IF4040 - Pemodelan Data Lanjut
Project 4

Disusun Oleh:

Kelompok 5

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2025

IDENTITAS KELOMPOK

Nomor Kelompok : 5

Anggota Kelompok : 1. Agil Fadhilah Sabri (13522006)
2. Bryan Cornelius Lauwrence (13522033)
3. Dhidit Abdi Aziz (13522040)
4. Maximilian Sulistiyo (13522061)

Kelas : K1

Dosen Pengampu : 1. Dr. Fazat Nur Azizah, S.T., M.Sc.
2. Tricya Esterina Widagdo, S.T., M.Sc.

1. EKSPLORASI BASIS DATA UNTUK DATA TEMPORAL

Untuk kasus ini, basis data relasional yang dipilih adalah PostgreSQL. PostgreSQL tidak memiliki *built-in system versioning* yang dapat menyimpan aspek temporal basis data sehingga perlu pembuatan secara manual. Basis data PostgreSQL menyediakan berbagai tipe data yang berkaitan dengan waktu sebagai berikut.

Tipe Data	Deskripsi
DATE	Menyimpan tanggal
TIME	Menyimpan waktu
TIMESTAMP	Menyimpan kombinasi tanggal dan waktu
TIMESTAMPZ	Seperti timestamp dengan tambahan zona waktu
INTERVAL	Durasi waktu

Dengan tipe-tipe tersebut, PostgreSQL dapat menerima kolom yang menyatakan dimensi waktu. Cotohnya, DATE ataupun TIMESTAMP dapat disimpan sebagai tipe untuk *valid time* atau *transaction time* sesuai kebutuhan.
Selain tipe data waktu, PostgreSQL pun menyediakan tipe data interval, yaitu:

Tipe Data	Deskripsi
TSRANGE	Interval timestamp
TSTZRANGE	Interval timestamp dengan zona waktu
DATERANGE	Interval tanggal

Tipe data interval tersebut dapat digunakan sebagai rentang dimensi waktu. Selain itu, tipe data interval memiliki operator berupa *overlap* (&&), *contains* (@>), *before* (<<), *after* (>>), dan *adjacent* (-|-).

Terakhir, PostgreSQL menyediakan operasi <, <=, >, dan >= untuk tipe data waktu. Fitur tersebut dapat dimanfaatkan untuk membuat *timeslice*. Memang tidak secara langsung menyediakan Allen’s 13 Relationship, tetapi keempat operasi tersebut dapat mendefinisikannya.

2. STUDI KASUS

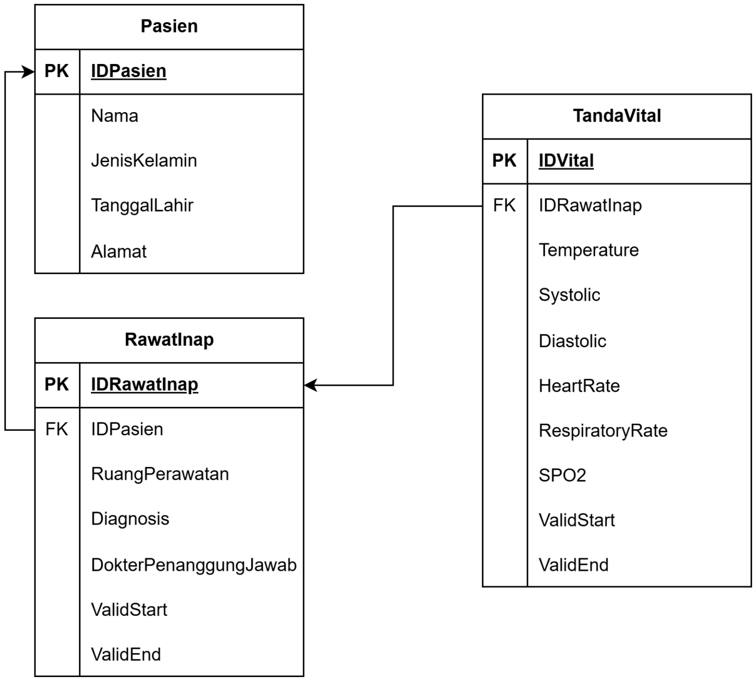
a. Deskripsi data

Data terinspirasi dari *website* nature.com, atau lebih tepatnya https://www.nature.com/articles/s41597-022-01899-x. Data tersebut berisi *health record dataset*, yang meliputi temperatur, tekanan darah, *heart rate*, dan data tanda vital lainnya. Namun, karena akses ke *dataset* tersebut dibatasi, sumber ini menjadi inspirasi kami untuk membuat suatu basis data temporal.

Data ini dapat menjadi data temporal karena datanya bersifat *time series*. Sebagai contoh, seorang pasien akan diawasi kondisi vitalnya setiap beberapa menit. Perubahan yang berkelanjutan ini dapat digunakan untuk mendeteksi munculnya penyakit pasien, reaksi akibat obat, dan lain-lain. Sebagai contoh, pasien yang awalnya sehat dan tiba-tiba melapor sakit dapat dilihat perubahan kondisi vitalnya untuk mengetahui kapan tepatnya pasien mengalami sakit secara tiba-tiba.

Selain data time series tanda vital, studi kasus ini juga memasukkan data rawat inap / admission pasien sebagai contoh data temporal yang bersifat periodik (tidak kontinu). Data admission mencatat interval waktu pasien masuk, dipindahkan, dan keluar dari unit perawatan tertentu (misalnya IGD, ICU, atau ruang rawat inap). Perubahan pada data ini tidak terjadi setiap saat, melainkan pada kejadian-kejadian tertentu yang memiliki valid time yang jelas.

b. Skema basis data



Gambar 1. Skema Relasional Basis Data Rumah Sakit

Data ini terdiri dari tiga tabel. Tabel pertama merupakan informasi umum pasien, tabel kedua berisi data rawat inap pasien sebagai data periodik dan tabel kedua berisi tanda vital pasien sebagai data *time series*. Untuk menyederhanakan persoalan, seluruh tanda vital pasien digabungkan dalam satu tabel dan *chronon*-nya disamakan. Selain itu, *chronon* untuk data ini diasumsikan setiap satu jam dengan anggapan bahwa pengukuran data baru dilakukan setiap beberapa jam. Representasi dimensi *valid time* akan ditambahkan bersamaan pada tabel TandaVital. Caranya dengan menambahkan dua kolom, yaitu ValidStart sebagai waku valid awal dan ValidEnd sebagai waktu valid selesai. Penjelasan setiap kolom dalam tabel sebagai berikut.

1. Tabel Pasien

Kolom	Tipe	Deskripsi
IDPasien	integer primary key	ID milik pasien
Nama	varchar(250)	Nama pasien
JenisKelamin	enum(‘L’, ‘P’)	L untuk pasien laki-laki dan P untuk pasien perempuan
TanggalLahir	date	Tanggal lahir pasien
Alamat	varchar(250)	Alamat tempat tinggal pasien

2. Tabel RawatInap

Kolom	Tipe	Deskripsi
IDRawatInap	integer primary key	ID rawat inap.
IDPasien	integer foreign key	*Foreign key* ke IDPasien pada tabel Pasien
RuangPerawatan	varchar(100)	Nama ruang rawat inap
Diagnosis	varchar(500)	Hasil diagnosis pasien
DokterPenanggungJawab	varchar(250)	Nama dokter yang bertanggung jawab terhadap pasien
ValidStart	timestamp	Waktu data valid di dunia nyata
ValidEnd	timestamp	Waktu data sudah tidak valid di dunia nyata, bernilai null jika data tidak tertutup

