

**LAPORAN FINAL PROJECT**  
**MATA KULIAH MANAJEMEN BASIS DATA**  
**“Sistem Basis Data Klinik”**



Disusun oleh:

Kelompok B01

1. Maisan Auliya 5025201137
2. Afril Muzzaqi Arif 5025201048
3. Shaloom David Togu 5025211242
4. Abdurrahman Farimza 5025201125

Kelas B

Dosen Pengampu: Sarwosri, S.Kom. M.T

**DEPARTEMEN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS**  
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**TAHUN 2022/2023**

## DAFTAR ISI

Daftar isi .....	2
Studi Kasus .....	3
Skema Database .....	5
Sequence .....	6
Function .....	11
Procedure .....	15
Trigger .....	19
Index .....	25

## STUDI KASUS

Studi kasus kami, "Perancangan Sistem Basis Data Klinik", merupakan suatu contoh perancangan sistem basis data yang bertujuan untuk mengorganisir dan mengelola data yang terkait dengan operasional sebuah klinik. Sistem basis data ini dirancang untuk mendukung berbagai fitur dan fungsionalitas yang terkait dengan pengelolaan klinik, termasuk transaksi pembelian obat, informasi pasien, rekam medis, poliklinik, dan dokter.. Berikut adalah penjelasan mengenai setiap entitas, serta fitur-fitur yang terkait:

1. Entitas Transaksi:
  - a. Deskripsi: Mencatat transaksi pembelian obat oleh pasien.
2. Entitas Obat:
  - a. Deskripsi: Menyimpan informasi mengenai obat yang tersedia di klinik.
3. Entitas Jenis Obat:
  - a. Deskripsi: Menyimpan kategori atau jenis obat yang digunakan untuk mengelompokkan obat-obatan.
4. Entitas Pasien:
  - a. Deskripsi: Mewakili informasi mengenai pasien yang berkunjung ke klinik.
5. Entitas Rekam Medis:
  - a. Deskripsi: Menyimpan catatan medis pasien yang meliputi riwayat penyakit, diagnosa, dan pengobatan.
6. Entitas Poli:
  - a. Deskripsi: Merepresentasikan poliklinik di klinik yang menyediakan layanan medis tertentu.
7. Entitas Dokter:
  - a. Deskripsi: Mencatat informasi mengenai dokter yang bertugas di klinik.

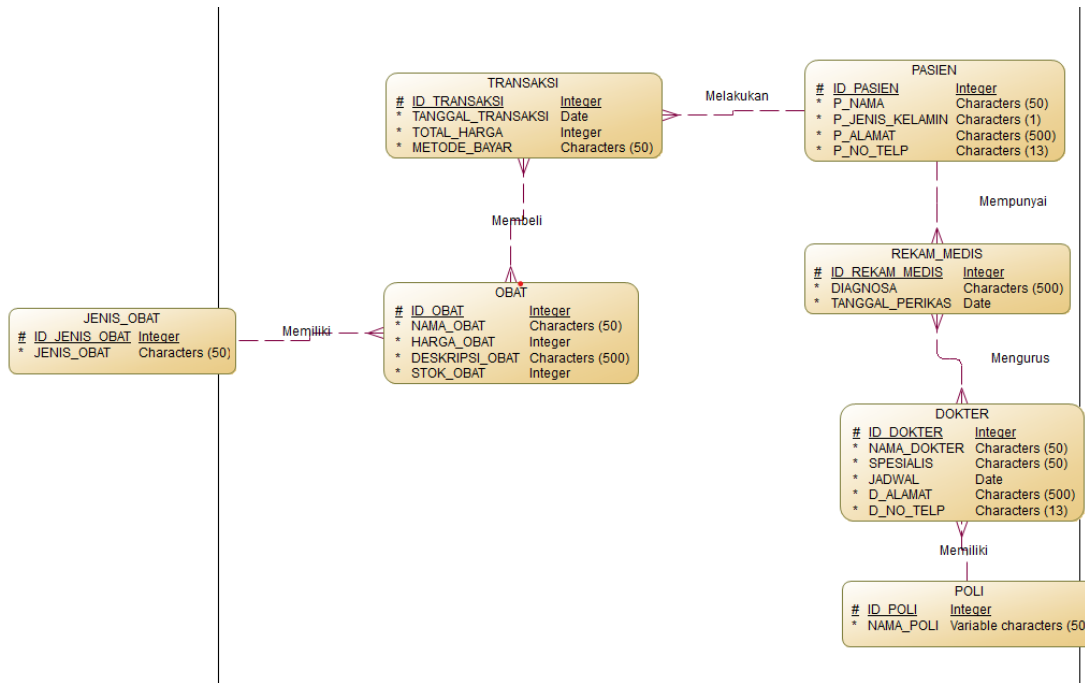
Dengan adanya sistem basis data yang dirancang dengan baik, data-data tersebut dapat disimpan, dikelola, dan dimanfaatkan secara efisien dalam operasional klinik, memudahkan pemantauan stok obat, pencatatan historis pasien, serta penjadwalan praktek dokter.

