

MAKALAH PROJECT APLIKASI BASIS DATA  
“RAWAT INAP RUMAH SAKIT ASRAMA HAJI”

Disusun Oleh:

Kelompok 7-19.2A.11

Rafif Setyo Nugroho	(19220737)
Matius Dimas Prasetia	(19220918)
Irfan Indra Kusuma	(19221071)
Muhammad Hidayatulloh	(19220042)
Muhammad Reza Pahlevy	(19220021)



**PROGRAM STUDI SISTEM INFORMASI**  
**UNIVERSITAS BINA SARANA INFORMATIKA (BSI) JATIWARINGIN**

## **KATA PENGANTAR**

Dalam era digital ini, teknologi telah memainkan peran yang semakin penting dalam menyediakan perawatan kesehatan yang berkualitas dan efisien. Aplikasi basis data yang didedikasikan untuk industri rumah sakit menjadi landasan penting dalam mengelola informasi pasien, memantau perawatan medis, dan meningkatkan efisiensi operasional.

Dengan menggunakan teknologi ini, tim medis dapat dengan mudah mengakses informasi yang relevan, memperoleh wawasan yang lebih baik tentang kondisi pasien, dan membuat keputusan medis yang tepat. Keuntungan utama dari aplikasi basis data bertemakan rawat inap rumah sakit adalah peningkatan efisiensi dan produktivitas

Dengan penggunaan aplikasi basis data bertemakan rawat inap rumah sakit, diharapkan bahwa perawatan medis akan menjadi lebih terkoordinasi, efisien, dan berkualitas. Kesehatan dan kesejahteraan pasien menjadi prioritas utama, dan teknologi ini membantu memastikan bahwa informasi yang diperlukan tersedia dengan mudah dan akurat bagi para profesional medis.

## DAFTAR ISI

KATA PENGANTAR .....	1
DAFTAR ISI.....	2
BAB I PENDAHULUAN	
A. KONSEP DASAR DATABASE.....	3
B. PENGERTIAN ERD DAN LRS.....	7
BAB II PEMBAHASAN	
A. PEMBUATAN DATABASE MELALUI COMMAND PROMPT .....	15
B. LANGKAH-LANGKAH PEMBUATAN DATABASE MENGGUNAKAN CMD .....	15
BAB III PENUTUP	
A. KESIMPULAN.....	20
B. SARAN .....	20

# **BAB I**

## **PENDAHULUAN**

### **A. KONSEP DASAR DATABASE**

Konsep dasar database adalah serangkaian prinsip dan elemen yang membentuk dasar dari struktur dan fungsi basis data. Pemahaman konsep dasar ini penting untuk merancang, mengelola, dan memanipulasi basis data dengan efisien.

Konsep dasar database mencakup beberapa elemen kunci yang membentuk struktur dasar dan prinsip pengelolaan data dalam sistem database. Berikut adalah beberapa konsep dasar database yang penting untuk dipahami:

1. Data: Data adalah informasi yang disimpan dalam database. Data dapat berupa angka, teks, gambar, audio, atau tipe data lainnya yang relevan dengan aplikasi atau sistem yang menggunakan database.

2. Tabel: Tabel adalah struktur dasar yang digunakan untuk mengorganisir data dalam database. Tabel terdiri dari kolom (field) yang mewakili jenis data yang berbeda dan baris (record) yang berisi nilai-nilai spesifik yang terkait dengan kolom tersebut.

3. Kolom (Field): Kolom adalah komponen tabel yang mendefinisikan jenis data yang akan disimpan di dalamnya. Setiap kolom memiliki nama unik dan tipe data yang sesuai (seperti teks, angka, tanggal, dll.).

4. Baris (Record): Baris atau record adalah entitas tunggal dalam tabel yang berisi nilai-nilai spesifik yang terkait dengan kolom-kolom dalam tabel tersebut. Baris-baris ini merepresentasikan entitas atau objek individu dalam domain aplikasi atau sistem yang menggunakan database.

5. Kunci (Key): Kunci adalah nilai unik yang digunakan untuk mengidentifikasi secara unik setiap baris dalam tabel. Kunci dapat digunakan untuk mencari, mengurutkan, dan menghubungkan data antara tabel yang berbeda.

6. Relasi antartabel: Relasi atau hubungan antartabel menggambarkan cara data terkait dan saling berhubungan antara tabel. Relasi ini dapat diwujudkan melalui penggunaan kunci asing (foreign key) yang menghubungkan kolom dalam satu tabel dengan kolom dalam tabel lainnya.

7. Kunci Asing (Foreign Key): Kunci asing adalah kolom dalam sebuah tabel yang merujuk pada nilai kunci utama (primary key) di tabel lain. Kunci asing digunakan untuk membangun relasi antartabel dan memfasilitasi integritas referensial, yang memastikan keberadaan data yang konsisten dan terkait di antara tabel-tabel tersebut.

8. Normalisasi: Normalisasi adalah proses desain database yang bertujuan untuk mengurangi redundansi dan meningkatkan integritas data. Normalisasi melibatkan pemisahan data ke dalam tabel-tabel yang lebih kecil dan lebih terorganisir, mengikuti aturan normalisasi seperti bentuk normal pertama (1NF), bentuk normal kedua (2NF), dan seterusnya.

9. Query: Query adalah pernyataan yang digunakan untuk mengambil, memperbarui, atau memanipulasi data dalam database. Query memungkinkan pengguna untuk mencari data yang spesifik, melakukan perhitungan, menyaring data, dan melakukan operasi lainnya pada database.

10. Integritas Data: Integritas data mengacu pada keabsahan, konsistensi, dan akurasi data dalam database. Integritas data dapat diterapkan melalui aturan, batasan, dan ketergantungan yang didefinisikan dalam struktur database, seperti aturan unik, aturan referensial, dan batasan lainnya.