3. Tabel TandaVital

Kolom	**Tipe**	**Deskripsi**
IDVital	integer primary key	ID per *chronon* untuk tiap pasien
IDRawatInap	integer foreign key	*Foreign key* ke IDRawatInap pada tabel RawatInap
Temperature	decimal(4,2)	Suhu tubuh pasien dalam °C
Systolic	integer	Tekanan darah pasien (systolic/diastolic), masing-masing satuan mmHg
Diastolic	integer	
HeartRate	integer	Jumlah denyut nadi per menit
RespiratoryRate	integer	Jumlah napas per menit
SPO2	integer	Persentase saturasi oksigen
ValidStart	timestamp	Waktu data valid di dunia nyata
ValidEnd	timestamp	Waktu data sudah tidak valid di dunia nyata, bernilai null jika data tidak tertutup

c. *User defined functions*

1. *Insert*, *Update*, dan *Delete*

Insert

```
-- Insert digunakan untuk memasukkan data baru ke dalam tabel RawatInap dan
TandaVital
CREATE OR REPLACE FUNCTION InsertRawatInap(
    p_IDPasien INT,
    p_RuangPerawatan VARCHAR,
    p_Diagnosis VARCHAR,
    p_DokterPenanggungjawab VARCHAR,
    p_ValidStart TIMESTAMP
)
RETURNS VOID AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM RawatInap
        WHERE IDPasien = p_IDPasien
          AND ValidEnd IS NULL
    ) THEN
        UPDATE RawatInap
        SET ValidEnd = p_ValidStart
        WHERE IDPasien = p_IDPasien
          AND ValidEnd IS NULL;
    END IF;

    INSERT INTO RawatInap (
        IDPasien,
        RuangPerawatan,
        Diagnosis,
        DokterPenanggungjawab,
        ValidStart,
        ValidEnd
    )
    VALUES (
        p_IDPasien,
        p_RuangPerawatan,
        p_Diagnosis,
        p_DokterPenanggungjawab,
        p_ValidStart,
        NULL
    );
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION InsertTandaVital(
    p_IDRawatInap INT,
    p_Temperature DECIMAL,
    p_Systolic INT,
    p_Diastolic INT,
    p_HeartRate INT,
    p_RespiratoryRate INT,
    p_SPO2 INT,
    p_ValidStart TIMESTAMP
)
RETURNS VOID AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM TandaVital
        WHERE IDRawatInap = p_IDRawatInap
          AND ValidEnd IS NULL
    ) THEN
        UPDATE TandaVital
        SET ValidEnd = p_ValidStart
        WHERE IDRawatInap = p_IDRawatInap
          AND ValidEnd IS NULL;
    END IF;

    INSERT INTO TandaVital (
        IDRawatInap,
        Temperature,
        Systolic,
        Diastolic,
        HeartRate,
        RespiratoryRate,
        SPO2,
        ValidStart
    )
    VALUES (
        p_IDRawatInap,
        p_Temperature,
        p_Systolic,
        p_Diastolic,
        p_HeartRate,
        p_RespiratoryRate,
        p_SPO2,
        p_ValidStart
    );
END;
$$ LANGUAGE plpgsql;
```

Update

```
-- Update digunakkan untuk meralat data yang salah pada tabel RawatInap dan TandaVital
CREATE OR REPLACE FUNCTION UpdateRawatInap(
    p_IDPasien INT,
    p_RuangPerawatan VARCHAR,
    p_Diagnosis VARCHAR,
    p_DokterPenanggungjawab VARCHAR
)
RETURNS VOID AS $$
```

```
BEGIN
    IF EXISTS (
        SELECT 1
        FROM RawatInap
        WHERE IDPasien = p_IDPasien
          AND ValidEnd IS NULL
    ) THEN
        UPDATE RawatInap
        SET RuangPerawatan = p_RuangPerawatan,
            Diagnosis = p_Diagnosis,
            DokterPenanggungjawab = p_DokterPenanggungjawab
        WHERE IDPasien = p_IDPasien
          AND ValidEnd IS NULL;
    END IF;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION UpdateTandaVital(
    p_IDRawatInap INT,
    p_Temperature DECIMAL,
    p_Systolic INT,
    p_Diastolic INT,
    p_HeartRate INT,
    p_RespiratoryRate INT,
    p_SPO2 INT
)
RETURNS VOID AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM TandaVital
        WHERE IDRawatInap = p_IDRawatInap
          AND ValidEnd IS NULL
    ) THEN
        UPDATE TandaVital
        SET Temperature = p_Temperature,
            Systolic = p_Systolic,
            Diastolic = p_Diastolic,
            HeartRate = p_HeartRate,
            RespiratoryRate = p_RespiratoryRate,
            SPO2 = p_SPO2
        WHERE IDRawatInap = p_IDRawatInap
          AND ValidEnd IS NULL;
    END IF;
END;
$$ LANGUAGE plpgsql;
```

Delete

```
-- Delete digunakan untuk menghapus data yang valid saat ini pada tabel
RawatInap dengan menandai
CREATE OR REPLACE FUNCTION DeleteRawatInap(
    p_IDPasien INT,
    p_ValidEnd TIMESTAMP
)
RETURNS VOID AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM RawatInap
        WHERE IDPasien = p_IDPasien
          AND ValidEnd IS NULL
    ) THEN
        UPDATE RawatInap
        SET ValidEnd = p_ValidEnd
        WHERE IDPasien = p_IDPasien
          AND ValidEnd IS NULL;
    END IF;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION DeleteTandaVital(
    p_IDRawatInap INT,
    p_ValidEnd TIMESTAMP
)
RETURNS VOID AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM TandaVital
          AND IDRawatInap = p_IDRawatInap
          AND ValidEnd IS NULL
    ) THEN
        UPDATE TandaVital
        SET ValidEnd = p_ValidEnd
          AND IDRawatInap = p_IDRawatInap
          AND ValidEnd IS NULL;
    END IF;
END;
$$ LANGUAGE plpgsql;
```

2. *Allen’s 13 Relationship* dan Predikat Temporal

Helper

```
-- Helper
-- menangani ValidEnd yang NULL sebagai Infinity
CREATE OR REPLACE FUNCTION get_end_time(ts TIMESTAMP)
RETURNS TIMESTAMP AS $$
BEGIN
    RETURN COALESCE(ts, 'infinity'::TIMESTAMP);
END;
$$ LANGUAGE plpgsql IMMUTABLE;
```

```
CREATE OR REPLACE FUNCTION temporal_intersects(s1 TIMESTAMP, e1 TIMESTAMP,
s2 TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN s1 < get_end_time(e2)  AND s2 < get_end_time(e1);
END;
$$ LANGUAGE plpgsql IMMUTABLE;
```

