

Nama : Angga Firmansyah

NRP : 5027241062

Laporan Analisis Studi Kasus Database Rumah Sakit HospitalIT

1.1 Daftar Tabel

Tabel Log Transaksi Harian

No	Tanggal	Info_Pasien_Lengkap	Dokter_Penanggung_Jawab	Keluhan_Awal	Daftar_Item_Tagihan	Total_Bayar_Final
1	2026-02-01	Budi Santoso (L) – Jl. Mawar No. 10, Surabaya	Dr. Hartono (Umm)	Demam, Pusing	Paracetamol 500mg (10), Vitamin C (5), Jasa Konsultasi Umum	85
2	2026-02-01	Siti Aminah (P) – Jl. Melati 5, Sidoarjo	Drg. Ratna (Gigi)	Gigi Geraham Kanan Nyeri	Asam Mefenamat (10), Amoxicillin (15), Cabut Gigi Bungsu, Jasa Gigi	750
3	2026-02-01	Andi Wijaya (L) – Jl. Anggrek 3, Gresik	Dr. Susan (Jantung)	Nyeri Dada Kiri	Clopidogrel (30), Atorvastatin (30), EKG Jantung, Jasa Spesialis	1.200.000

4	2026-02-02	Budi Santoso (L) – Jl. Mawar No. 10, Surabaya	Dr. Hartono (Umm)	Batuk Berdahak	OBH Sirup (1), CTM (10), Jasa Konsultasi Umum	65
5	2026-02-02	Dewi Sartika (P) – Jl. Kamboja 8, Surabaya	Dr. Bambang (Anak)	Demam Tinggi Anak	Sanmol Drop (1), Puyer Batuk Racikan (10), Jasa Spesialis Anak	200
6	2026-02-03	Eko Prasetyo (L) – Jl. Kenanga 12, Sidoarjo	Dr. Susan (Jantung)	Kontrol Hipertensi	Amlodipine 10mg (30), Bisoprolol (30), Jasa Spesialis	450
7	2026-02-03	Siti Aminah (P) – Jl. Melati 5, Sidoarjo	Drg. Ratna (Gigi)	Kontrol Bekas Cabut	Betadine Kumur (1), Kapas Steril (1), Kontrol Gigi	100
8	2026-02-04	Fajar Nugraha (L) – Jl. Durian 1, Surabaya	Dr. Hartono (Umm)	Luka Lecet Jatuh Motor	Betadine Salep (1), Perban Gulung (2), Asam Mefenamat (10), Jasa Rawat Luka	150
9	2026-02-04	Gita Gutawa (P) – Jl. Mangga 20, Sidoarjo	Drg. Ratna (Gigi)	Karang Gigi	Listerine Cool Mint (1), Scaling Gigi Atas Bawah	350

10	2026-02-05	Andi Wijaya (L) - Jl. Anggrek 3, Gresik	Dr. Susan (Jantung)	Sesak Napas Ringan	Furosemide (10), Spironolactone (10), EKG Jantung, Jasa Spesialis	950
11	2026-02-05	Haryanto (L) - Jl. Nangka 9, Surabaya	Dr. Bambang (Anak)	Imunisasi Polio	- , Jasa Suntik Vaksin	150
12	2026-02-06	Indah Permata (P) - Jl. Jeruk 7, Gresik	Dr. Hartono (Umm)	Masuk Angin	Tolak Angin Cair (5), Vitamin C (10), Jasa Konsultasi Umum	55
13	2026-02-06	Joko Susilo (L) - Jl. Apel 4, Surabaya	Drg. Ratna (Gigi)	Pasang Behel	Paket Ortho Lengkap (1), Sikat Gigi Ortho (1), Jasa Pasang Behel	5.500.000
14	2026-02-07	Budi Santoso (L) - Jl. Mawar No. 10, Surabaya	Dr. Susan (Jantung)	Cek Jantung Rutin	- , Treadmill Test, Jasa Spesialis	750
15	2026-02-07	Kartika Sari (P) - Jl. Salak 11, Sidoarjo	Dr. Bambang (Anak)	Diare Anak	Oralit Sachet (5), Zinc (10), Lacto-B (5), Jasa Spesialis Anak	250