Fitur-fitur dari studi kasus "Perancangan Sistem Basis Data Klinik" ini dapat mencakup:

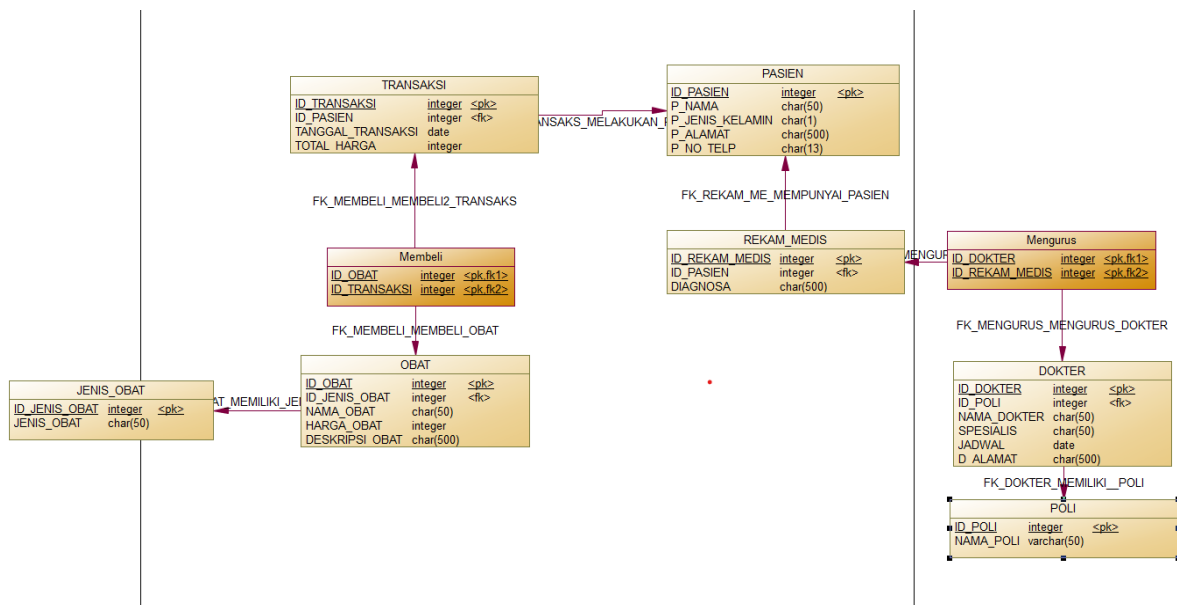
- Pencatatan transaksi pembelian obat oleh pasien.
- Manajemen obat, termasuk informasi stok, harga, dan jenis obat.
- Informasi pasien, termasuk riwayat penyakit, diagnosa, dan pengobatan dalam rekam medis.
- Hubungan antara pasien dengan rekam medis yang berkaitan.
- Informasi mengenai poliklinik yang tersedia di klinik.
- Informasi mengenai dokter yang bertugas di klinik, termasuk spesialisasi dan jadwal praktek.