Allen’s 13 Relationship

```
-- 1. BEFORE (X < Y): X terjadi sepenuhnya sebelum Y
CREATE OR REPLACE FUNCTION allen_before(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN get_end_time(e1) < s2;
END;
$$ LANGUAGE plpgsql IMMUTABLE;

-- 2. AFTER (Y > X): Kebalikan dari Before
CREATE OR REPLACE FUNCTION allen_after(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN s1 > get_end_time(e2);
END;
$$ LANGUAGE plpgsql IMMUTABLE;

-- 3. MEETS (X m Y): X bertemu langsung dengan Y (akhir X = awal Y)
CREATE OR REPLACE FUNCTION allen_meets(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN get_end_time(e1) = s2;
END;
$$ LANGUAGE plpgsql IMMUTABLE;

-- 4. MET BY (Y mi X): Kebalikan dari Meets
CREATE OR REPLACE FUNCTION allen_met_by(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN s1 = get_end_time(e2);
END;
$$ LANGUAGE plpgsql IMMUTABLE;

-- 5. OVERLAPS (X o Y): X mulai sebelum Y, dan berakhir di tengah-tengah Y
CREATE OR REPLACE FUNCTION allen_overlaps(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN s1 < s2
```

```
       AND get_end_time(e1) > s2
       AND get_end_time(e1) < get_end_time(e2);
END;
$$ LANGUAGE plpgsql IMMUTABLE;

-- 6. OVERLAPPED BY (Y oi X): Kebalikan dari Overlaps
CREATE OR REPLACE FUNCTION allen_overlapped_by(s1 TIMESTAMP, e1 TIMESTAMP,
s2 TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN s1 > s2
       AND s1 < get_end_time(e2)
       AND get_end_time(e1) > get_end_time(e2);
END;
$$ LANGUAGE plpgsql IMMUTABLE;

-- 7. STARTS (X s Y): Mulai bersamaan, tapi X selesai lebih dulu
CREATE OR REPLACE FUNCTION allen_starts(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN s1 = s2
       AND get_end_time(e1) < get_end_time(e2);
END;
$$ LANGUAGE plpgsql IMMUTABLE;

-- 8. STARTED BY (Y si X): Mulai bersamaan, tapi X selesai belakangan
CREATE OR REPLACE FUNCTION allen_started_by(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN s1 = s2
       AND get_end_time(e1) > get_end_time(e2);
END;
$$ LANGUAGE plpgsql IMMUTABLE;

-- 9. DURING (X d Y): X sepenuhnya ada di dalam Y
CREATE OR REPLACE FUNCTION allen_during(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN s1 > s2
       AND get_end_time(e1) < get_end_time(e2);
END;
$$ LANGUAGE plpgsql IMMUTABLE;

-- 10. CONTAINS (Y di X): Y sepenuhnya ada di dalam X
CREATE OR REPLACE FUNCTION allen_contains(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN s1 < s2
       AND get_end_time(e1) > get_end_time(e2);
END;
```

```
$$ LANGUAGE plpgsql IMMUTABLE;

-- 11. FINISHES (X f Y): Selesai bersamaan, tapi X mulai belakangan
CREATE OR REPLACE FUNCTION allen_finishes(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN get_end_time(e1) = get_end_time(e2)
       AND s1 > s2;
END;
$$ LANGUAGE plpgsql IMMUTABLE;

-- 12. FINISHED BY (Y fi X): Selesai bersamaan, tapi X mulai lebih awal
CREATE OR REPLACE FUNCTION allen_finished_by(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN get_end_time(e1) = get_end_time(e2)
       AND s1 < s2;
END;
$$ LANGUAGE plpgsql IMMUTABLE;

-- 13. EQUALS (X = Y): Identik
CREATE OR REPLACE FUNCTION allen_equals(s1 TIMESTAMP, e1 TIMESTAMP, s2
TIMESTAMP, e2 TIMESTAMP)
RETURNS BOOLEAN AS $$
BEGIN
    RETURN s1 = s2
       AND get_end_time(e1) = get_end_time(e2);
END;
$$ LANGUAGE plpgsql IMMUTABLE;
```

Predikat Temporal: *Changed*

```
-- ===========================
-- Changed
-- Mengecek apakah ada perubahan nilai pada suatu kolom dalam interval waktu
tertentu
-- ===========================
CREATE OR REPLACE FUNCTION temporal_changed(
    _table_name TEXT,
    _id_col TEXT,
    _id_val INT,
    _target_col TEXT,
    _start_time TIMESTAMP,
    _end_time TIMESTAMP
)
RETURNS BOOLEAN AS $$
DECLARE
    _is_changed BOOLEAN;
    _sql TEXT;
BEGIN
```

```
    _sql := format(
        'SELECT COUNT(DISTINCT %I) > 1
         FROM %I
         WHERE %I = %L
         AND temporal_intersects(ValidStart, ValidEnd, %L, %L)',
         _target_col, _table_name, _id_col, _id_val, _start_time, _end_time
    );

    EXECUTE _sql INTO _is_changed;
    RETURN COALESCE(_is_changed, FALSE);
END;
$$ LANGUAGE plpgsql STABLE;
```

Predikat Temporal: *Trend*

```
-- ===========================
-- Trend
-- Mengecek tren data: INCREASING, DECREASING, STABLE, FLUCTUATING
-- ===========================
CREATE OR REPLACE FUNCTION temporal_trend(
    _table_name TEXT,
    _id_col TEXT,
    _id_val INT,
    _target_col TEXT,
    _start_time TIMESTAMP,
    _end_time TIMESTAMP
)
RETURNS TEXT AS $$
DECLARE
    _is_increasing BOOLEAN;
    _is_decreasing BOOLEAN;
    _sql TEXT;
BEGIN
    _sql := format(
        'WITH OrderedData AS (
            SELECT %I::FLOAT as val, ValidStart
            FROM %I
            WHERE %I = %L
            AND temporal_intersects(ValidStart, ValidEnd, %L, %L)
            ORDER BY ValidStart ASC
        ),
        TrendCheck AS (
            SELECT
                BOOL_AND(next_val >= val) as all_increasing,
                BOOL_AND(next_val <= val) as all_decreasing
            FROM (
                SELECT val, LEAD(val) OVER (ORDER BY ValidStart) as next_val
                FROM OrderedData
            ) sub
            WHERE next_val IS NOT NULL
        )
        SELECT all_increasing, all_decreasing FROM TrendCheck;',
        _target_col, _table_name, _id_col, _id_val, _start_time, _end_time
    );

    EXECUTE _sql INTO _is_increasing, _is_decreasing;

    IF _is_increasing AND NOT _is_decreasing THEN RETURN 'INCREASING';
    ELSIF _is_decreasing AND NOT _is_increasing THEN RETURN 'DECREASING';
    ELSIF _is_increasing AND _is_decreasing THEN RETURN 'STABLE';
    ELSE RETURN 'FLUCTUATING';
    END IF;
END;
$$ LANGUAGE plpgsql STABLE;
```

Predikat Temporal: *Speed*

```
-- ===========================
-- Speed
-- Menghitung rata-rata kecepatan perubahan
-- ===========================
CREATE OR REPLACE FUNCTION temporal_speed(
    _table_name TEXT,
    _id_col TEXT,
    _id_val INT,
    _target_col TEXT,
    _start_time TIMESTAMP,
    _end_time TIMESTAMP
)
RETURNS FLOAT AS $$
DECLARE
    _delta_val FLOAT;
    _delta_time_hours FLOAT;
    _sql TEXT;
BEGIN
    -- Mengambil selisih nilai akhir dan nilai awal, serta selisih waktu
    _sql := format(
        'WITH Data AS (
            SELECT %I::FLOAT as val, EXTRACT(EPOCH FROM ValidStart) as ts
            FROM %I
            WHERE %I = %L AND temporal_intersects(ValidStart, ValidEnd, %L,
%L)
        ),
        Stats AS (
            SELECT
                (SELECT val FROM Data ORDER BY ts DESC LIMIT 1) - (SELECT
val FROM Data ORDER BY ts ASC LIMIT 1) as d_val,
                (SELECT ts FROM Data ORDER BY ts DESC LIMIT 1) - (SELECT ts
FROM Data ORDER BY ts ASC LIMIT 1) as d_ts
        )
        SELECT d_val, d_ts FROM Stats',
         _target_col, _table_name, _id_col, _id_val, _start_time, _end_time
    );
    EXECUTE _sql INTO _delta_val, _delta_time_hours;

    -- Konversi detik ke jam (3600), handle pembagian dengan nol
    IF _delta_time_hours IS NULL OR _delta_time_hours = 0 THEN
        RETURN 0.0;
    END IF;

    RETURN _delta_val / (_delta_time_hours / 3600.0);
END;
$$ LANGUAGE plpgsql STABLE;
```