16	2026-02-08	Laila (P) - Jl. Anggur 2, Surabaya	Dr. Hartono (Umm)	Alergi Gatal	CTM (15), Salep Gatal (1), Jasa Konsultasi Umum	75
17	2026-02-08	Muhammad (L) - Jl. Melon 5, Gresik	Drg. Ratna (Gigi)	Gigi Depan Patah	Tambal Komposit (1), Jasa Gigi	400
18	2026-02-09	Nina (P) - Jl. Semangka 8, Sidoarjo	Dr. Bambang (Anak)	Batuk Pilek	OBH Sirup Anak (1), Vitamin C (5), Jasa Spesialis Anak	180
19	2026-02-09	Oscar (L) - Jl. Nanas 1, Surabaya	Dr. Susan (Jantung)	Jantung Berdebar	Bisoprolol (15), EKG Jantung, Jasa Spesialis	500
20	2026-02-10	Putri (P) - Jl. Jambu 3, Surabaya	Dr. Hartono (Umm)	Maag Kambuh	Antasida Doen (10), Omeprazole (10), Jasa Konsultasi Umum	95

Tabel Data mentah Stock Gudang

Kode_Barang	Nama_Item_Obat	Kategori_Varian	Stok_Saat_Ini	Harga_Beli_Satuan	Harga_Jual_Satuan	Expired_Date
OBT-001	Paracetamol 500mg	Tablet, Analgesik (Pereda Nyeri), Antipiretik	150	200	500	2027-12-31
OBT-002	Paracetamol Syrup	Sirup, Antipiretik (Demam), Anak-Anak	20	10000	15000	2026-05-20
MED-003	Vitamin C 500mg	Tablet, Suplemen Vitamin, Daya Tahan Tubuh	500	500	1000	2028-01-01
1004	Asam Mefenamat	Kaplet, NSAID (Anti Inflamasi), Nyeri Gigi	45	400	800	2026-08-15
OBT-005	Amoxicillin 500mg	Kapsul, Antibiotik, Obat Keras	30	800	1200	2026-02-01
OBT-006	Betadine Gargle 190ml	Cairan, Obat Kumur, Antiseptik Mulut	15	25000	35000	2029-10-10
MED-ASP	Aspirin Cardio	Tablet, Pengencer Darah, Jantung	200	1500	2500	2027-03-20
MED-CLO	Clopidogrel 75 mg	Tablet, Antiplatelet, Obat Jantung/Stroke	100	3000	5000	2027-04-15

OBT-009	Atorvastatin 20mg	Tablet, Statin, Penurun Kolesterol	80	3000	4500	2026-12-12
OBT-010	OBH Sirup Batuk	Sirup, Ekspektoran, Obat Batuk Hitam	12	8000	12000	2025-12-30
OBT-011	CTM (Chlorpheniramine)	Tablet, Antihistamin, Obat Alergi	1000	50	200	2030-01-01
OBT-012	Sanmol Drop Bayi	Drops/Tetes, Antipiretik, Bayi < 1 Tahun	5	18000	25000	2026-06-01
89922 1	Puyer Batuk Pilek Racikan	Serbuk/Puyer, Racikan, Batuk & Flu	50	2000	5000	2026-01-10
MED-AML	Amlodipine 10mg	Tablet, Calcium Channel Blocker, Anti Hipertensi	40	800	1500	2027-08-08
MED-BIS	Bisoprolol 5mg	Tablet, Beta Blocker, Gagal Jantung	60	2000	3000	2027-09-09
ALKES -01	Perban Gulung 10cm	Roll, Alat Kesehatan, Pembalut Luka	20	3000	5000	-

ALKES -02	Betadine Salep Luka	Salep/Ointment, Antiseptik Kulit, Luka Luar	10	10000	15000	2028-02-02
OBT-L IST	Listerine Cool Mint	Cairan, Mouthwash, Penyegar Mulut	8	20000	28000	2027-05-05
1020	Furosemide 40mg	Tablet, Diuretik, Pembuang Cairan Tubuh	100	400	800	2026-11-11
1021	Spironolactone 25mg	Tablet, Diuretik Hemat Kalium, Hipertensi	90	800	1200	2026-10-10
HERB AL-1	Tolak Angin Cair	Cairan Sachet, Obat Herbal, Masuk Angin	200	3000	4500	2026-12-31
ORTH O-PKT	Paket Ortho (Bracket+Kawat)	Set Alat, Orthodontic, Gigi	2	2500000	3500000	-
ORTH O-TB	Sikat Gigi Ortho	Alat, Sikat Gigi Khusus, Kebersihan Mulut	15	25000	35000	-
OBT-025	Oralit Sachet	Serbuk, Rehidrasi, Obat Diare	300	500	1000	2029-01-01