## SKEMA DATABASE

### CDM



### PDM



## **ACTIVE DATABASE SEQUENCE**

**Sequence untuk membuat insert berikutnya pada tabel pasien menjadi increment by 1 (akan langsung menambah id tabel + 1)**

Code:

```
CREATE SEQUENCE pasien_sequence  
START WITH 1002  
INCREMENT BY 1  
NO MAXVALUE  
NO MINVALUE  
CACHE 1;  
  
ALTER TABLE PASIEN  
ALTER COLUMN ID_PASIEN SET DEFAULT nextval('pasien_sequence');
```

Penjelasan :

**Pada sequeunce ini, kami meulai dari 1002 (Start with 1002) karena kami sudah melakukan 1001 insert data sebelum sequence ini dijalankan dan terlihat bahwa ID\_PASIEN akan melakukan increment + 1 pada setiap insert baru.**

Hasil:

Kami melakukan insert pertama menggunakan:

```
INSERT INTO pasien (ID_PASIEN, NAMA_PASIEN, JENIS_KELAMIN,
ALAMAT, NO_TELP)
VALUES (nextval('pasien_sequence'), 'Maisan', 'M', 'Jl. Contoh No. 123',
'081234567890');
```

Dan insert berikutnya:

```
INSERT INTO pasien (ID_PASIEN, NAMA_PASIEN, JENIS_KELAMIN,
ALAMAT, NO_TELP)
VALUES (nextval('pasien_sequence'), 'Fani', 'M', 'Jl. Contoh No. 123',
'081234567890');
```

Dan hasilnya setelah saya melakukan order by desc:

	 id_pasien [PK] integer 	nama_pasien character varying (50) 	jenis_kelamin character (1) 	alamat character varying (100) 	no_telp character varying (13) 
1	1003	Fani	<div>Editable column</div> M	Jl. Contoh No. 123	081234567890
2	1002	Maisan	M	Jl. Contoh No. 123	081234567890
3	1001	John Doe	M	Jl. Contoh No. 123	081234567890
4	1000	Miriam	F	22372 Little Fleur Street	713-349-5457
5	999	Bryna	F	84831 2nd Terrace	676-740-4642
6	998	Jeremias	M	4 Eastlawn Street	993-584-4762
7	997	Dorena	F	63 Meadow Ridge Place	479-367-4204
8	996	Walsh	M	4224 Lakeland Drive	100-374-0201
9	995	Giralda	F	5772 Lighthouse Bay Terrace	523-777-7960

Terlihat bahwa insert terbaru akan langsung mendapatkan id+1 dari id sebelumnya

## VIEW

1. View untuk melihat poli yang paling banyak menangani pasien

Author : Maisan

Query :

```
SELECT id_poli, nama_poli, count(rekam_medis.id_pasien) as  
jumlah_pasien FROM public.poli  
join dokter using (id_poli)  
join rekam_medis_dokter using(id_dokter)  
join rekam_medis using(id_rekammed)  
group by id_poli  
order by jumlah_pasien desc limit 1
```

Penjelasan :

melakukan join dengan 3 tabel yaitu tabel rekam medis, rekam medis\_dokter dan dokter untuk mendapatkan poli yang paling banyak dikunjungi berdasarkan rekam medis pasien

Hasil :

Explain	Data Output	Messages	Notifications
	<b>id_poli</b> [PK] integer	<b>nama_poli</b> character varying (100)	<b>jumlah_pasien</b> bigint
1	6	Umum	3

2. View untuk melihat jenis obat dengan stok tersedikit

Author : Farim

Query : CREATE VIEW view\_stok\_terkecil AS



```

SELECT jenis_obat>Nama_Jenis_Obat, MIN(obat.Stok) AS Stok_Tersedikit
FROM jenis_obat
JOIN obat ON jenis_obat.ID_Jenis_Obat = obat.ID_Jenis_Obat
GROUP BY jenis_obat>Nama_Jenis_Obat
ORDER BY MIN(obat.Stok) ASC;

```

Penjelasan : view ini menampilkan urutan jenis obat dengan stok tersedikit

Hasil :

Nama_Jenis_Obat	Stok_Tersedikit
Analgesik	10
Antibiotik	15
Antasida	20
Antihistamin	25
Antiinflamasi	30