Predikat Temporal: *Acceleration*

```
-- ===========================
-- Acceleration
-- Menghitung rata-rata percepatan perubahan
-- ===========================
CREATE OR REPLACE FUNCTION temporal_acceleration(
    _table_name TEXT,
    _id_col TEXT,
    _id_val INT,
    _target_col TEXT,
    _start_time TIMESTAMP,
    _end_time TIMESTAMP
)
RETURNS FLOAT AS $$
DECLARE
    _mid_time TIMESTAMP;
    _speed_1 FLOAT;
    _speed_2 FLOAT;
    _total_hours FLOAT;
BEGIN
    _mid_time := _start_time + ((_end_time - _start_time) / 2);

    -- Hitung speed awal (memanggil fungsi speed)
    _speed_1 := temporal_speed(_table_name, _id_col, _id_val, _target_col,
_start_time, _mid_time);

    -- Hitung speed akhir
    _speed_2 := temporal_speed(_table_name, _id_col, _id_val, _target_col,
_mid_time, _end_time);

    _total_hours := EXTRACT(EPOCH FROM (_end_time - _start_time)) / 3600.0;

    IF _total_hours = 0 THEN RETURN 0; END IF;

    RETURN (_speed_2 - _speed_1) / _total_hours;
END;
$$ LANGUAGE plpgsql STABLE;
```

3. *Temporal Operations*

Temporal Projection

```
CREATE OR REPLACE FUNCTION temporal_projection(
    _table_name TEXT,
    _columns TEXT[]
)
RETURNS TABLE (
    data JSONB,
    ValidStart TIMESTAMP,
    ValidEnd TIMESTAMP
)
AS $$
DECLARE
    _cols_json TEXT;
BEGIN
    _cols_json := (
        SELECT string_agg(format('''%s'', %I', col, col), ', ')
        FROM unnest(_columns) col
    );

    RETURN QUERY EXECUTE format(
        'SELECT jsonb_build_object(%s) as data, validstart, validend
         FROM %I
         ORDER BY validstart',
        _cols_json, _table_name
    );
END;
$$ LANGUAGE plpgsql STABLE;
```

Temporal Selection

```
CREATE OR REPLACE FUNCTION temporal_selection(
    _table_name TEXT, _where_condition TEXT
)
RETURNS TABLE (
    data JSONB,
    ValidStart TIMESTAMP,
    ValidEnd TIMESTAMP
)
AS $$
BEGIN
    RETURN QUERY EXECUTE format(
        'SELECT row_to_json(t.*)::jsonb as data, t.validstart, t.validend
         FROM %I t
         WHERE %s
         ORDER BY validstart',
        _table_name, _where_condition
    );
END;
$$ LANGUAGE plpgsql STABLE;
```

Temporal Union

```
CREATE OR REPLACE FUNCTION temporal_union(
    _table_name TEXT,
    _condition1 TEXT,
    _condition2 TEXT
)
RETURNS TABLE (
    data JSONB,
    ValidStart TIMESTAMP,
    ValidEnd TIMESTAMP
)
AS $$
BEGIN
    RETURN QUERY EXECUTE format(
        'SELECT row_to_json(t.*)::jsonb as data, t.validstart, t.validend
         FROM %I t
         WHERE %s

         UNION

         SELECT row_to_json(t.*)::jsonb as data, t.validstart, t.validend
         FROM %I t
         WHERE %s

         ORDER BY validstart',
        _table_name, _condition1, _table_name, _condition2
    );
END;
$$ LANGUAGE plpgsql STABLE;
```

Temporal Set Difference

```
CREATE OR REPLACE FUNCTION temporal_set_difference(
    _table_name TEXT,
    _id_column TEXT,
    _condition_include TEXT,
    _condition_exclude TEXT
)
RETURNS TABLE (
    data JSONB,
    ValidStart TIMESTAMP,
    ValidEnd TIMESTAMP
)
AS $$
BEGIN
    RETURN QUERY EXECUTE format(
        'SELECT row_to_json(t1.*)::jsonb as data, t1.validstart, t1.validend
         FROM %I t1
         WHERE %s
         AND NOT EXISTS (
             SELECT 1 FROM %I t2
             WHERE t2.%I = t1.%I
             AND %s
             AND t2.validstart < t1.validstart
         )
         ORDER BY t1.validstart',
        _table_name, _condition_include,
        _table_name, _id_column, _id_column,
        _condition_exclude
    );
END;
$$ LANGUAGE plpgsql STABLE;
```

Temporal Join

```
CREATE OR REPLACE FUNCTION temporal_join(
    _table1 TEXT,
    _table2 TEXT,
    _join_column TEXT,
    _columns_t1 TEXT[],
    _columns_t2 TEXT[]
)
RETURNS TABLE (
    data JSONB,
    ValidStart_T1 TIMESTAMP,
    ValidEnd_T1 TIMESTAMP,
    ValidStart_T2 TIMESTAMP,
    ValidEnd_T2 TIMESTAMP
)
AS $$
DECLARE
    _select_t1 TEXT;
    _select_t2 TEXT;
BEGIN
    -- Build column selections for table 1
    _select_t1 := (
        SELECT string_agg(format('''%s'', t1.%I', col, col), ', ')
        FROM unnest(_columns_t1) col
    );

    -- Build column selections for table 2
    _select_t2 := (
        SELECT string_agg(format('''%s'', t2.%I', col, col), ', ')
        FROM unnest(_columns_t2) col
    );

    RETURN QUERY EXECUTE format(
        'SELECT
            jsonb_build_object(%s, %s) as data,
            t1.validstart as ValidStart_T1,
            t1.validend as ValidEnd_T1,
            t2.validstart as ValidStart_T2,
            t2.validend as ValidEnd_T2
         FROM %I t1
         INNER JOIN %I t2 ON t1.%I = t2.%I
         WHERE temporal_intersects(t1.validstart, t1.validend,
t2.validstart, t2.validend)
         ORDER BY t1.validstart, t2.validstart',
        _select_t1, _select_t2,
        _table1, _table2, _join_column, _join_column
    );
END;
$$ LANGUAGE plpgsql STABLE;
```

Temporal Timeslice

```
CREATE OR REPLACE FUNCTION temporal_timeslice(
    _table_name TEXT, _timeslice TIMESTAMP
)
RETURNS TABLE (
    data JSONB,
    validstart TIMESTAMP,
    validend TIMESTAMP
)
AS $$
BEGIN
    RETURN QUERY EXECUTE format(
        'SELECT row_to_json(t.*)::jsonb as data, t.validstart, t.validend
         FROM %I t
         WHERE t.validstart <= %L::TIMESTAMP
         AND (t.validend IS NULL OR t.validend > %L::TIMESTAMP)
         ORDER BY t.validstart',
        _table_name, _timeslice, _timeslice
    );
END;
$$ LANGUAGE plpgsql STABLE;

CREATE OR REPLACE FUNCTION temporal_timeslice(
    _table_name TEXT,
    _timeslice TIMESTAMPTZ
)
RETURNS TABLE (
    data JSONB,
    validstart TIMESTAMP,
    validend TIMESTAMP
)
AS $$
BEGIN
    -- Cast to TIMESTAMP and call the main function
    RETURN QUERY
    SELECT * FROM temporal_timeslice(_table_name, _timeslice::TIMESTAMP);
END;
$$ LANGUAGE plpgsql STABLE;
```