Memahami konsep dasar database adalah langkah awal yang penting dalam merancang, mengelola, dan menggunakan sistem database dengan efektif. Konsep-konsep ini membentuk dasar pemahaman tentang bagaimana data disimpan, diorganisir, dan diakses dalam lingkungan database.

Database memiliki sejumlah kelebihan dan kekurangan yang perlu dipertimbangkan saat merancang dan menggunakan sistem database. Berikut adalah beberapa kelebihan dan kekurangan umum dari penggunaan database:

#### Kelebihan Database:

1. Pemeliharaan Data Terpusat: Database memungkinkan penyimpanan data terpusat yang mudah diakses dan dikelola. Data dapat disimpan dalam tabel terstruktur, memungkinkan integrasi data dan pengelolaan yang lebih efisien.

2. **Konsistensi Data:** Database memungkinkan penerapan aturan dan ketergantungan yang memastikan konsistensi data. Integritas referensial, aturan unik, dan konstrain lainnya dapat diterapkan untuk memastikan bahwa data tetap konsisten dan valid.
3. **Akses dan Manipulasi Data yang Efisien:** Database menyediakan mekanisme query yang kuat untuk mengambil dan memanipulasi data. Dengan menggunakan bahasa query seperti SQL, pengguna dapat mengambil informasi yang spesifik, melakukan perhitungan, menyaring data, dan melaksanakan operasi lainnya dengan efisien.

#### Kekurangan Database:

1. **Biaya:** Mengelola sistem database dapat melibatkan biaya yang signifikan. Ini termasuk biaya perangkat keras, perangkat lunak database, lisensi, administrasi, dan keahlian untuk mengelola dan mengoptimalkan sistem database.
2. **Kompleksitas:** Sistem database bisa kompleks dan membutuhkan pemahaman yang mendalam tentang struktur data, desain database, dan bahasa query seperti SQL. Dibutuhkan keahlian dan waktu yang signifikan untuk merancang, mengelola, dan memelihara sistem database yang efektif.
3. **Ketergantungan Terhadap Sistem:** Ketika menggunakan database, organisasi menjadi sangat tergantung pada keberlanjutan, ketersediaan, dan kinerja sistem database. Gangguan atau kegagalan pada sistem database dapat berdampak negatif pada akses dan keberlanjutan layanan.

## B. PENGERTIAN ERD DAN LRS

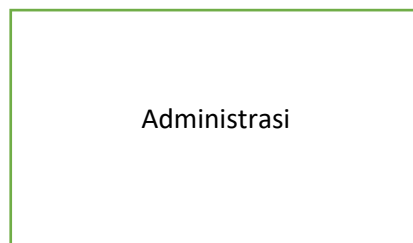
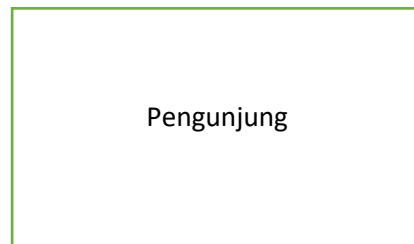
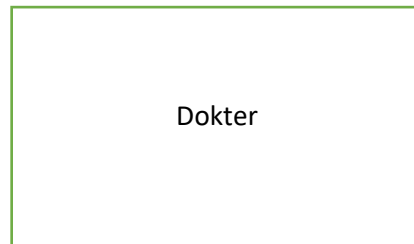
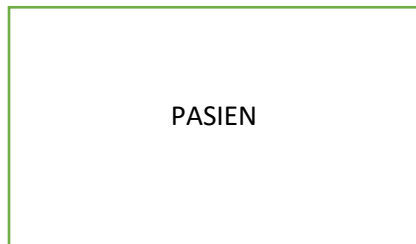
### 1. Entity Relational Diagram (ERD)

ERD (Entity-Relationship Diagram) adalah diagram yang digunakan untuk menggambarkan hubungan antara entitas dalam sebuah basis data. ERD adalah bagian penting dalam perancangan sistem database karena membantu menggambarkan struktur data dan hubungannya dengan jelas dan intuitif.

### 2. Tahapan ERD

#### a. Tahap 1: Penentuan Entitas

Entitas merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entitas biasanya digambarkan dengan persegi panjang



Pasien:

- Id (primary key)
- Nm\_pasien
- Alamat pasien
- Nohp\_pasien
- Id\_dokter
- Id\_ruangan
- Id\_transaksi
- Keterangan

Dokter:

- Id (primary key)
- Nm\_dokter
- Alamat\_dokter
- Nohp\_dokter

Ruangan:

- Id (primary key)
- Nm\_ruangan

Pengunjung:

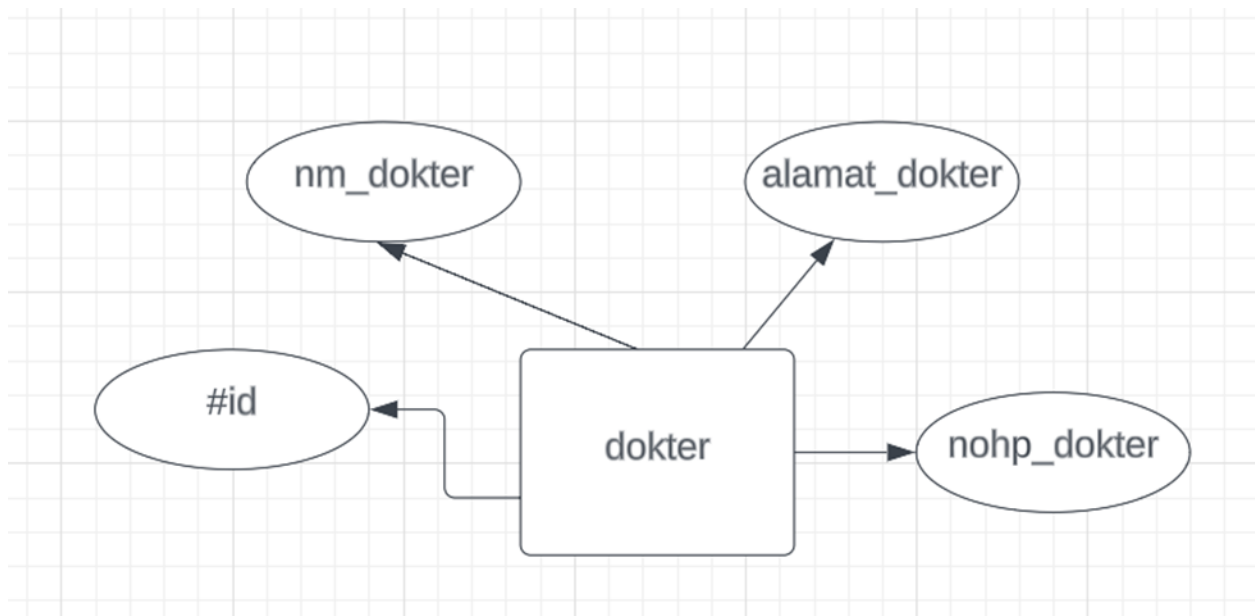
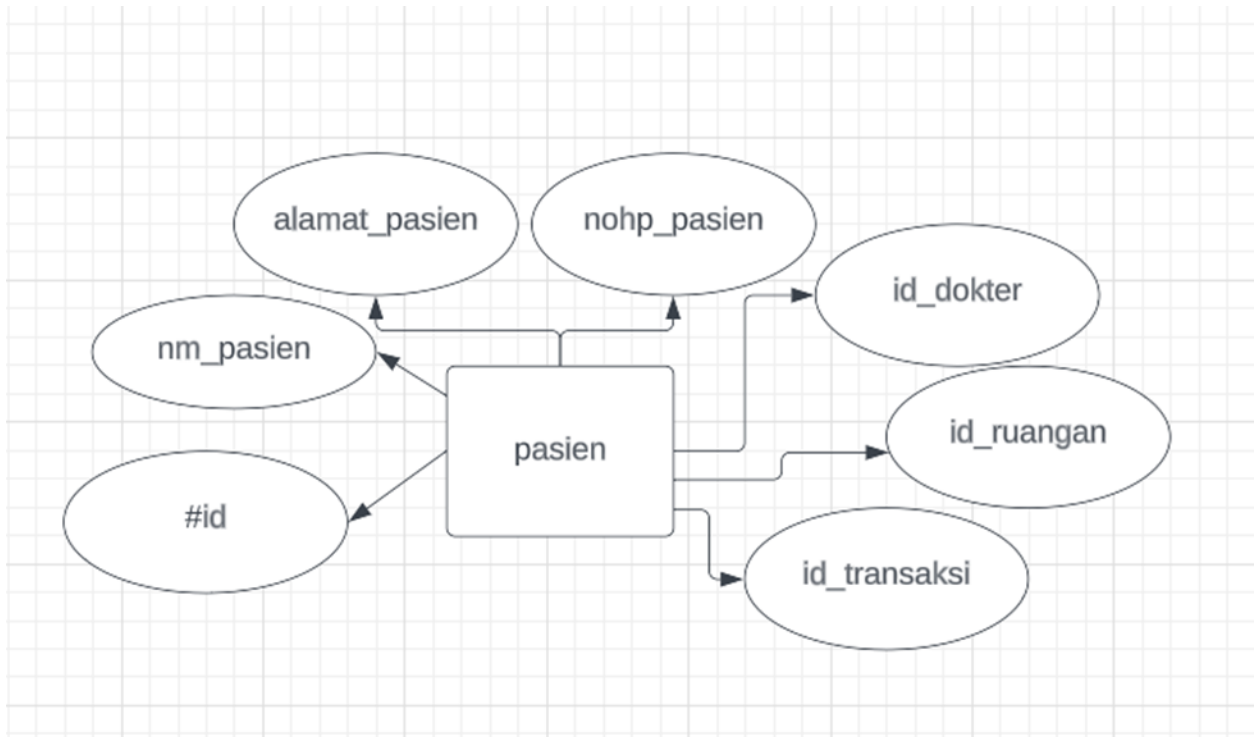
- Id (primary key)
- Nm\_pengunjung
- Id\_pasien
- Alamat\_pengunjung