### 3. View untuk menampilkan nama nama pasien yang di periksa dokter X

Author : Afril

Query :

```

CREATE VIEW View_DaftarPasienDokterX AS
SELECT P.nama_pasien
FROM Pasien AS P
JOIN RekamMedis AS RM ON P.id_pasien = RM.id_pasien
JOIN Dokter AS D ON RM.id_dokter = D.id_dokter
WHERE D.nama_dokter = 'X';

```

```

SELECT * FROM View_DaftarPasienDokterX;

```

Penjelasan :View ini akan menghasilkan daftar nama-nama pasien yang telah diperiksa oleh dokter dengan nama 'X'

Hasil :

nama_pasien
John Doe
Jane Smith
Michael Johnson

4. View untuk ...

Author : Shaloom

Query :

Penjelasan :

Hasil :

dst sesuai jumlah anggota pada kelompok.

## **FUNCTION**

1. Function untuk menampilkan history transaksi dari pasien dengan parameter ID pasien

Author : Maisan

function :

### **CREATE OR REPLACE FUNCTION**

**get\_patient\_transaction\_history(patient\_id INT)**

**RETURNS TABLE (**

**ID\_TRANSAKSI INT,**

**TANGGAL\_TRANSAKSI DATE,**

**TOTAL\_HARGA INT,**

**METODE\_BAYAR VARCHAR(50)**

**)**

**AS \$\$**

**BEGIN**

**RETURN QUERY**

**SELECT**

**t.ID\_TRANSAKSI AS ID\_TRANSAKSI,**

**t.TANGGAL\_TRANSAKSI AS tTANGGAL\_TRANSAKSI,**

**t.TOTAL\_HARGA AS TOTAL\_HARGA,**

**t.METODE\_BAYAR AS payment\_method**

**FROM**

**TRANSAKSI t**

**WHERE**

**t.ID\_PASIEN = patient\_id;**

**END;**

**\$\$ LANGUAGE plpgsql;**

Penjelasan :

Function ini akan me-return id\_transaksi, tanggal, total harga dan metode bayar berdasarkan ID pasien, atau bisa dibilang akan mengembalikan semua history transaksi dari pasien

Hasil :

disini saya melakukan wuery menampilkan semua transaksi, untuk mengecek kebenarannya

Explain

Data Output

Messages

Notifications

	<div><div><div><div></div><div></div></div><div><div>id_transaksi</div><div>[PK] integer</div></div></div><div><div></div><div></div></div></div>	<div><div><div><div></div><div></div></div><div><div>id_pasien</div><div>integer</div></div></div><div><div></div><div></div></div></div>	<div><div><div><div></div><div></div></div><div><div>tanggal_transaksi</div><div>date</div></div></div><div><div></div><div></div></div></div>	<div><div><div><div></div><div></div></div><div><div>total_harga</div><div>integer</div></div></div><div><div></div><div></div></div></div>	<div><div><div><div></div><div></div></div><div><div>metode_bayar</div><div>character varying (50)</div></div></div><div><div></div><div></div></div></div>
1	1	1	2021-04-11	285000	Kredit
2	2	2	2021-05-13	261000	Tunai
3	3	3	2021-05-17	96255	Tunai
4	4	4	2021-05-25	104225	Kredit
5	5	5	2021-06-11	285000	Tunai
6	6	6	2021-06-11	117365	Kredit
7	7	7	2021-06-11	96255	Tunai

Terlihat bahwa untuk pasien dengan ID 3, hanya memiliki 1 transaksi dan kemudian saya menjalankan function tadi, dan ini hasilnya.

Explain

Data Output

Messages

Notifications

	id_transaksi integer	tanggal_transaksi date	total_harga integer	metode_bayar character varying
1	3	2021-05-17	96255	Tunai

## 2. Function untuk menghitung total pembayaran

Author : Farim

function : CREATE FUNCTION hitungTotalPembayaran(ID\_Transaksi INT) RETURNS DECIMAL(10,2)

```

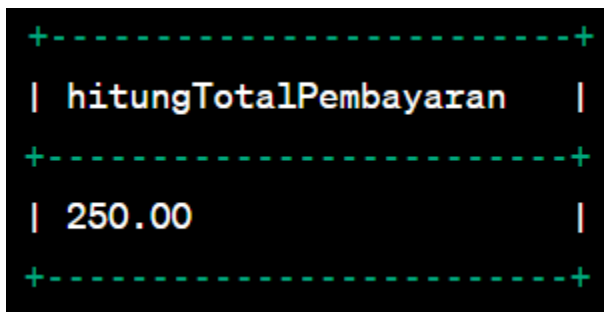
BEGIN
    DECLARE total DECIMAL(10,2);
    SELECT SUM(harga) INTO total
    FROM obat
    WHERE pembayaran_obat = ID_Transaksi;

    RETURN total;
END;