4. *Coalesce*

Fungsi Generic Coalesce untuk Tabel Apapun yang Memiliki ValidStart/ValidEnd

```
CREATE OR REPLACE FUNCTION temporal_coalesce(
    _table_name TEXT,
    _group_columns TEXT[],
    _where_clause TEXT DEFAULT NULL
)
RETURNS TABLE (
    data JSONB,
    validstart TIMESTAMP,
    validend TIMESTAMP,
    coalesced_count BIGINT
)
AS $$
DECLARE
    _group_cols_select TEXT;
    _group_cols_partition TEXT;
    _group_cols_json TEXT;
    _where_sql TEXT;
BEGIN
    -- validasi
    IF _table_name IS NULL OR _table_name = '' THEN
        RAISE EXCEPTION '_table_name tidak boleh kosong';
    END IF;
    IF _group_columns IS NULL OR array_length(_group_columns, 1) IS NULL
THEN
        RAISE EXCEPTION '_group_columns tidak boleh kosong';
    END IF;

    -- build dynamic SQL parts
    _group_cols_select := (
        SELECT string_agg(format('%I', col), ', ') FROM
unnest(_group_columns) col
    );
    _group_cols_partition := _group_cols_select;
    _group_cols_json := (
        SELECT string_agg(format('''%s'', %I', col, col), ', ') FROM
unnest(_group_columns) col
    );

    IF _where_clause IS NOT NULL AND _where_clause <> '' THEN
        _where_sql := format('WHERE %s', _where_clause);
    ELSE
        _where_sql := '';
    END IF;

    RETURN QUERY EXECUTE format(
        'WITH ordered_data AS (
            SELECT
                %s,
                validstart,
                validend,
                COALESCE(validend, ''infinity''::TIMESTAMP) as
validend_normalized,
                -- cek adjacent: LAG(validend) >= validstart berarti
lanjutan grup
                CASE
                    WHEN LAG(COALESCE(validend, ''infinity''::TIMESTAMP))
OVER (
                        PARTITION BY %s ORDER BY validstart
                    ) >= validstart THEN 0
                    ELSE 1
                END as is_new_group
            FROM %I %s
        ),
        grouped_data AS (
            SELECT *, SUM(is_new_group) OVER (
                PARTITION BY %s ORDER BY validstart ROWS UNBOUNDED PRECEDING
            ) as group_num
            FROM ordered_data
        ),
        coalesced AS (
            SELECT
                %s,
                MIN(validstart) as validstart,
                CASE WHEN MAX(validend_normalized) = ''infinity''::TIMESTAMP
                     THEN NULL ELSE MAX(validend_normalized) END as
validend,
                COUNT(*) as coalesced_count
            FROM grouped_data
            GROUP BY %s, group_num
        )
        SELECT jsonb_build_object(%s), validstart, validend, coalesced_count
        FROM coalesced ORDER BY validstart',
        _group_cols_select, _group_cols_partition, _table_name, _where_sql,
        _group_cols_partition, _group_cols_select, _group_cols_partition,
_group_cols_json
    );
END;
```

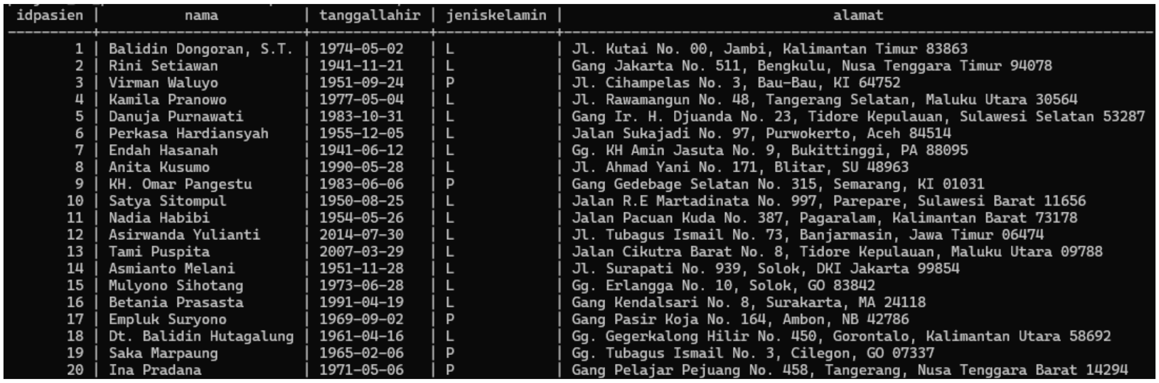
Fungsi Coalesce khusus Tanda Vital dengan tambahan fitur khusus *tracking* ID mana saja yang digabung

```
CREATE OR REPLACE FUNCTION coalesce_tandavital(
    _idrawatinap INT,
    _vital_columns TEXT[]
)
RETURNS TABLE (
    idrawatinap INT,
    vital_data JSONB,
    validstart TIMESTAMP,
    validend TIMESTAMP,
    coalesced_count BIGINT,
    merged_ids INT[]
)
AS $$
DECLARE
    _vital_cols_select TEXT;
    _vital_cols_partition TEXT;
    _vital_cols_json TEXT;
    _where_clause TEXT;
BEGIN
    _vital_cols_select := (
        SELECT string_agg(format('%I', col), ', ') FROM
unnest(_vital_columns) col
    );
    _vital_cols_partition := _vital_cols_select;
    _vital_cols_json := (
        SELECT string_agg(format('''%s'', %I', col, col), ', ') FROM
unnest(_vital_columns) col
    );

    IF _idrawatinap IS NOT NULL THEN
        _where_clause := format('WHERE idrawatinap = %L', _idrawatinap);
    ELSE
        _where_clause := '';
    END IF;

    RETURN QUERY EXECUTE format(
        'WITH ordered_data AS (
            SELECT
                t.idrawatinap, t.idvital, %s, t.validstart, t.validend,
                COALESCE(t.validend, ''infinity''::TIMESTAMP) as
validend_norm,
                CASE
                    WHEN LAG(COALESCE(t.validend, ''infinity''::TIMESTAMP))
OVER (
                        PARTITION BY t.idrawatinap, %s ORDER BY t.validstart
                    ) >= t.validstart THEN 0
                    ELSE 1
                END as is_new_group
            FROM tandavital t %s
        ),
        grouped_data AS (
            SELECT *, SUM(is_new_group) OVER (
                PARTITION BY idrawatinap, %s ORDER BY validstart ROWS
UNBOUNDED PRECEDING
            ) as group_num
            FROM ordered_data
        )
        SELECT
            idrawatinap,
            jsonb_build_object(%s) as vital_data,
            MIN(validstart)::TIMESTAMP,
            CASE WHEN MAX(validend_norm) = ''infinity''::TIMESTAMP
                 THEN NULL ELSE MAX(validend_norm) END::TIMESTAMP,
            COUNT(*)::BIGINT,
            array_agg(idvital ORDER BY validstart)::INT[]
        FROM grouped_data
        GROUP BY idrawatinap, %s, group_num
        ORDER BY idrawatinap, MIN(validstart)',
        _vital_cols_select, _vital_cols_partition, _where_clause,
        _vital_cols_partition, _vital_cols_json, _vital_cols_partition
    );
END;
```

d. Sampel data

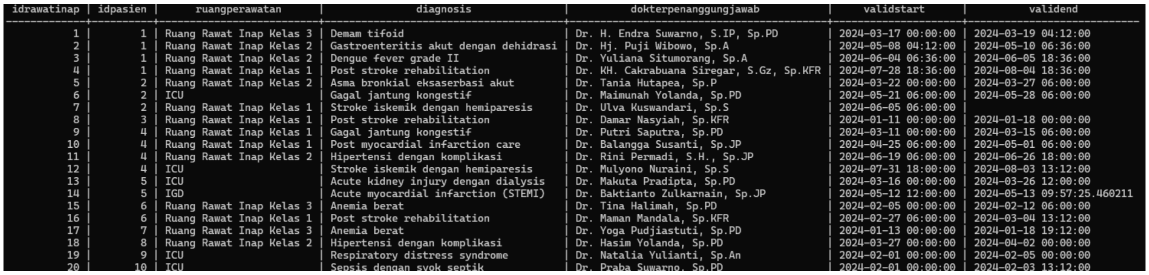
Sampel data masing-masing tabel akan di-*limit* sejumlah dua puluh data.