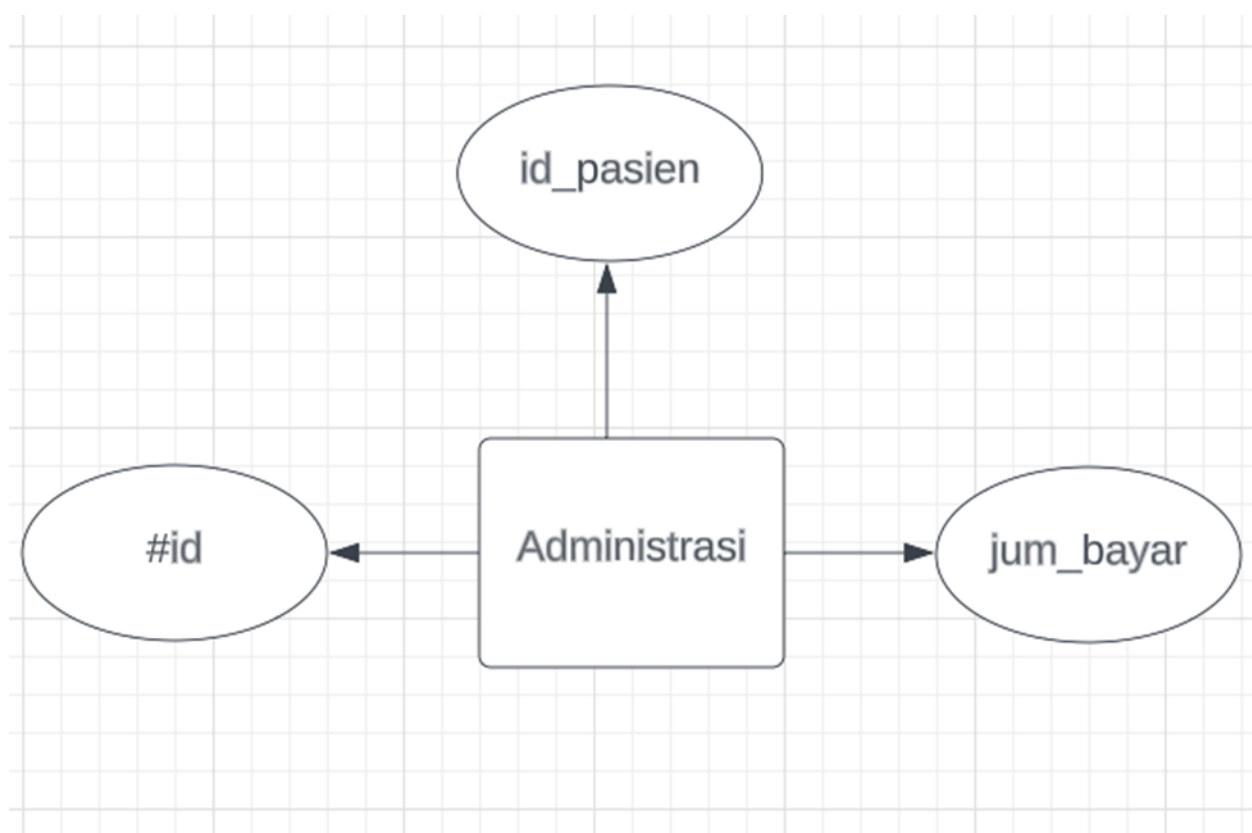
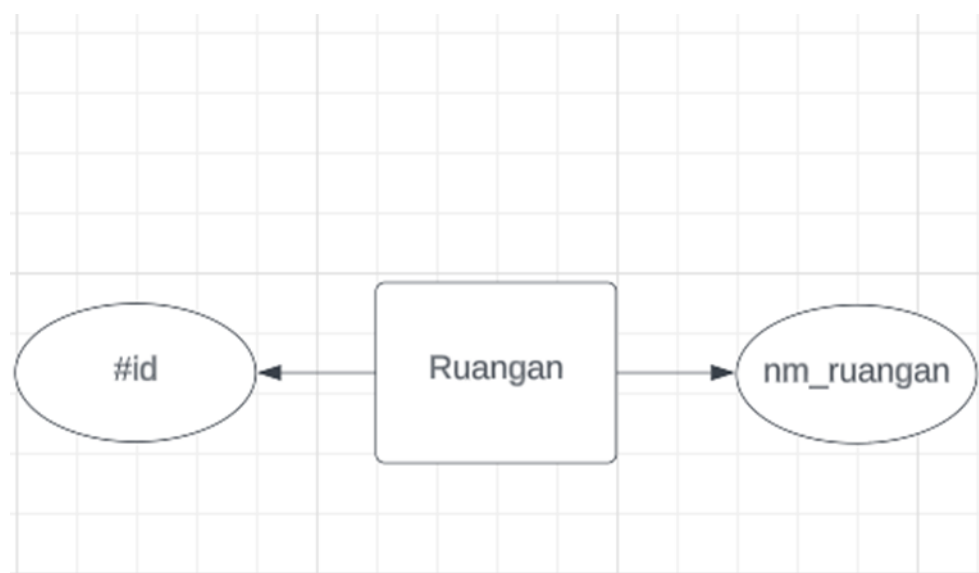
Administrasi:

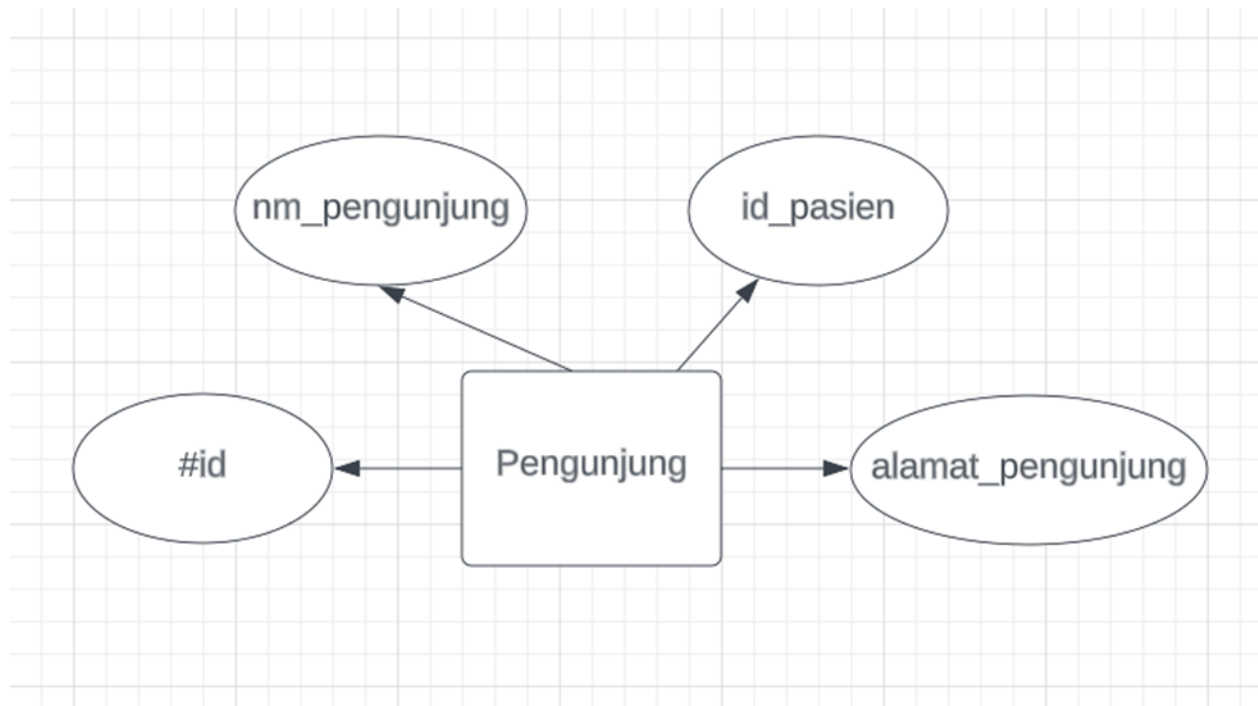
- Id (primary key)
- Id\_pasien
- Jum\_bayar