```

Penjelasan : Function ini merupakan sebuah fungsi yang menerima parameter Id\_Transaksi dan menghitung total pembayaran obat berdasarkan ID transaksi yang diberikan.

Hasil :



```

+-----+
| hitungTotalPembayaran |
+-----+
| 250.00                |
+-----+

```

### 3. Function untuk menampilkan daftar obat yang tersedia

Author : Afril

function :

```

CREATE FUNCTION GetDaftarObatTersedia()
RETURNS TABLE (nama_obat VARCHAR(255), nama_jenis_obat
    VARCHAR(255))
AS $$
BEGIN
    RETURN QUERY
    SELECT o.nama_obat, j.nama_jenis_obat
    FROM obat o
    JOIN jenis_obat j ON o.id_jenis_obat = j.id_jenis_obat;
END;

```

***\$\$ LANGUAGE plpgsql;***

Penjelasan : Fungsi ini akan mengambil data dari tabel obat dan jenis\_obat dengan melakukan operasi join berdasarkan kolom id\_jenis\_obat. Kemudian, akan ditampilkan nama obat dari tabel obat dan nama jenis obat dari tabel jenis\_obat yang sesuai dengan relasinya.

Hasil :

nama_obat	nama_jenis_obat
-----+-----	
Paracetamol	Analgesik
Amoxicillin	Antibiotik
Loratadine	Antihistamin
Omeprazole	Antasida
Simvastatin	Lipid Lowering

4. Function untuk ...

Author :Shaloom

function :

Penjelasan :

Hasil :

## **PROCEDURE**

1. Procedure untuk menghapus data pasien terkait menggunakan parameter ID pasien

Author : Maisan

procedure :

### **CREATE OR REPLACE PROCEDURE**

**delete\_patient\_and\_related\_data(patient\_id INT)**

**AS \$\$**

**BEGIN**

**DELETE FROM TRANSAKSI\_OBAT WHERE ID\_TRANSAKSI IN  
(SELECT ID\_TRANSAKSI FROM TRANSAKSI WHERE ID\_PASIEN =  
patient\_id);**

**DELETE FROM TRANSAKSI WHERE ID\_PASIEN = patient\_id;**

**DELETE FROM REKAM\_MEDIS\_DOKTER WHERE ID\_REKAMMED  
IN (SELECT ID\_REKAMMED FROM REKAM\_MEDIS WHERE  
ID\_PASIEN = patient\_id);**

**DELETE FROM REKAM\_MEDIS WHERE ID\_PASIEN = patient\_id;**

**DELETE FROM PASIEN WHERE ID\_PASIEN = patient\_id;**

**END;**

**\$\$ LANGUAGE plpgsql;**

Penjelasan :

**Procedure ini akan menghapus semua data pasien, bukan hanya di table pasien saja, tetapi semua data pada tabel manapun yang terkait dengan pasien yang dipilih.**

Hasil :

saya memanggil **CALL delete\_patient\_and\_related\_data(3)**; dimana saya akan menghapus semua data terkait pasien dengan ID 3. kemudian saya cek pada 3 tabel ini (tidak semua tabel / tabel ini dijadikan sampel), yaitu tabel pasien ,transaksi dan rekam medis.