1. Tabel Pasien



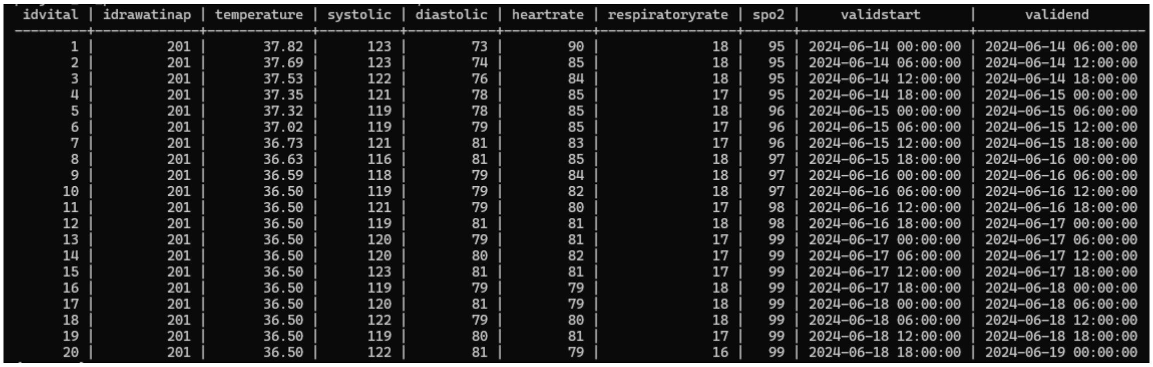
```
 idpasien |           nama           | tanggallahir | jeniskelamin |                            alamat
----------+--------------------------+--------------+--------------+-------------------------------------------------------------------
        1 | Balidin Dongoran, S.T.   | 1974-05-02   | L            | Jl. Kutai No. 00, Jambi, Kalimantan Timur 83863
        2 | Rini Setiawan            | 1941-11-21   | L            | Gang Jakarta No. 511, Bengkulu, Nusa Tenggara Timur 94078
        3 | Virman Waluyo            | 1951-09-24   | P            | Jl. Cihampelas No. 3, Bau-Bau, KI 64752
        4 | Kamila Pranowo           | 1977-05-04   | L            | Jl. Rawamangun No. 48, Tangerang Selatan, Maluku Utara 30564
        5 | Danuja Purnawati         | 1983-10-31   | L            | Gang Ir. H. Djuanda No. 23, Tidore Kepulauan, Sulawesi Selatan 53287
        6 | Perkasa Hardiansyah      | 1955-12-05   | L            | Jalan Sukajadi No. 97, Purwokerto, Aceh 84514
        7 | Endah Hasanah            | 1941-06-12   | L            | Gg. KH Amin Jasuta No. 9, Bukittinggi, PA 88095
        8 | Anita Kusumo             | 1990-05-28   | L            | Jl. Ahmad Yani No. 171, Blitar, SU 48963
        9 | KH. Omar Pangestu        | 1983-06-06   | P            | Gang Gedebage Selatan No. 315, Semarang, KI 01031
       10 | Satya Sitompul           | 1950-08-25   | L            | Jalan R.E Martadinata No. 997, Parepare, Sulawesi Barat 11656
       11 | Nadia Habibi             | 1954-05-26   | L            | Jalan Pacuan Kuda No. 387, Pagaralam, Kalimantan Barat 73178
       12 | Asirwanda Yulianti       | 2014-07-30   | L            | Jl. Tubagus Ismail No. 73, Banjarmasin, Jawa Timur 06474
       13 | Tami Puspita             | 2007-03-29   | L            | Jalan Cikutra Barat No. 8, Tidore Kepulauan, Maluku Utara 09788
       14 | Asmianto Melani          | 1951-11-28   | L            | Jl. Surapati No. 939, Solok, DKI Jakarta 99854
       15 | Mulyono Sihotang         | 1973-06-28   | L            | Gg. Erlangga No. 10, Solok, GO 83842
       16 | Betania Prasasta         | 1991-04-19   | L            | Gang Kendalsari No. 8, Surakarta, MA 24118
       17 | Empluk Suryono           | 1969-09-02   | P            | Gang Pasir Koja No. 164, Ambon, NB 42786
       18 | Dt. Balidin Hutagalung   | 1961-04-16   | L            | Gg. Gegerkalong Hilir No. 450, Gorontalo, Kalimantan Utara 58692
       19 | Saka Marpaung            | 1965-02-06   | P            | Gg. Tubagus Ismail No. 3, Cilegon, GO 07337
       20 | Ina Pradana              | 1971-05-06   | P            | Gang Pelajar Pejuang No. 458, Tangerang, Nusa Tenggara Barat 14294
```

2. Tabel RawatInap



```
 idrawatinap | idpasien |      ruangperawatan       |                diagnosis                 |          dokterpenanggungjawab          |     validstart      |          validend
-------------+----------+---------------------------+------------------------------------------+-----------------------------------------+---------------------+----------------------------
           1 |        1 | Ruang Rawat Inap Kelas 3  | Demam tifoid                             | Dr. H. Endra Suwarno, S.IP, Sp.PD       | 2024-03-17 00:00:00 | 2024-03-19 04:12:00
           2 |        1 | Ruang Rawat Inap Kelas 2  | Gastroenteritis akut dengan dehidrasi    | Dr. Hj. Puji Wibowo, Sp.A               | 2024-05-08 04:12:00 | 2024-05-10 06:36:00
           3 |        1 | Ruang Rawat Inap Kelas 2  | Dengue fever grade II                    | Dr. Yuliana Situmorang, Sp.A            | 2024-06-04 06:36:00 | 2024-06-05 18:36:00
           4 |        1 | Ruang Rawat Inap Kelas 1  | Post stroke rehabilitation               | Dr. KH. Cakrabuana Siregar, S.Gz, Sp.KFR | 2024-07-28 18:36:00 | 2024-08-04 18:36:00
           5 |        2 | Ruang Rawat Inap Kelas 2  | Asma bronkial eksaserbasi akut           | Dr. Tania Hutapea, Sp.P                 | 2024-03-22 00:00:00 | 2024-03-27 06:00:00
           6 |        2 | ICU                       | Gagal jantung kongestif                  | Dr. Maimunah Yolanda, Sp.PD             | 2024-05-21 06:00:00 | 2024-05-28 06:00:00
           7 |        2 | Ruang Rawat Inap Kelas 1  | Stroke iskemik dengan hemiparesis        | Dr. Ulva Kuswandari, Sp.S               | 2024-06-05 06:00:00 |
           8 |        3 | Ruang Rawat Inap Kelas 1  | Post stroke rehabilitation               | Dr. Damar Nasyiah, Sp.KFR               | 2024-01-11 00:00:00 | 2024-01-18 00:00:00
           9 |        4 | Ruang Rawat Inap Kelas 1  | Gagal jantung kongestif                  | Dr. Putri Saputra, Sp.PD                | 2024-03-11 00:00:00 | 2024-03-15 06:00:00
          10 |        4 | Ruang Rawat Inap Kelas 1  | Post myocardial infarction care          | Dr. Balangga Susanti, Sp.JP             | 2024-04-25 06:00:00 | 2024-05-01 06:00:00
          11 |        4 | Ruang Rawat Inap Kelas 2  | Hipertensi dengan komplikasi             | Dr. Rini Permadi, S.H., Sp.JP           | 2024-06-19 06:00:00 | 2024-06-26 18:00:00
          12 |        4 | ICU                       | Stroke iskemik dengan hemiparesis        | Dr. Mulyono Nuraini, Sp.S               | 2024-07-31 18:00:00 | 2024-08-03 13:12:00
          13 |        5 | ICU                       | Acute kidney injury dengan dialysis      | Dr. Makuta Pradipta, Sp.PD              | 2024-03-16 00:00:00 | 2024-03-26 12:00:00
          14 |        5 | IGD                       | Acute myocardial infarction (STEMI)      | Dr. Baktianto Zulkarnain, Sp.JP         | 2024-05-12 12:00:00 | 2024-05-13 09:57:25.460211
          15 |        6 | Ruang Rawat Inap Kelas 3  | Anemia berat                             | Dr. Tina Halimah, Sp.PD                 | 2024-02-05 00:00:00 | 2024-02-12 06:00:00
          16 |        6 | Ruang Rawat Inap Kelas 1  | Post stroke rehabilitation               | Dr. Maman Mandala, Sp.KFR               | 2024-02-27 06:00:00 | 2024-03-04 13:12:00
          17 |        7 | Ruang Rawat Inap Kelas 3  | Anemia berat                             | Dr. Yoga Pudjiastuti, Sp.PD             | 2024-01-13 00:00:00 | 2024-01-18 19:12:00
          18 |        8 | Ruang Rawat Inap Kelas 2  | Hipertensi dengan komplikasi             | Dr. Hasim Yolanda, Sp.PD                | 2024-03-27 00:00:00 | 2024-04-02 00:00:00
          19 |        9 | ICU                       | Respiratory distress syndrome            | Dr. Natalia Yulianti, Sp.An             | 2024-02-01 00:00:00 | 2024-02-05 00:00:00
          20 |       10 | ICU                       | Sepsis dengan syok septik                | Dr. Praba Suwarno, Sp.PD                | 2024-02-01 00:00:00 | 2024-02-03 13:12:00
```