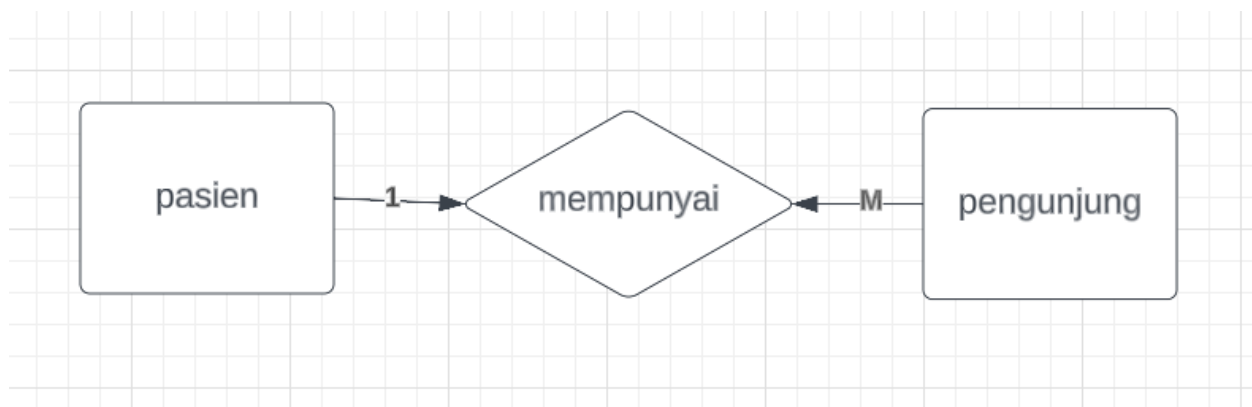
b. Tahap 2 : Penentuan Atribut







### C. Tahap 3 : Penentuan Kardinalitas Relasi

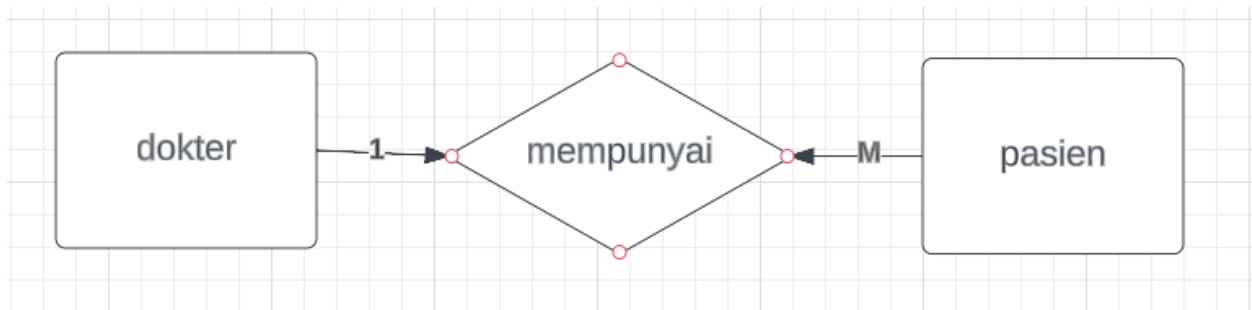


Tabel Utama: Pasien

Tabel Kedua : Pengunjung

Relationship : one to many

Attribute penghubung : id\_pasien (FK id di pasien untuk table pengunjung)

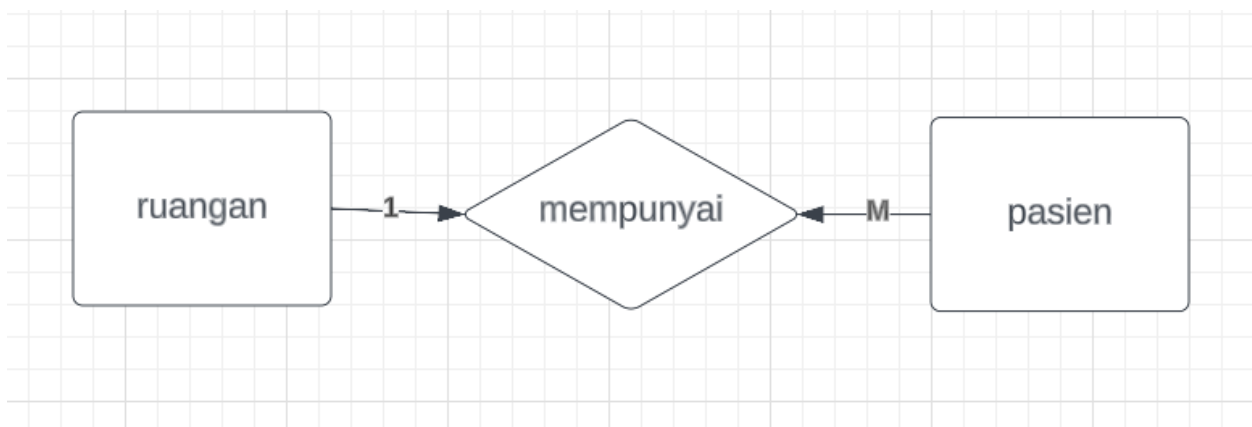


Tabel utama : dokter

Table kedua: pasien

Relationship: one to many

Atribut penghubung: id\_dokter(FK Id di dokter untuk table pasien)