	id_transaksi [PK] integer	id_pasien integer	tanggal_transaksi date	total_harga integer	metode_bayar character varying (50)
1	1	1	2021-05-13	285000	Kredit
2	2	2	2021-05-13	261000	Tunai
3	4	4	2021-05-25	104225	Kredit
4	5	5	2021-06-11	285000	Tunai
5	6	6	2021-06-11	117365	Kredit
6	7	7	2021-06-11	96255	Tunai
7	8	8	2021-06-11	203000	Kredit

	id_pasien [PK] integer	nama_pasien character varying (50)	jenis_kelamin character (1)	alamat character varying (100)	no_telp character varying (13)
1	1	Barhan Akmal Falaudin	M	Jl Proklamasi 56, Dki Jakarta	082244607846
2	2	Monica Narda Davita	F	Jl Enggano Bl -15/15, Dki Jakarta	0895342292555
3	4	Nur Muhammad Ainul Yaqin	F	Jl Prof Dr Supomo 141, Dki Jakarta	085101339177
4	5	Farros Hilmi Syafei	M	Jl H Daud 31 RT 010/10, Dki Jakarta	082234481973
5	6	ADIFA WIDYADHANI CHANDA D	F	Margomulyo Permai H-8, Propinsi Jawa Timur	085816271128
6	7	Nadya Permata Sari	F	Jl Mangga Dua Raya Harco Mangga Dua Bl A-2/99, Dki Jakarta	081259100510
7	8	Arya Widia Putra	M	Jl H Nawi Raya 9-A, Dki Jakarta	081236033096
8	9	Muhammad Rolanov Wowor	M	Jl Iskandarsyah Raya 66 C Graha Iskandarsyah, Dki Jakarta	081334197320

	id_rekammed [PK] integer	id_pasien integer	diagnosa character varying (50)	tanggal_periksa date
1	1	1	Asam Lambung	2021-04-11
2	2	2	Darah Rendah	2021-05-11
3	4	4	Sakit Kepala	2021-05-11
4	5	5	Maag	2021-06-11
5	6	6	Flu dan Batuk	2021-06-11
6	7	7	Demam	2021-06-11
7	8	8	Migrain	2021-06-11
8	9	9	Asam Lambung	2021-08-11



terlihat bahwa id no.3 sudah tidak ada, karena telah dihilangkan menggunakan procedure dengan parameter ID 3

2. Procedure untuk menambah data rekam medis

Author : Farim

procedure : CREATE PROCEDURE tambahRekamMedis(ID\_Pasien INT, tanggal DATE, riwayat\_penyakit TEXT, diagnosa TEXT, pengobatan TEXT)  
BEGIN

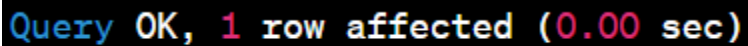
INSERT INTO rekam\_medis (ID\_Pasien, Tanggal\_Rekam\_Medis, Riwayat\_Penyakit, Diagnosa, Pengobatan)

VALUES (ID\_Pasien, tanggal, riwayat\_penyakit, diagnosa, pengobatan);

END;

Penjelasan : Query procedure di atas merupakan sebuah prosedur yang digunakan untuk menambahkan data rekam medis baru ke dalam tabel rekam medis

Hasil :



Query OK, 1 row affected (0.00 sec)

3. Procedure untuk memperbarui jumlah stok obat dengan memberikan ID obat dan jumlah stok baru

Author : Afril

procedure :

**CREATE PROCEDURE UpdateStock(**

**@ObatID INT,**

**@JumlahStokBaru INT**

**)**

**AS**

**BEGIN**

**UPDATE Obat**

**SET JumlahStok = @JumlahStokBaru**

**WHERE ObatID = @ObatID;**

**END;**

Penjelasan :Prosedur ini memungkinkan pengguna untuk dengan mudah memperbarui jumlah stok obat di klinik dengan memberikan ID obat dan jumlah stok baru yang diinginkan.

Hasil :  
Sebelum pembaruan:

Tabel "Obat":

ObatID	NamaObat	JumlahStok
1	Paracetamol	50
2	Amoxicillin	30
3	Ibuprofen	20

Setelah menjalankan prosedur dengan parameter sebagai berikut:

@ObatID = 2

@JumlahStokBaru = 40

Setelah pembaruan:

Tabel "Obat":

ObatID	NamaObat	JumlahStok
1	Paracetamol	50
2	Amoxicillin	40
3	Ibuprofen	20

#### 4. Procedure untuk ...

Author :Shaloom

procedure :

Penjelasan :

Hasil :

