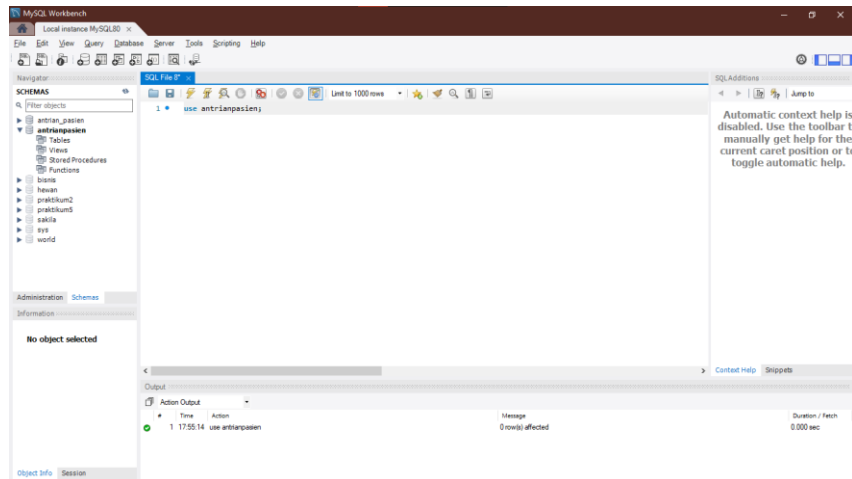
#### 1. Membuka MySQL Workbench



#### 2. Klik pada Local Instance MySQL80



### 3. Menggunakan database yang baru dibuat



## 2.2. MEMBUAT TABEL DAN FIELD

### 1. Membuat tabel Pasien

```
CREATE TABLE Pasien (  
    id_pasien INT PRIMARY KEY,  
    nama_pasien VARCHAR(255),  
    alamat VARCHAR(255),  
    tanggal_lahir DATE  
);
```

	id_pasien	nama_pasien	alamat	tanggal_lahir
*	NULL	NULL	NULL	NULL

### 2. Membuat tabel Jenis\_Pelayanan

```
CREATE TABLE Jenis_Pelayanan (  
    id_pelayanan INT PRIMARY KEY,  
    nama_pelayanan VARCHAR(255)  
);
```

	id_pelayanan	nama_pelayanan
*	NULL	NULL

### 3. Membuat tabel Locket\_Pelayanan

```
CREATE TABLE Locket_Pelayanan (  
    id_loket INT PRIMARY KEY,  
    nama_loket VARCHAR(255),  
    id_pelayanan INT,  
    FOREIGN KEY (id_pelayanan) REFERENCES Jenis_Pelayanan(id_pelayanan)  
);
```

	id_loket	nama_loket	id_pelayanan
*	NULL	NULL	NULL

### 4. Membuat tabel Antrian

```
CREATE TABLE Antrian (  
    id_antrian INT PRIMARY KEY,  
    id_loket INT,  
    id_pasien INT,  
    tgl_antrian DATE,  
    nomor_antrian INT,  
    FOREIGN KEY (id_loket) REFERENCES Locket_Pelayanan(id_loket),  
    FOREIGN KEY (id_pasien) REFERENCES Pasien(id_pasien)  
);
```

	id_antrian	id_loket	id_pasien	tgl_antrian	nomor_antrian
*	NULL	NULL	NULL	NULL	NULL

## 5. Membuat tabel Laporan\_Transaksi

```
CREATE TABLE Laporan_Transaksi (  
    id_transaksi INT PRIMARY KEY,  
    id_antrian INT,  
    tgl_transaksi DATE,  
    jumlah_pembayaran VARCHAR(50),  
    FOREIGN KEY (id_antrian) REFERENCES Antrian(id_antrian)  
);
```

	id_transaksi	id_antrian	tgl_transaksi	jumlah_pembayaran
*	NULL	NULL	NULL	NULL

## 2.3. MENGISI TABEL DENGAN QUERY SQL CRUD

### 1. Mengisi tabel Pasien

```
INSERT INTO pasien  
VALUES  
    ('7123456', 'Christian', 'Bekasi', '2004-03-25'),  
    ('7134567', 'Anasyah ', 'Cikarang', '1990-01-20'),  
    ('7145678', 'Syahbar Udin', 'Jakarta', '1998-05-10'),  
    ('7156789', 'Vannesa', 'Bekasi', '2001-12-01'),  
    ('7167890', 'Roy Martin', 'Cikarang', '1995-08-09');
```

	id_pasien	nama_pasien	alamat	tanggal_lahir
▶	7123456	Christian	Bekasi	2004-03-25
	7134567	Anasyah	Cikarang	1990-01-20
	7145678	Syahbar Udin	Jakarta	1998-05-10
	7156789	Vannesa	Bekasi	2001-12-01
	7167890	Roy Martin	Cikarang	1995-08-09
*	NULL	NULL	NULL	NULL



## 2. Mengisi tabel Jenis\_Pelayanan

```
INSERT INTO jenis_pelayanan
VALUES      ('6312345', 'Pemeriksaan UMUM'),
            ('6323456', 'Pemeriksaan Gigi'),
            ('6345467', 'Gawat Darurat'),
            ('6345678', 'KB (Keluarga Berencana)'),
            ('6356789', 'Farmasi/Apotek');
```

	id_pelayanan	nama_pelayanan
►	6312345	Pemeriksaan UMUM
	6323456	Pemeriksaan Gigi
	6345467	Gawat Darurat
	6345678	KB (Keluarga Berencana)
	6356789	Farmasi/Apotek