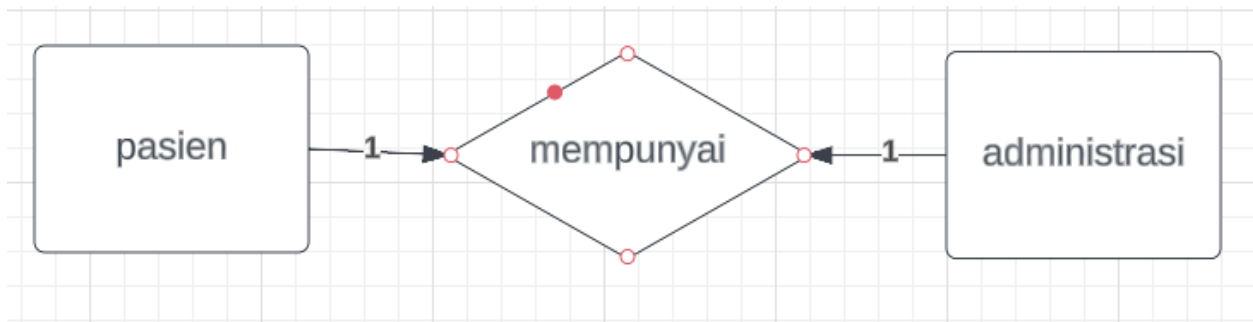


Tabel utama : ruangan

Table kedua: pasien

Relationship: one to many

Atribut penghubung: id\_ruangan(FK Id di ruangan untuk table pasien)



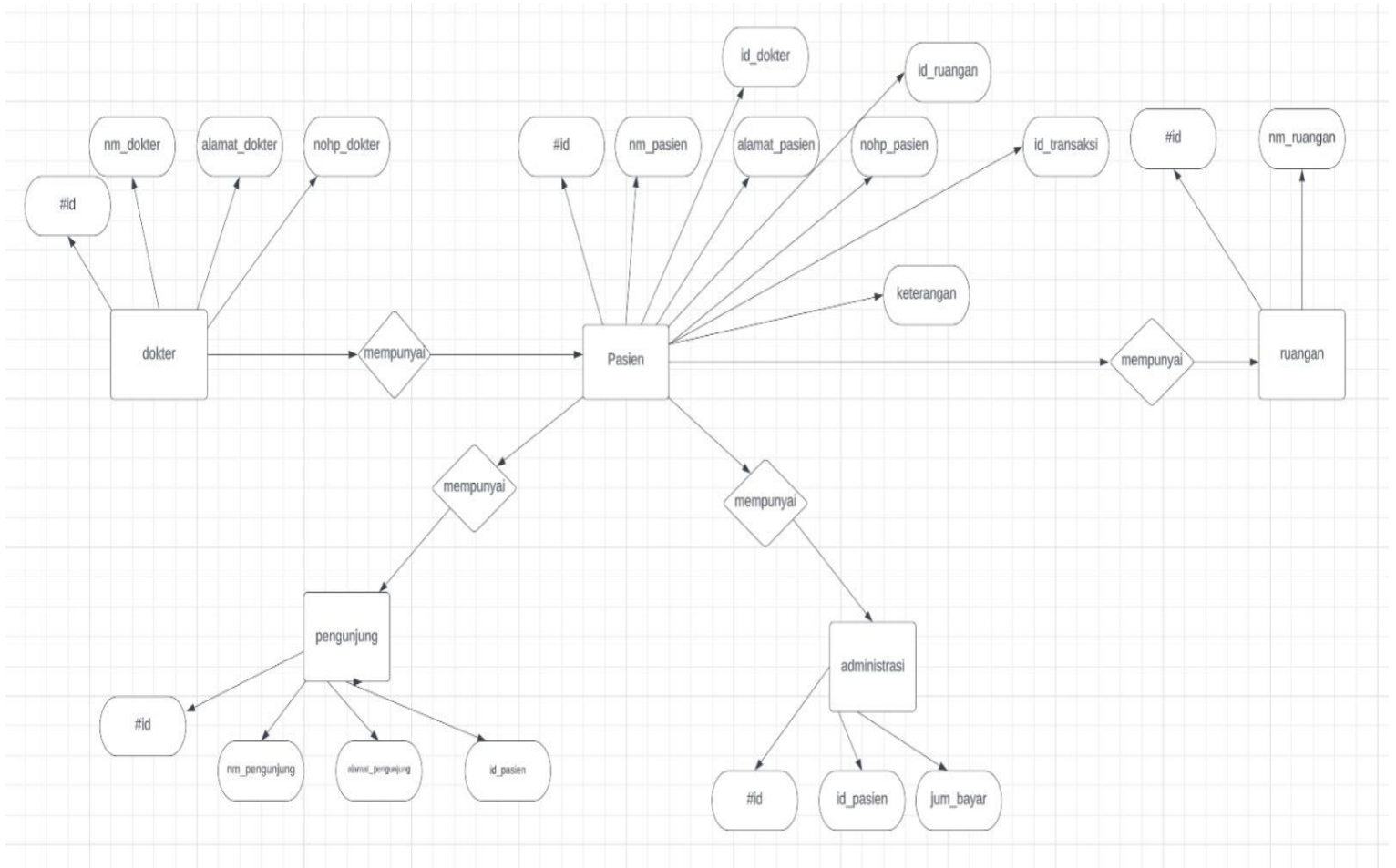
Tabel utama : pasien

Table kedua: administrasi

Relationship: one to one

Atribut penghubung: id\_transaksi(FK Id di administrasi untuk table pasien) dan id\_pasien(FK id di pasien untuk table administrasi)

#### D. Tahap 4 Pembuatan ERD



### 3. Logical Relational Structure (LRS)

Logical Relational Structure (LRS) adalah struktur logis yang digunakan dalam desain basis data relasional. Ini merujuk pada cara data disusun dan dihubungkan dalam bentuk tabel dan hubungan antartabel di dalam basis data relasional.