## TRIGGER

1. Trigger untuk melakukan insert tanggal otomatis pada tabel rekam medis (after insert)

Author : Maisan

trigger :

```
CREATE OR REPLACE FUNCTION auto_rekamed_date()
```

```
RETURNS TRIGGER
```

```
LANGUAGE PLPGSQL
```

```
AS
```

```
$$
```

```
BEGIN
```

```
NEW.TANGGAL_PERIKSA = CURRENT_TIMESTAMP;
```

```
RETURN NEW;
```

```
END;
```

```
$$;
```

```
CREATE TRIGGER fill_rekamed_date
```

```
BEFORE INSERT
```

```
ON REKAM_MEDIS
```

```
FOR EACH ROW
```

```
EXECUTE PROCEDURE auto_rekamed_date();
```

Penjelasan :

Trigger ini membuat user tidak perlu memasukkan input data tanggal secara manual lagi, sistem akan langsung membuat tanggal nya berdasarkan waktu saat ini.

Hasil :

Ketika saya melakukan insert pada tabel rekam medis menggunakan syntax ini

**INSERT INTO REKAM\_MEDIS VALUES (34,34,'Asam Lambung');**

Terlihat bahwa saya tidak memasukkan tanggal, karena trigger sudah dipasang, maka hasilnya menjadi otomatis seperti ini

id_rekammed [PK] integer	id_pasien integer	diagnosa character varying (50)	tanggal_periksa date
26	27	Demam	2021-12-11
27	28	Migrain	2021-12-15
28	29	Asam Lambung	2021-12-15
29	30	Maag	2021-12-15
30	31	Darah Tinggi	2021-12-20
31	32	Darah Rendah	2021-12-20
32	33	Luka Bakar	2021-12-25
33	34	Asam Lambung	2023-06-17

**Terlihat pada baris paling bawah, tanggal akan otomatis di set menggunakan tanggal sekarang.**

## 2. Trigger untuk mengupdate stok obat setelah transaksi

Author : Farim

trigger : CREATE TRIGGER updateStokObat AFTER INSERT ON transaksi

FOR EACH ROW

BEGIN

DECLARE ID\_obat INT;

DECLARE jumlah\_beli INT;

SELECT pembayaran\_obat, COUNT(\*) INTO ID\_Obat, jumlah\_beli  
FROM obat

WHERE pembayaran\_obat = NEW.ID\_Transaksi;

```

UPDATE obat
SET stok = stok - jumlah_beli
WHERE ID_Obat = ID_Obat;
END;

```

Penjelasan : trigger ini dirancang untuk mengupdate stok obat setelah terjadi transaksi pembelian obat baru, sehingga stok obat di dalam tabel obat akan mengikuti perubahan setiap kali ada transaksi baru yang terjadi.

Hasil :

```

SELECT * FROM obat;

+-----+-----+-----+
| ID_Obat | pembayaran_obat | Stok |
+-----+-----+-----+
| 1       | 1               | 50   |
| 2       | 1               | 30   |
+-----+-----+-----+

-- Menjalankan transaksi baru
INSERT INTO transaksi (ID_Transaksi) VALUES (1);

-- Contoh data setelah trigger dijalankan
SELECT * FROM obat;

+-----+-----+-----+
| ID_Obat | pembayaran_obat | Stok |
+-----+-----+-----+
| 1       | 1               | 49   |
| 2       | 1               | 29   |
+-----+-----+-----+

```

3. Trigger untuk membuat catatan medis baru untuk pasien

Author : Afril

trigger :

```
CREATE TRIGGER tambah_catatan_medis
AFTER INSERT ON entitas_transaksi
FOR EACH ROW
BEGIN

    DECLARE id_pasien INT;
    DECLARE id_rekam_medis INT;
    DECLARE tgl_sekarang DATE;

    -- Mendapatkan informasi pasien dari transaksi
    SELECT pasien_id INTO id_pasien FROM Transaksi WHERE id = NEW.id;

    -- Mendapatkan tanggal saat ini
    SET tgl_sekarang = CURDATE();

    -- Mencari rekam medis pasien yang sudah ada atau membuat yang baru jika
    belum ada
    SELECT id INTO id_rekam_medis FROM rekam_medis WHERE pasien_id
    = id_pasien;

    IF id_rekam_medis IS NULL THEN
        -- Jika belum ada rekam medis, buat yang baru
        INSERT INTO rekam_medis (pasien_id, tgl_pembuatan) VALUES
        (id_pasien, tgl_sekarang);
    ELSE
        -- Jika sudah ada rekam medis, update tanggal pembuatan terakhir
        UPDATE rekam_medis SET tgl_pembuatan = tgl_sekarang WHERE id =
        id_rekam_medis;
    END IF;
```