## 3. Mengisi tabel Loker\_Pelayanan

```
INSERT INTO Loker_Pelayanan
VALUES      ('5132456', 'Loker 1', '6345467'),
            ('5143567', 'Loker 2', '6345678'),
            ('5154678', 'Loker 3', '6356789');
```

	id_loket	nama_loket	id_pelayanan
►	5132456	Loker 1	6345467
	5143567	Loker 2	6345678
	5154678	Loker 3	6356789

#### 4. Mengisi tabel Antrian

INSERT INTO antrian

VALUES ('92123456', '5132456', '7123456', '2023-06-05', '001'),  
( '92434567', '5154678', '7134567', '2023-06-05', '002'),  
( '92545678', '5143567', '7156789', '2023-06-05', '003'),  
( '92656789', '5132456', '7167890', '2023-06-05', '004'),  
( '92767890', '5143567', '7145678', '2023-06-05', '005');

	id_antrian	id_loket	id_pasien	tgl_antrian	nomor_antrian
▶	92123456	5132456	7123456	2023-06-05	1
	92434567	5154678	7134567	2023-06-05	2
	92545678	5143567	7156789	2023-06-05	3
	92656789	5132456	7167890	2023-06-05	4
	92767890	5143567	7145678	2023-06-05	5

#### 5. Mengisi tabel laporan\_transaksi

INSERT INTO laporan\_transaksi

VALUES ('83123456', '92123456', '2023-06-05', '850.000'),  
( '83234567', '92434567', '2023-06-05', '1.000.000'),  
( '83435678', '92545678', '2023-06-05', '500.000'),  
( '83656789', '92656789', '2023-06-05', '850.000'),  
( '83767890', '92767890', '2023-06-05', '500.000');

	id_transaksi	id_antrian	tgl_transaksi	jumlah_pembayaran
▶	83123456	92123456	2023-06-05	850.000
	83234567	92434567	2023-06-05	1.000.000
	83435678	92545678	2023-06-05	500.000
	83656789	92656789	2023-06-05	850.000
	83767890	92767890	2023-06-05	500.000

## 2.4. CONTOH OPERASI JOIN TABEL

1. Menampilkan jumlah tagihan pasien Bernama Vannesa

```
SELECT lt.jumlah_pembayaran  
FROM Pasien p  
JOIN Antrian a ON a.id_pasien = p.id_pasien  
JOIN Laporan_Transaksi lt ON lt.id_antrian = a.id_antrian  
WHERE p.nama_pasien = 'Vannesa';
```

	jumlah_pembayaran
▶	500.000

2. Menampilkan Pasien yang menggunakan pelayanan Gawat Darurat

```
SELECT p.nama_pasien  
FROM Pasien p  
JOIN Antrian a ON a.id_pasien = p.id_pasien  
JOIN Loker_Pelayanan lp ON lp.id_loket = a.id_loket  
JOIN Jenis_Pelayanan jp ON jp.id_pelayanan = lp.id_pelayanan  
WHERE jp.nama_pelayanan = 'Gawat Darurat';
```

	nama_pasien
▶	Christian
	Roy Martin

3. Menampilkan Pasien yang menggunakan loket ke 2 untuk melakukan transaksi

```
SELECT p.nama_pasien, lt.jumlah_pembayaran  
FROM Pasien p  
JOIN Antrian a ON a.id_pasien = p.id_pasien  
JOIN Loker_Pelayanan lp ON lp.id_loket = a.id_loket  
JOIN Jenis_Pelayanan jp ON jp.id_pelayanan = lp.id_pelayanan  
JOIN Laporan_Transaksi lt ON lt.id_antrian = a.id_antrian  
WHERE lp.id_loket = '5143567';
```

	nama_pasien	jumlah_pembayaran
►	Vannesa	500.000
	Syahbar Udin	500.000

4. Menampilkan pasien dan tagihannya yang lahir pada tahun 1990 sampai 2001

```
SELECT p.nama_pasien, lt.jumlah_pembayaran, p.tanggal_lahir
```

```
FROM Pasien p
```

```
JOIN Antrian a ON a.id_pasien = p.id_pasien
```

```
JOIN Laporan_Transaksi lt ON lt.id_antrian = a.id_antrian
```

```
WHERE p.tanggal_lahir BETWEEN '1990-01-01' AND '2001-12-31';
```

	nama_pasien	jumlah_pembayaran	tanggal_lahir
►	Anasyah	1.000.000	1990-01-20
	Syahbar Udin	500.000	1998-05-10
	Vannesa	500.000	2001-12-01
	Roy Martin	850.000	1995-08-09

## **BAB III PENUTUP**

### **3.1. KESIMPULAN**

Pembuatan database antrian pasien bertujuan untuk memberikan solusi yang terorganisir dalam mengelola data pasien, memantau antrian, dan melacak transaksi pembayaran yang terkait. Dengan menggunakan database, informasi pasien dapat diakses dengan mudah dan cepat, meminimalkan risiko kesalahan dalam proses administrasi, serta mempercepat proses pelayanan di loket-loket yang berbeda.

Untuk membuat tabel, kita bisa menggunakan bahasa SQL dan menuliskan secara manual sesuai dengan sintaks bahasa SQL atau kita bisa membuat tabel dengan langsung menggunakan GUI yang tersedia di server MySQL.

### **3.2. SARAN**

Ada pun saran yang dapat kami sampaikan yaitu sebaiknya modul yang telah diberikan lebih baik di perbaharui sesuai dengan software terbaru, agar mahasiswa tidak kebingungan saat mengerjakan Tugas ataupun Praktikum.