OBT-026	Zinc Dispersible	Tablet Larut, Suplemen Mineral, Diare Anak	150	1000	1500	2028-05-05
OBT-027	Lacto-B Sachet	Serbuk, Probiotik, Kesehatan Pencernaan	50	6000	8000	2026-07-20
OBT-028	Omeprazole 20mg	Kapsul, PPI (Proton Pump Inhibitor), Lambung	100	500	1000	2027-05-05
OBT-029	Antasida Doen	Tablet Kunyah, Antasida, Maag/Asam Lambung	200	200	500	2028-01-01
OBT-030	Salep Gatal	Cream, Kortikosteroid/Antijamur, Kulit	25	5000	8000	2026-11-20
9999 9	Kapas Steril	Pack, Alat Kesehatan, Pembersih Luka	50	2000	3000	-

1.2 Analisis Tabel

Tabel Log Transaksi Harian = Terjadi banyak nya redundansi data dan ketidakrapian data. Ditinjau pada kolom Info_Pasien_Lengkap yang memiliki

data yang terulang, serta kategorisasi data jamak yang memungkinkan untuk adanya penumpukan data.

Tabel Stock Gudang = Kode barang yang tidak konsisten untuk setiap produk. Di rekomendasikan untuk membuat data dengan format yang lebih konsisten.

Diatas merupakan bentuk tabel UNF (Unnormalized Normal Form) yang berisi daftar tabel yang berisi data bertumpuk dan memiliki data yang belum lengkap. Dibawah ini merupakan step-by-step pembersihan datanya.

1.3 Normalisasi 1NF

NORMALISASI 1NF	
TRANSAKSI	
PK	
	<code>id_transaksi</code>
	<code>tanggal_transaksi</code>
	<code>nama_pasien</code>
	<code>jenis_kelamin_pasien</code>
	<code>alamat_pasien</code>
	<code>nama_dokter</code>
	<code>keluhan_pasien</code>
	<code>total_biaya_pasien</code>

DETAIL_ITEM	
PK	
	<code>id_detail</code>
	<code>id_transaksi</code>
	<code>jenis_transaksi</code>
	<code>jenis_item</code>
	<code>nama_item</code>
	<code>quantity</code>
	<code>harga_satuan</code>
	<code>sub_total</code>

Pada normalisasi 1NF, telah dihilangkan data yang bertumpuk, dan telah ditambahkan variabel unik seperti "id_transaksi" agar tidak terjadi data kembar.

1.4 Normalisasi 2NF

Normalisasi 2NF					
PASIEN			DOKTER		SPEASIALISASI_DOKTER
PK	id_pasien		PK	Nama_dokter	
	nama_pasien			id_dokter	
	jenis_kelamin			no_telp	
	alamat_pasien			email_dokter	
	kota_pasien		FK	id_spesialisasi	
	no_telp			nama_spesialisasi	
				deskripsi	
OBAT			TRANSAKSI		DETAIL_TRANSAKSI_OBAT
PK	id_obat		PK	id_transaksi	
	nama_obat			tanggal_transaksi	
	stok_obat		FK	id_pasien	
	harga_obat		FK	id_dokter	
FK	id_kategori			keluhan	
				total_biasa	

Dengan adanya pengurangan redundansi data, dan pembuatan junction table sebagai langkah preventif. Atribut sudah memiliki induk sendiri dan data dokter sudah dapat dipisah.

1.5 Normalisasi 3NF

Normalisasi 3NF

KATEGORI_OBAT	
PK	id_kategori
	nama_kategori
	bentuk_sediaan
	golongan
	indikasi

OBAT	
PK	id_obat
	nama_obat
	stok
	harga_beli
	harga_jual
	expired_date
	is_active
FK	id_kategori

TRANSAKSI	
PK	id_transaksi
	tanggal
	id_pasien
	id_dokter
	keluhan
	diagnosa
	status

DETAIL_TRANSAKSI	
PK	id_detail
FK	id_transaksi
FK	id_obat
	quantity
	harga_satuan
	subtotal

Hasil normalisasi 3NF adalah tidak ada transitive dependency, yaitu atribut non-key (termasuk juga FK) hanya bergantung pada PK. Kategorisasi di persingkat, seperti di kategori_obat yang memiliki atribut sendiri. Dengan normalisasi ini, apabila obat

terhapus maka data di kategori tidak hilang. Melainkan di datanya. Dan juga berlaku untuk nama kategori juga ter update pada 1 tabel data

1.6 ERD

<https://drive.google.com/file/d/1R4XL23-BqvIMNicWp5sRM7OkJaVcZPVH/view?usp=sharing>

Diatas merupakan sistem ERD dengan tabel data yang menyesuaikan dengan normalisasinya.