3. Tabel TandaVital



```
 idvital | idrawatinap | temperature | systolic | diastolic | heartrate | respiratoryrate | spo2 |     validstart      |      validend
---------+-------------+-------------+----------+-----------+-----------+-----------------+------+---------------------+---------------------
       1 |         201 |       37.82 |      123 |        73 |        90 |              18 |   95 | 2024-06-14 00:00:00 | 2024-06-14 06:00:00
       2 |         201 |       37.69 |      123 |        74 |        85 |              18 |   95 | 2024-06-14 06:00:00 | 2024-06-14 12:00:00
       3 |         201 |       37.53 |      122 |        76 |        84 |              18 |   95 | 2024-06-14 12:00:00 | 2024-06-14 18:00:00
       4 |         201 |       37.35 |      121 |        78 |        85 |              17 |   95 | 2024-06-14 18:00:00 | 2024-06-15 00:00:00
       5 |         201 |       37.32 |      119 |        78 |        85 |              18 |   96 | 2024-06-15 00:00:00 | 2024-06-15 06:00:00
       6 |         201 |       37.02 |      119 |        79 |        85 |              17 |   96 | 2024-06-15 06:00:00 | 2024-06-15 12:00:00
       7 |         201 |       36.73 |      121 |        81 |        83 |              17 |   96 | 2024-06-15 12:00:00 | 2024-06-15 18:00:00
       8 |         201 |       36.63 |      116 |        81 |        85 |              18 |   97 | 2024-06-15 18:00:00 | 2024-06-16 00:00:00
       9 |         201 |       36.59 |      118 |        79 |        84 |              18 |   97 | 2024-06-16 00:00:00 | 2024-06-16 06:00:00
      10 |         201 |       36.50 |      119 |        79 |        82 |              18 |   97 | 2024-06-16 06:00:00 | 2024-06-16 12:00:00
      11 |         201 |       36.50 |      121 |        79 |        80 |              17 |   98 | 2024-06-16 12:00:00 | 2024-06-16 18:00:00
      12 |         201 |       36.50 |      119 |        81 |        81 |              18 |   98 | 2024-06-16 18:00:00 | 2024-06-17 00:00:00
      13 |         201 |       36.50 |      120 |        79 |        81 |              17 |   99 | 2024-06-17 00:00:00 | 2024-06-17 06:00:00
      14 |         201 |       36.50 |      120 |        80 |        82 |              17 |   99 | 2024-06-17 06:00:00 | 2024-06-17 12:00:00
      15 |         201 |       36.50 |      123 |        81 |        81 |              17 |   99 | 2024-06-17 12:00:00 | 2024-06-17 18:00:00
      16 |         201 |       36.50 |      119 |        79 |        79 |              18 |   99 | 2024-06-17 18:00:00 | 2024-06-18 00:00:00
      17 |         201 |       36.50 |      120 |        81 |        79 |              18 |   99 | 2024-06-18 00:00:00 | 2024-06-18 06:00:00
      18 |         201 |       36.50 |      122 |        79 |        80 |              18 |   99 | 2024-06-18 06:00:00 | 2024-06-18 12:00:00
      19 |         201 |       36.50 |      119 |        80 |        81 |              17 |   99 | 2024-06-18 12:00:00 | 2024-06-18 18:00:00
      20 |         201 |       36.50 |      122 |        81 |        79 |              16 |   99 | 2024-06-18 18:00:00 | 2024-06-19 00:00:00
```

3. IMPLEMENTASI QUERY

a. DDL *statement*

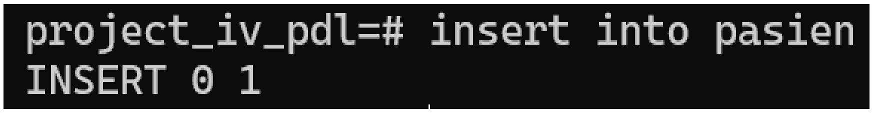
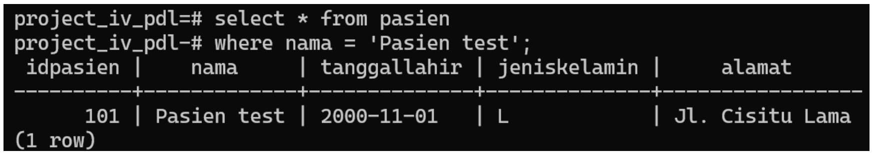
<table>
<tr><td>Deskripsi</td><td>Membuat tabel Pasien</td></tr>
<tr><td>Query</td><td><pre>
CREATE TABLE Pasien (
    IDPasien SERIAL PRIMARY KEY,
    Nama VARCHAR(250) NOT NULL,
    TanggalLahir DATE NOT NULL,
    JenisKelamin CHAR(1) NOT NULL
      CHECK (JenisKelamin IN ('L','P')),
    Alamat VARCHAR(250) NOT NULL
);
</pre></td></tr>
</table>

<table>
<tr><td>Deskripsi</td><td>Membuat tabel RawatInap</td></tr>
<tr><td>Query</td><td><pre>
CREATE TABLE RawatInap (
    IDRawatInap SERIAL PRIMARY KEY,
    IDPasien INT NOT NULL REFERENCES Pasien(IDPasien),
    RuangPerawatan VARCHAR(100) NOT NULL,
    Diagnosis VARCHAR(500),
    DokterPenanggungjawab VARCHAR(250),
    ValidStart TIMESTAMP NOT NULL,
    ValidEnd TIMESTAMP,
    CONSTRAINT chk_rawatinap_time
      CHECK (ValidEnd IS NULL OR ValidEnd > ValidStart)
);
</pre></td></tr>
</table>

<table>
<tr><td>Deskripsi</td><td>Membuat tabel TandaVital</td></tr>
<tr><td>Query</td><td><pre>
CREATE TABLE TandaVital(
    IDVital SERIAL PRIMARY KEY,
    IDRawatInap INT REFERENCES RawatInap(IDRawatInap),
    Temperature DECIMAL(4,2) NOT NULL,
    Systolic INT NOT NULL,
    Diastolic INT NOT NULL,
    HeartRate INT NOT NULL,
    RespiratoryRate INT NOT NULL,
    SPO2 INT NOT NULL,
    ValidStart TIMESTAMP NOT NULL,
    ValidEnd TIMESTAMP,
    CONSTRAINT chk_vital_time
      CHECK (ValidEnd IS NULL OR ValidEnd > ValidStart)
);
</pre></td></tr>
</table>

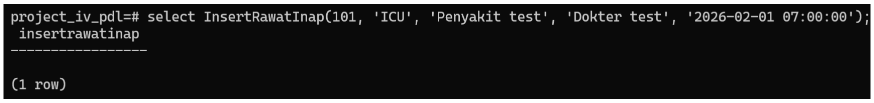
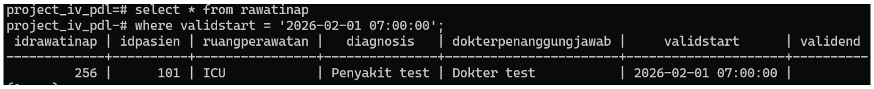
b. SQL *statement*