Penjelasan :Jika pasien tersebut belum memiliki rekam medis sebelumnya, maka akan dibuat catatan medis baru dengan tanggal pembuatan saat ini. Jika pasien tersebut sudah memiliki rekam medis sebelumnya, maka tanggal pembuatan rekam medis akan diperbarui menjadi tanggal saat ini.

Hasil :

Misalkan kita memiliki tabel entitas\_transaksi

```
entitas_transaksi
-----
id (int)
pasien_id (int)
obat_id (int)
tanggal_pembelian (date)
```

dan tabel entitas\_rekam\_medis

```
entitas_rekam_medis
-----
id (int)
pasien_id (int)
tgl_pembuatan (date)
```

Ketika sebuah transaksi pembelian obat oleh pasien dilakukan, trigger tambah\_catatan\_medis akan diaktifkan dan membuat catatan medis baru

Sebelum trigger dijalankan tabel entitas\_transaksi:

id	pasien_id	obat_id	tanggal_pembelian
1	101	201	2023-06-15
2	102	202	2023-06-16

Tabel entitas\_rekam\_medis:

id	pasien_id	tgl_pembuatan
1	101	2023-06-14
2	102	2023-06-13

Setelah trigger dijalankan:

Tabel entitas\_transaksi tidak berubah.

Tabel entitas\_rekam\_medis:

id	pasien_id	tgl_pembuatan	
1	101	2023-06-15	(diperbarui)
2	102	2023-06-16	(diperbarui)
3	103	2023-06-17	(baru)

4. Trigger untuk ...

Author :Shaloom

trigger :

Penjelasan :

Hasil :



## INDEX

Table Name : Pasien

Query :

**CREATE INDEX IF NOT EXISTS idx\_name\_pasien**

**ON public.pasien USING btree**

**(nama\_pasien COLLATE pg\_catalog."default" text\_pattern\_ops ASC NULLS LAST)**


**TABLESPACE pg\_default;**

Penjelasan :

Melakukan Testing Menggunakan

**explain analyze select \* from pasien where nama\_pasien like 'Maisan';**

Sebelum Indexing :

Explain	Data Output	Messages	Notifications
	<b>QUERY PLAN</b> text		
1	Seq Scan on pasien (cost=0.00..22.50 rows=1 width=45) (actual time=0.191..0.191 rows=1 loops=1)		
2	[...] Filter: ((nama_pasien)::text ~~ 'Maisan'::text)		
3	[...] Rows Removed by Filter: 1002		
4	Planning Time: 0.096 ms		
5	Execution Time: 0.211 ms		

Setelah Indexing:

Explain	Data Output	Messages	Notifications
	<div>QUERY PLAN</div> <div>text</div>		
1	Index Scan using idx_name_pasien on pasien (cost=0.28..8.29 rows=1 width=45) (actual time=0.026...		
2	[...] Index Cond: ((nama_pasien)::text = 'Maisan'::text)		
3	[...] Filter: ((nama_pasien)::text ~~ 'Maisan'::text)		
4	Planning Time: 1.581 ms		
5	Execution Time: 0.041 ms		

Terlihat bahwa pada process 1 telah ada kalimat “Index Scan” (Poin 1) dan pada Execution Time nya sendiri (Poin 5) , dimana jika tanpa menggunakan indexing, Execution time nya adalah 0.211 ms sedangkan setelah melakukan indexing, Execution time menurun menjadi 0.041 ms. Ini berarti bahwa indexing berhasil dan waktu eksekusi query untuk mencari nama “maisn” menjadi lebih cepat ( menggunakan 1000+ data pada tabel pasien ).