2.1 SQL

Dibawah ini merupakan querry untuk menjawab seluruh pertanyaan :

```
--  
=====  
==  
-- B.1.1 BUAT DATABASE  
--  
=====  
==  
-- Buat database baru bernama rsit_db  
CREATE DATABASE rsit_db;  
USE rsit_db;  
  
--  
=====  
==  
-- B.1.2 TABEL MASTER (Data Pokok)  
--  
=====  
==
```

-- TABEL 1: SPESIALISASI (Poli: Umum, Gigi, Jantung, Anak)

-- Kenapa: Setiap dokter punya spesialisasi

CREATE TABLE spesialisasi (

id_spesialisasi INT PRIMARY KEY AUTO_INCREMENT,

nama_spesialisasi VARCHAR(50) NOT NULL,

deskripsi TEXT

);

-- TABEL 2: KATEGORI OBAT (Tablet, Sirup, Kapsul, dll)

-- Kenapa: Obat perlu dikelompokkan

CREATE TABLE kategori_obat (

id_kategori INT PRIMARY KEY AUTO_INCREMENT,

nama_kategori VARCHAR(50),

bentuk_sediaan VARCHAR(20)

);

-- TABEL 3: DOKTER

-- Kenapa: Simpan data tenaga medis

-- Foreign Key: id_spesialisasi → menghubungkan ke tabel spesialisasi

CREATE TABLE dokter (

id_dokter INT PRIMARY KEY AUTO_INCREMENT,

nama_dokter VARCHAR(100),

id_spesialisasi INT,

no_telp VARCHAR(15),

FOREIGN KEY (id_spesialisasi) REFERENCES spesialisasi(id_spesialisasi)

);

-- TABEL 4: PASIEN

-- Kenapa: Simpan data pasien yang berobat

CREATE TABLE pasien (

id_pasien INT PRIMARY KEY AUTO_INCREMENT,

nama_pasien VARCHAR(100),

jenis_kelamin ENUM('L', 'P'), -- L = Laki-laki, P = Perempuan

alamat VARCHAR(255),

```

kota VARCHAR(50),
no_telp VARCHAR(15)
);

-- TABEL 5: OBAT
-- Kenapa: Simpan master data obat
-- Foreign Key: id_kategori → menghubungkan ke tabel kategori_obat
CREATE TABLE obat (
    id_obat VARCHAR(20) PRIMARY KEY, -- Contoh: OBT-001
    nama_obat VARCHAR(100),
    id_kategori INT,
    stok INT DEFAULT 0,
    harga_beli DECIMAL(12,2),
    harga_jual DECIMAL(12,2),
    FOREIGN KEY (id_kategori) REFERENCES kategori_obat(id_kategori)
);

-- =====
-- =====
-- B.I.3 TABEL TRANSAKSI (Data Operasional)
-- =====
-- =====

-- TABEL 6: TRANSAKSI (Header/Nota utama)
-- Kenapa: Simpan setiap kunjungan pasien
-- Foreign Key: id_pasien dan id_dokter → menghubungkan ke pasien & dokter
CREATE TABLE transaksi (
    id_transaksi INT PRIMARY KEY AUTO_INCREMENT,
    tanggal DATETIME DEFAULT NOW(),
    id_pasien INT,
    id_dokter INT,
    keluhan TEXT,

```

```
total_biaya DECIMAL(12,2) DEFAULT 0,  
FOREIGN KEY (id_pasien) REFERENCES pasien(id_pasien),  
FOREIGN KEY (id_dokter) REFERENCES dokter(id_dokter)  
);
```

-- TABEL 7: DETAIL OBAT (Obat yang diresepkan)

-- Kenapa: Satu transaksi bisa banyak obat
-- Foreign Key: id_transaksi → transaksi, id_obat → obat

```
CREATE TABLE detail_obat (
```

```
    id_detail INT PRIMARY KEY AUTO_INCREMENT,  
    id_transaksi INT,  
    id_obat VARCHAR(20),  
    quantity INT,  
    harga_satuan DECIMAL(12,2),  
    FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi),  
    FOREIGN KEY (id_obat) REFERENCES obat(id_obat)
```

```
);
```

--

=====

--

-- B.1.4 TABEL LOG (Audit Trail)

--

=====

--

-- TABEL 8: LOG RESTOCK (Riwayat penambahan stok)

-- Kenapa: Lacak siapa yang menambah stok, kapan, berapa

```
CREATE TABLE log_restock (
```

```
    id_log INT PRIMARY KEY AUTO_INCREMENT,  
    id_obat VARCHAR(20),  
    jumlah_masuk INT,  
    stok_sebelum INT,  
    stok_sesudah INT,
```

```
petugas VARCHAR(50),
FOREIGN KEY (id_obat) REFERENCES obat(id_obat)
);
```