Dengan menggunakan LRS, desainer basis data dapat mengatur dan menghubungkan data dalam basis data relasional secara logis. Ini memungkinkan pemodelan yang fleksibel dan efisien dari entitas dan hubungan antara mereka. LRS juga memfasilitasi integritas data, karena hubungan antartabel dapat digunakan untuk menerapkan batasan referensial dan menjaga konsistensi data di seluruh basis data.

Berikut Contoh Dari LRS



## BAB II

### PEMBAHASAN

#### A. PEMBUATAN DATABASE MELALUI COMMAND PROMPT

Pembuatan database melalui Command Prompt dapat dilakukan dengan menggunakan perintah SQL atau perintah spesifik yang diberikan oleh sistem manajemen basis data yang digunakan

#### B. LANGKAH-LANGKAH PEMBUATAN DATABASE MENGGUNAKAN CMD

1. Buka mysql

```
c:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2508
Server version: 10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

2. Buat database

```
MariaDB [(none)]> create database uas_pak_gofar;
Query OK, 1 row affected (0.005 sec)

MariaDB [(none)]> _
```

3. Aktifakn db

```
MariaDB [(none)]> use uas_pak_gofar;
Database changed
MariaDB [uas_pak_gofar]>
```

4. Buat table dokter

```
MariaDB [uas_pak_gofar]> create table dokter (
    -> id int not null auto_increment,
    -> nm_dokter varchar (50),
    -> alamat_dokter varchar (50),
    -> nohp_dokter varchar (50),
    -> primary key (id));
Query OK, 0 rows affected (0.364 sec)

MariaDB [uas_pak_gofar]>
```

5. Buat table ruangan



```
MariaDB [uas_pak_gofar]> create table ruangan (  
  -> id int not null auto_increment,  
  -> nm_ruangan varchar (50) unique,  
  -> primary key (id));  
Query OK, 0 rows affected (0.327 sec)  
  
MariaDB [uas_pak_gofar]> _
```

6. Buat table pasien

```
MariaDB [uas_pak_gofar]> create table pasien (  
  -> id int not null auto_increment,  
  -> nm_pasien varchar (50),  
  -> alamat_pasien varchar (50),  
  -> nohp_pasien varchar (50),  
  -> id_dokter int,  
  -> id_ruangan int,  
  -> id_transaksi int unique,  
  -> primary key (id),  
  -> foreign key (id_dokter) references dokter(id));  
Query OK, 0 rows affected (0.728 sec)
```

7. Tambahkan foreign key id\_ruangan pada table pasien

```
MariaDB [uas_pak_gofar]> alter table pasien  
  -> add foreign key (id_ruangan) references ruangan(id);  
Query OK, 0 rows affected (1.029 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
MariaDB [uas_pak_gofar]> _
```

8. Buat table administrasi

```
MariaDB [uas_pak_gofar]> create table administrasi (  
  -> id int not null auto_increment,  
  -> id_pasien int unique,  
  -> jum_bayar int (50),  
  -> primary key (id),  
  -> foreign key (id_pasien) references pasien(id));  
Query OK, 0 rows affected (0.289 sec)  
  
MariaDB [uas_pak_gofar]>
```

9. Menambahkan foreign key id\_transaksi pada tabel pasien

```
MariaDB [uas_pak_gofar]> alter table pasien
-> add foreign key (id_transaksi) references administrasi(id);
Query OK, 0 rows affected (0.948 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [uas_pak_gofar]>
```

10. Menambahkan column keterangan pada table pasien

```
MariaDB [uas_pak_gofar]> alter table pasien
-> add keterangan enum('belum lunas', 'lunas') default 'belum lunas' unique;
Query OK, 0 rows affected (1.001 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [uas_pak_gofar]>
```

```
MariaDB [uas_pak_gofar]> alter table pasien
-> drop index keterangan;_
```

11. Membuat trigger untuk keterangan lunas atau belum

```
MariaDB [uas_pak_gofar]> create trigger keterangan_lunas after insert on administrasi
-> for each row
-> update pasien
-> set keterangan = 'lunas'
-> where id in (select id_pasien from administrasi);
Query OK, 0 rows affected (0.153 sec)
```

12. Membuat trigger untuk id\_transaksi jika sudah bayar maka akan muncul id (table administrasi) nya

```
CREATE DEFINER='root'@'localhost' TRIGGER `update_id_transaksi` AFTER INSERT
ON
`administrasi` FOR EACH ROW update pasien as u join administrasi as a on u.id
= a.id_pasien set u.id_transaksi = a.id
```

13. buat table pengunjung

```

MariaDB [uas_pak_gofar]> create table pengunjung (
  -> id int not null auto_increment,
  -> nm_pengunjung varchar(50),
  -> alamat_pengunjung varchar (50),
  -> id_pasien int,
  -> primary key (id),
  -> foreign key (id_pasien) references pasien(id));
Query OK, 0 rows affected (0.492 sec)

```

14. menambahkan data pda table ruangan

```

MariaDB [uas_pak_gofar]> insert into ruangan (nm_ruangan)
  -> values
  -> ('kamar mawar'),
  -> ('kamar melati'),
  -> ('kamar anggrek');
Query OK, 3 rows affected (0.133 sec)
Records: 3 Duplicates: 0 Warnings: 0

```

15. menambahkan data pada table dokter

```

MariaDB [uas_pak_gofar]> insert into dokter (nm_dokter, alamat_dokter, nohp_dokter)
  -> values
  -> ('tono', 'bekasi','089999999'),
  -> ('wawan keder', 'jakarta timur','08922229'),
  -> ('anies', 'cengkareng','0892220020');
Query OK, 3 rows affected (0.093 sec)
Records: 3 Duplicates: 0 Warnings: 0

```

16. tambah data pada table pasien