-- TABEL 9: LOG HARGA (Riwayat perubahan harga)

-- Kenapa: Lacak perubahan harga untuk audit

```
CREATE TABLE log_harga (
    id_log INT PRIMARY KEY AUTO_INCREMENT,
    id_obat VARCHAR(20),
    harga_lama DECIMAL(12,2),
    harga_baru DECIMAL(12,2),
    FOREIGN KEY (id_obat) REFERENCES obat(id_obat)
);
```

--

=====

--

-- B.2.1 ISI DATA MASTER

--

=====

--

-- Isi Spesialisasi

```
INSERT INTO spesialisasi (nama_spesialisasi, deskripsi) VALUES
('Umum', 'Poliklinik Umum'),
('Gigi', 'Spesialis Gigi'),
('Jantung', 'Spesialis Jantung'),
('Anak', 'Spesialis Anak');
```

-- Isi Kategori Obat

```
INSERT INTO kategori_obat (nama_kategori, bentuk_sediaan) VALUES
('Analgesik', 'Tablet'),
('Antibiotik', 'Kapsul'),
```

```
('Vitamin', 'Tablet'),  
('Antiseptik', 'Cairan');
```

-- Isi Dokter

-- id_spesialisasi: 1=Umum, 2=Gigi, 3=Jantung, 4=Anak

```
INSERT INTO dokter (nama_dokter, id_spesialisasi, no_telp) VALUES  
('Dr. Hartono', 1, '081234567890'),  
('Drg. Ratna', 2, '081234567891'),  
('Dr. Susan', 3, '081234567892'),  
('Dr. Bambang', 4, '081234567893');
```

-- Isi Pasien

```
INSERT INTO pasien (nama_pasien, jenis_kelamin, alamat, kota, no_telp) VALUES  
('Budi Santoso', 'L', 'Jl. Mawar 10', 'Surabaya', '081111111111'),  
('Siti Aminah', 'P', 'Jl. Melati 5', 'Sidoarjo', '081222222222'),  
('Andi Wijaya', 'L', 'Jl. Anggrek 3', 'Gresik', '081333333333');
```

-- Isi Obat

-- id_kategori: 1=Analgesik, 2=Antibiotik, 3=Vitamin, 4=Antiseptik

```
INSERT INTO obat (id_obat, nama_obat, id_kategori, stok, harga_beli, harga_jual)  
VALUES  
('OBT-001', 'Paracetamol 500mg', 1, 100, 200, 500),  
('OBT-002', 'Amoxicillin 500mg', 2, 50, 800, 1200),  
('OBT-003', 'Vitamin C 500mg', 3, 200, 500, 1000),  
('OBT-004', 'Betadine', 4, 30, 10000, 15000);
```

--

```
=====
```

--

-- B.2.2 ISI DATA TRANSAKSI (Contoh: Pasien Budi berobat)

--

```
=====
```

--

```
-- Langkah 1: Buat transaksi header
INSERT INTO transaksi (id_pasien, id_dokter, keluhan)
VALUES (1,1,'Demam dan pusing');

-- Anggap transaksi ID yang terbuat adalah 1

-- Langkah 2: Isi detail obat yang diresepkan
-- Transaksi 1: Paracetamol 10 tablet @500 = 5000
INSERT INTO detail_obat (id_transaksi, id_obat, quantity, harga_satuan)
VALUES (1, 'OBT-001', 10, 500);

-- Transaksi 1: Vitamin C 5 tablet @1000 = 5000
INSERT INTO detail_obat (id_transaksi, id_obat, quantity, harga_satuan)
VALUES (1, 'OBT-003', 5, 1000);

-- Langkah 3: Update total biaya transaksi
UPDATE transaksi
SET total_biaya = 10000 -- 5000 + 5000
WHERE id_transaksi = 1;
```

B. Trigger dan Delimiter

```
--=====
===
-- B.3.1 STORED PROCEDURE: INPUT TRANSAKSI SEDERHANA
==
=====

-- Fungsi: Membuat transaksi baru dengan otomatis hitung total

DELIMITER //
```

```

CREATE PROCEDURE sp_TransaksiBaru(
    IN p_id_pasien INT,      -- Input: ID pasien
    IN p_id_dokter INT,     -- Input: ID dokter
    IN p_id_obat VARCHAR(20), -- Input: Kode obat
    IN p_qty INT,           -- Input: Jumlah obat
    OUT p_total DECIMAL(12,2) -- Output: Total biaya
)
BEGIN
    DECLARE v_harga DECIMAL(12,2);
    DECLARE v_id_transaksi INT;

    -- 1. Ambil harga obat dari tabel master
    SELECT harga_jual INTO v_harga FROM obat WHERE id_obat = p_id_obat;

    -- 2. Buat transaksi header
    INSERT INTO transaksi (id_pasien, id_dokter, keluhan)
    VALUES (p_id_pasien, p_id_dokter, 'Auto-generated');

    -- 3. Ambil ID transaksi yang baru dibuat
    SET v_id_transaksi = LAST_INSERT_ID();

    -- 4. Insert detail obat
    INSERT INTO detail_obat (id_transaksi, id_obat, quantity, harga_satuan)
    VALUES (v_id_transaksi, p_id_obat, p_qty, v_harga);

    -- 5. Hitung total
    SET p_total = p_qty * v_harga;

    -- 6. Update total di header
    UPDATE transaksi SET total_biaya = p_total WHERE id_transaksi = v_id_transaksi;
END //
DELIMITER ;

```

```
--  
=====  
==  
-- CARA PAKAI:  
--  
=====  
==  
-- Panggil procedure dengan parameter:  
CALL sp_TransaksiBaru(1, 1, 'OBT-001', 5, @total);  
  
-- Lihat hasil total:  
SELECT @total AS Total_Biaya;  
  
-- Lihat transaksi yang terbuat:  
SELECT * FROM transaksi ORDER BY id_transaksi DESC LIMIT 1;  
  
--  
=====  
==  
-- B.3.2 STORED PROCEDURE: RESTOCK OBAT  
--  
=====  
==  
-- Fungsi: Menambah stok obat dan catat di log  
  
DELIMITER //  
  
CREATE PROCEDURE sp_Restock(  
    IN p_id_obat VARCHAR(20), -- Kode obat  
    IN p_jumlah INT, -- Jumlah masuk  
    IN p_petugas VARCHAR(50) -- Nama petugas  
)
```

```

BEGIN

DECLARE v_stok_lama INT;
DECLARE v_stok_baru INT;

-- 1. Ambil stok lama
SELECT stok INTO v_stok_lama FROM obat WHERE id_obat = p_id_obat;

-- 2. Hitung stok baru
SET v_stok_baru = v_stok_lama + p_jumlah;

-- 3. Update stok obat
UPDATE obat SET stok = v_stok_baru WHERE id_obat = p_id_obat;

-- 4. Catat di log
INSERT INTO log_restock (id_obat, jumlah_masuk, stok_sebelum, stok_sesudah,
petugas)
VALUES (p_id_obat, p_jumlah, v_stok_lama, v_stok_baru, p_petugas);

END //


DELIMITER ;

-- =====
-- CARA PAKAI:
-- =====
-- Tambah stok Paracetamol 50 pcs oleh petugas "Admin":
CALL sp_Restock('OBT-001', 50, 'Admin');

-- Cek hasil:
SELECT * FROM obat WHERE id_obat = 'OBT-001';

```

```
SELECT * FROM log_restock WHERE id_obat = 'OBT-001';
```

```
--
```

```
=====
```

```
==
```

```
-- B.4.2 TRIGGER: AUDIT HARGA
```

```
--
```

```
=====
```

```
==
```

```
-- Fungsi: Otomatis catat jika harga obat diubah
```

```
DELIMITER //
```

```
CREATE TRIGGER audit_harga AFTER UPDATE ON obat
FOR EACH ROW
BEGIN
    -- Jika harga jual berubah, catat ke log
    IF OLD.harga_jual != NEW.harga_jual THEN
        INSERT INTO log_harga (id_obat, harga_lama, harga_baru)
        VALUES (NEW.id_obat, OLD.harga_jual, NEW.harga_jual);
    END IF;
END //
```

```
DELIMITER ;
```

```
--
```

```
=====
```

```
==
```

```
-- CARA TEST:
```

```
--
```

```
=====
```

```
==
```

```
-- Ubah harga Paracetamol dari 500 menjadi 600:
```

```
UPDATE obat SET harga_jual = 600 WHERE id_obat = 'OBT-001';
```

```
-- Cek log perubahan:  
SELECT * FROM log_harga WHERE id_obat = 'OBT-001';  
  
--  
=====  
==  
-- B.5.1 LAPORAN: KINERJA DOKTER  
--  
=====  
==  
-- Menampilkan: Nama dokter, spesialisasi, jumlah pasien, total pendapatan  
  
SELECT  
    d.nama_dokter,  
    s.nama_spesialisasi,  
    COUNT(t.id_transaksi) AS jumlah_pasien,  
    SUM(t.total_biaya) AS total_pendapatan  
FROM dokter d  
JOIN spesialisasi s ON d.id_spesialisasi = s.id_spesialisasi  
LEFT JOIN transaksi t ON d.id_dokter = t.id_dokter  
GROUP BY d.id_dokter, d.nama_dokter, s.nama_spesialisasi;  
  
--  
=====  
==  
-- HASIL CONTOH:  
--  
=====  
==  
-- nama_dokter | nama_spesialisasi | jumlah_pasien | total_pendapatan  
-- Dr. Hartono | Umum          | 5           | 2500000  
-- Drg. Ratna  | Gigi          | 3           | 5000000  
-- Dr. Susan   | Jantung        | 2           | 7500000
```