```

MariaDB [uas_pak_gofar]> insert into pasien
  -> (nm_pasien, alamat_pasien, nohp_pasien,id_dokter,id_ruangan)
  -> values
  -> ('kurnia', 'bekasi', '089123' ,2, 2);
Query OK, 1 row affected (0.107 sec)

```

17. tambah data pada administrasi

```

MariaDB [uas_pak_gofar]> insert into administrasi
  -> (id_pasien, jum_bayar)
  -> values
  -> (7, 50000000),
  -> (10, 22000000),
  -> (8, 24200000);
Query OK, 3 rows affected (0.266 sec)
Records: 3  Duplicates: 0  Warnings: 0

```

18. tambah data pada pengunjung

```

MariaDB [uas_pak_gofar]> insert into pengunjung
  -> (nm_pengunjung, alamat_pengunjung, id_pasien)
  -> values
  -> ('dimas', 'jakarta barat', 7),
  -> ('dayat', 'jakarta selatan', 6);
Query OK, 2 rows affected (0.091 sec)
Records: 2  Duplicates: 0  Warnings: 0

```

19. Menampilkan data dari database uas\_pak\_gofar

```

MariaDB [uas_pak_gofar]> select * from administrasi
  -> ;
+----+-----+-----+
| id | id_pasien | jum_bayar |
+----+-----+-----+
| 6  |          | 7         | 50000000 |
| 7  |         10 | 22000000  |
| 8  |          | 8         | 24200000 |
+----+-----+-----+
3 rows in set (0.037 sec)

```

```

MariaDB [uas_pak_gofar]> select * from dokter;
+----+-----+-----+-----+
| id | nm_dokter | alamat_dokter | nohp_dokter |
+----+-----+-----+-----+
| 1  | tono      | bekasi        | 08999999    |
| 2  | wawan keder | jakarta timur | 08922229    |
| 3  | anies     | cengkareng    | 0892220020  |
+----+-----+-----+-----+
3 rows in set (0.015 sec)

```

```
MariaDB [uas_pak_gofar]> select * from pasien;
```

id	nm_pasien	alamat_pasien	nohp_pasien	id_dokter	id_ruangan	id_transaksi	keterangan
6	yayan selamat	bogor	089992929	2	2	NULL	belum lunas
7	yati males	bandung	089993232	3	1	6	lunas
8	udin petot	bekasi	08233232	1	1	8	lunas
9	nurdi k.o	bekasi	08222334	1	3	NULL	belum lunas
10	berman	depok	08222345	3	3	7	lunas
11	kurnia	bekasi	089123	2	2	NULL	belum lunas

```
6 rows in set (0.039 sec)
```

```
MariaDB [uas_pak_gofar]> select * from pengunjung;
```

id	nm_pengunjung	alamat_pengunjung	id_pasien
1	dimas	jakarta barat	7
2	dayat	jakarta selatan	6

```
2 rows in set (0.003 sec)
```

```
MariaDB [uas_pak_gofar]> select * from ruangan;
```

id	nm_ruangan
3	kamar anggrek
1	kamar mawar
2	kamar melati

```
3 rows in set (0.014 sec)
```

## 20. Menampilkan Data Berdasarkan INNER JOIN

```
MariaDB [uas_pak_gofar]> select nm_pasien, dokter.nm_dokter from pasien
-> inner join dokter on pasien.id_dokter = dokter.id;
```

nm_pasien	nm_dokter
yayan selamat	wawan keder
yati males	anies
udin petot	tono
nurdi k.o	tono
berman	anies
kurnia	wawan keder

6 rows in set (0.002 sec)

```
MariaDB [uas_pak_gofar]> select nm_pasien, dokter.nm_dokter, ruangan.nm_ruangan
-> , id_transaksi from pasien
-> inner join dokter on pasien.id_dokter = dokter.id
-> inner join ruangan on pasien.id_ruangan = ruangan.id;
```

nm_pasien	nm_dokter	nm_ruangan	id_transaksi
udin petot	tono	kamar mawar	8
nurdi k.o	tono	kamar anggrek	NULL
yayan selamat	wawan keder	kamar melati	NULL
kurnia	wawan keder	kamar melati	NULL
yati males	anies	kamar mawar	6
berman	anies	kamar anggrek	7

6 rows in set (0.012 sec)

```
MariaDB [uas_pak_gofar]> select nm_pengunjung, pasien.nm_pasien from pengunjung
-> inner join pasien on pengunjung.id_pasien = pasien.id;
```

nm_pengunjung	nm_pasien
dimas	yati males
dayat	yayan selamat

2 rows in set (0.001 sec)

## **BAB III**

### **PENUTUP**

#### **A. KESIMPULAN**

1. Database (Sistem Basis Data) merupakan suatu komponen data yang telah dibuat dan saling berhubungan satu dengan yang lain nya dan tersimpan dalam suatu komputer sehingga di gunakan perangkat lunak untuk memanipulasinya.
2. ERD merupakan pemodelan basis data dengan menggunakan diagram relasi nya antar entitas, untuk memodelkan suatu data dan hubungan antar data untuk menggambar nya digunakan beberapa simbol yang tepat.
3. LRS untuk menggambarkan struktur record record yang sudah kita buat berdasarkan ERD atau pada tabel tabel yang terbentuk dari hasil relasi antar himpunan entitas.
4. Command prompt dapat digunakan sebagai salah satu sarana untuk membuat database yang kita inginkan.

#### **B. SARAN**

1. Diharapkan dengan dibuatnya makalah ini dengan baik, mahasiswa bisa mempelajari jadi lebih mengerti step atau langkah langkah dalam pembuatan database tersebut.
2. . Kepada seluruh pembaca kiranya akan memberikan kritik dan saran yang bersifat membangun sehingga apa yang kita harapkan dari isi tulisan ini dapat berguna bagi pembaca maupun penulis.