3.1 NoSQL

```
//  
=====  
==  
// C.1.1 GUNAKAN DATABASE  
//  
=====  
==  
use rsit_nosql;  
  
//  
=====  
==  
// C.1.2 INSERT DATA REKAM MEDIS (Poli Umum - Sederhana)  
//  
=====  
==  
// Struktur: Data pasien poli umum dengan gejala sederhana  
  
db.rekam_medis.insertOne({  
    nama_pasien: "Budi Santoso",  
    no_rm: "RM-001",  
    poli: "Umum",  
    tanggal: new Date("2026-02-01"),  
    status: "Rawat Jalan",  
    gejala: ["Demam", "Pusing", "Batuk"],  
    tanda_vital: {  
        suhu: 38.5,  
        tensi: "120/80",  
        nadi: 90  
    },  
    diagnosa: "Demam Viral",  
    biaya: 85000
```

```
});  
  
//  
=====  
==  
// C.1.3 INSERT DATA REKAM MEDIS (Poli Gigi - Kompleks)  
//  
=====  
==  
// Struktur: Array of objects untuk kondisi gigi  
  
db.rekam_medis.insertOne({  
    nama_pasien: "Siti Aminah",  
    no_rm: "RM-002",  
    poli: "Gigi",  
    tanggal: new Date("2026-02-01"),  
    status: "Rawat Jalan",  
    odontogram: [ // Array of objects  
        { gigi: "Graham Atas Kanan", kondisi: "Berlubang", tindakan: "Tambal" },  
        { gigi: "Graham Bawah Kiri", kondisi: "Karang Gigi", tindakan: "Scaling" }  
    ],  
    catatan: "Pasien takut jarum suntik",  
    biaya: 750000  
});  
  
//  
=====  
==  
// C.1.4 INSERT DATA REKAM MEDIS (Poli Jantung - Nested Object)  
//  
=====  
==  
// Struktur: Nested object untuk hasil lab
```

```
db.rekam_medis.insertOne({
    nama_pasien: "Andi Wijaya",
    no_rm: "RM-003",
    poli: "Jantung",
    tanggal: new Date("2026-02-01"),
    status: "Rawat Jalan",
    hasil_lab: { // Nested object
        kolesterol: { total: 240, ldl: 160, hdl: 40 },
        gula_darah: { puasa: 110, post_prandial: 140 }
    },
    diagnosa: "Hipertensi",
    biaya: 1200000
});

// =====
==

// C.1.5 INSERT BANYAK DATA SEKALIGUS (Rawat Inap)
// =====
==

db.rekam_medis.insertMany([
    {
        nama_pasien: "Ahmad Fauzi",
        no_rm: "RM-004",
        poli: "Jantung",
        tanggal_masuk: new Date("2026-02-05"),
        status: "Rawat Inap",
        kategori: "VIP",
        biaya_kamar: 2000000
    },
    {
        nama_pasien: "Rina Wulandari",

```

```

        no_rm: "RM-005",
        poli: "Umum",
        tanggal_masuk: new Date("2026-02-08"),
        status: "Rawat Inap",
        kategori: "VIP",
        biaya_kamar: 1500000
    },
    {
        nama_pasien: "Bambang Sutejo",
        no_rm: "RM-006",
        poli: "Jantung",
        tanggal_masuk: new Date("2026-02-01"),
        status: "Rawat Inap",
        kategori: "Reguler",
        biaya_kamar: 500000
    }
]);

```

- Query dasar Find :

```

// =====
== 
// C.2.1 CARI SEMUA PASIEN
// =====
== 
db.rekam_medis.find();

// =====
== 
// C.2.2 CARI PASIEN POLI TERTENTU

```

```
//  
=====  
==  
// Cari pasien poli Umum saja  
db.rekam_medis.find({ poli: "Umum" });  
  
//  
=====  
==  
// C.2.3 CARI PASIEN DENGAN KONDISI TERTENTU  
//  
=====  
==  
// Cari pasien rawat inap saja  
db.rekam_medis.find({ status: "Rawat Inap" });  
  
//  
=====  
==  
// C.2.4 CARI PASIEN DENGAN BIAYA TERTENTU  
//  
=====  
==  
// Cari pasien dengan biaya > 500000  
db.rekam_medis.find({ biaya: { $gt: 500000 } });  
  
//  
=====  
==  
// C.2.5 TAMPILKAN FIELD TERTENTU SAJA  
//  
=====  
==  
// Hanya tampilkan nama dan poli (sembunyikan _id)
```

```
db.rekam_medis.find({}, { nama_pasien: 1, poli: 1, _id: 0 });

    - Update Data :
// =====
==  

// C.3.1 UPDATE SATU DOKUMEN
// =====
==  

// Ubah diagnosa Budi Santoso
db.rekam_medis.updateOne(
  { nama_pasien: "Budi Santoso" }, // Kondisi cari
  { $set: { diagnosa: "Demam Berdarah" } } // Data baru
);

// =====
==  

// C.3.2 UPDATE MASSAL: TAMBAH JADWAL KONTROL (H+7)
// =====
==  

// Untuk semua pasien rawat inap, tambahkan jadwal kontrol 7 hari lagi

db.rekam_medis.updateMany(
  { status: "Rawat Inap" }, // Kondisi: yang rawat inap saja
  {
    $set: {
      jadwal_kontrol: new Date(new Date().getTime() + 7*24*60*60*1000)
    }
  }
);
```

```
//  
=====  
==  
// C.3.3 UPDATE: NAIKKAN BIAYA KAMAR 10% UNTUK VIP  
//  
=====  
==  
  
db.rekam_medis.updateMany(  
  { kategori: "VIP", status: "Rawat Inap" }, // Kondisi: VIP & rawat inap  
  { $mul: { biaya_kamar: 1.10 } } // Kalikan dengan 1.10 (naik 10%)  
);  
  
// Cek hasil:  
db.rekam_medis.find({ kategori: "VIP" }, { nama_pasien: 1, biaya_kamar: 1 });  
  
- Aggregation :  
  
//  
=====  
==  
// C.4.1 HITUNG JUMLAH PASIEN PER POLI  
//  
=====  
==  
  
db.rekam_medis.aggregate([  
  { $group: { _id: "$poli", jumlah: { $sum: 1 } } },  
  { $sort: { jumlah: -1 } }  
]);  
  
// Hasil contoh:  
// { "_id": "Umum", "jumlah": 10 }  
// { "_id": "Gigi", "jumlah": 5 }
```

```
// { "_id": "Jantung", "jumlah": 3 }

// =====
==

// C.4.2 HITUNG TOTAL BIAYA PER POLI
//
=====

==

db.rekam_medis.aggregate([
  { $group: {
    _id: "$poli",
    total_biaya: { $sum: "$biaya" },
    rata_rata: { $avg: "$biaya" }
  }}
]);

// =====
==

// C.4.3 ANALISIS GEJALA PALING SERING (TOP 5)
//
=====

==

db.rekam_medis.aggregate([
  // Pecah array gejala menjadi dokumen terpisah
  { $unwind: "$gejala" },

  // Kelompokkan dan hitung
  { $group: { _id: "$gejala", frekuensi: { $sum: 1 } } },

  // Urutkan dari yang terbanyak
```

```

{ $sort: { frekuensi: -1 } },

// Ambil 5 teratas
{ $limit: 5 }
]);


// Hasil contoh:
// { "_id": "Demam", "frekuensi": 50 }
// { "_id": "Batuk", "frekuensi": 30 }
// { "_id": "Pusing", "frekuensi": 25 }

// =====
==

// C.4.4 STATISTIK PASIEN USIA > 40 (RISIKO TINGGI)
// =====
==

// Asumsikan kita punya field usia
db.rekam_medis.aggregate([
    // Filter usia > 40
    { $match: { usia: { $gt: 40 } } },

    // Kelompokkan berdasarkan diagnosa
    { $group: {
        _id: "$diagnosa",
        jumlah_pasien: { $sum: 1 },
        rata_rata_biaya: { $avg: "$biaya" }
    }},
    // Urutkan dari risiko tertinggi (biaya tertinggi)
    { $sort: { rata_rata_biaya: -1 } }
]);



```