1. *Insert*, *Update*, dan *Delete*

Deskripsi	Menambah pasien baru
Query	`insert into pasien (nama, tanggallahir, jeniskelamin, alamat) values ('Pasien test', '2000-11-01', 'L', 'Jl. Cisitu Lama');`
Hasil	 

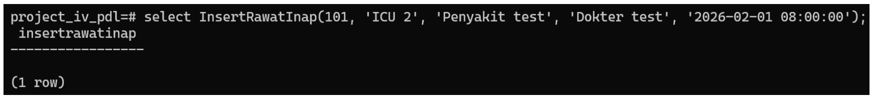
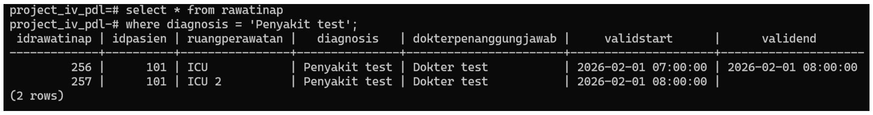
```
project_iv_pdl=# insert into pasien
INSERT 0 1
```

```
project_iv_pdl=# select * from pasien
project_iv_pdl-# where nama = 'Pasien test';
 idpasien |    nama     | tanggallahir | jeniskelamin |     alamat
----------+-------------+--------------+--------------+-----------------
      101 | Pasien test | 2000-11-01   | L            | Jl. Cisitu Lama
(1 row)
```

Deskripsi	Insert data RawatInap untuk pasien tersebut
Keterangan tambahan	Pasien masuk rawat inap
Query	`select InsertRawatInap(101, 'ICU', 'Penyakit test', 'Dokter test', '2026-02-01 07:00:00');`
Hasil	 

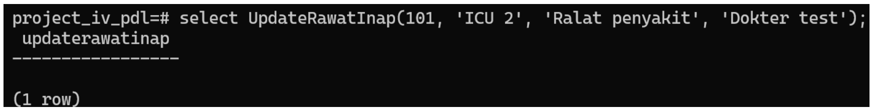
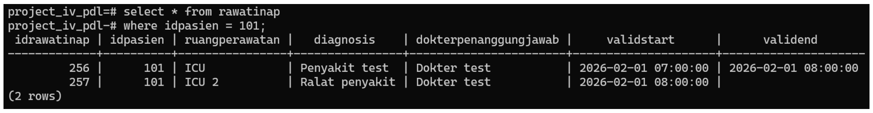
```
project_iv_pdl=# select InsertRawatInap(101, 'ICU', 'Penyakit test', 'Dokter test', '2026-02-01 07:00:00');
 insertrawatinap
-----------------

(1 row)
```

```
project_iv_pdl=# select * from rawatinap
project_iv_pdl-# where validstart = '2026-02-01 07:00:00';
 idrawatinap | idpasien | ruangperawatan |   diagnosis   | dokterpenanggungjawab |     validstart      | validend
-------------+----------+----------------+---------------+-----------------------+---------------------+----------
         256 |      101 | ICU            | Penyakit test | Dokter test           | 2026-02-01 07:00:00 |
```

Deskripsi	Insert data RawatInap yang berbeda untuk pasien tersebut
Keterangan tambahan	Karena ada pergantian fakta, misalnya pasien pindah ruangan, sehingga perlu insert data baru
Query	`select InsertRawatInap(101, 'ICU 2', 'Penyakit test', 'Dokter test', '2026-02-01 08:00:00');`
Hasil	 

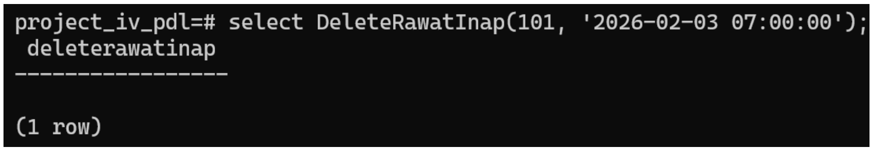
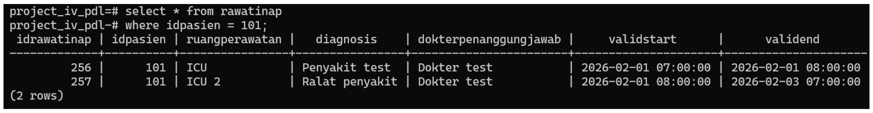
```
project_iv_pdl=# select InsertRawatInap(101, 'ICU 2', 'Penyakit test', 'Dokter test', '2026-02-01 08:00:00');
 insertrawatinap
-----------------

(1 row)
```

```
project_iv_pdl=# select * from rawatinap
project_iv_pdl-# where diagnosis = 'Penyakit test';
 idrawatinap | idpasien | ruangperawatan |   diagnosis   | dokterpenanggungjawab |     validstart      |      validend
-------------+----------+----------------+---------------+-----------------------+---------------------+---------------------
         256 |      101 | ICU            | Penyakit test | Dokter test           | 2026-02-01 07:00:00 | 2026-02-01 08:00:00
         257 |      101 | ICU 2          | Penyakit test | Dokter test           | 2026-02-01 08:00:00 |
(2 rows)
```

Deskripsi	Update data RawatInap untuk pasien tersebut
Keterangan tambahan	Karena terdapat kesalahan *input*, misalnya pasien diagnosis yang di-*input* salah
Query	`select UpdateRawatInap(101, 'ICU 2', 'Ralat penyakit', 'Dokter test');`
Hasil	 

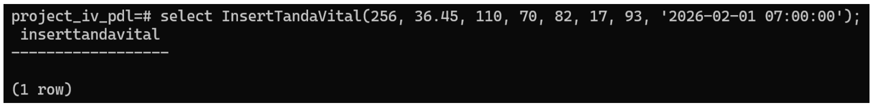
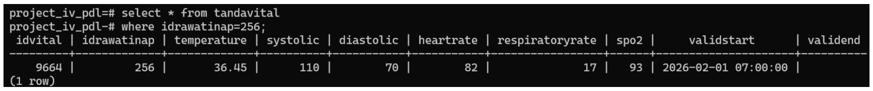
```
project_iv_pdl=# select UpdateRawatInap(101, 'ICU 2', 'Ralat penyakit', 'Dokter test');
 updaterawatinap
-----------------

(1 row)
```

```
project_iv_pdl=# select * from rawatinap
project_iv_pdl-# where idpasien = 101;
 idrawatinap | idpasien | ruangperawatan |   diagnosis    | dokterpenanggungjawab |     validstart      |      validend
-------------+----------+----------------+----------------+-----------------------+---------------------+---------------------
         256 |      101 | ICU            | Penyakit test  | Dokter test           | 2026-02-01 07:00:00 | 2026-02-01 08:00:00
         257 |      101 | ICU 2          | Ralat penyakit | Dokter test           | 2026-02-01 08:00:00 |
(2 rows)
```

Deskripsi	Delete data RawatInap pasien tersebut
Keterangan tambahan	Pasien telah sembuh dan selesai dirawat inap
Query	`select DeleteRawatInap(101, '2026-02-03 07:00:00');`
Hasil	 

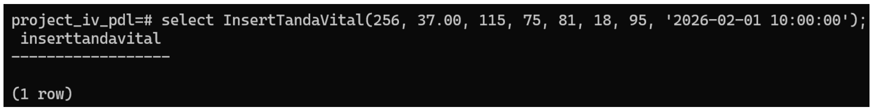
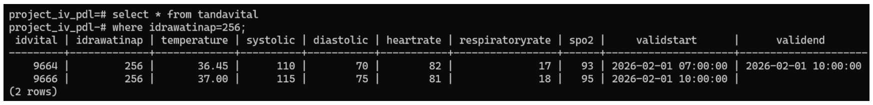
```
project_iv_pdl=# select DeleteRawatInap(101, '2026-02-03 07:00:00');
 deleterawatinap
-----------------

(1 row)
```

```
project_iv_pdl=# select * from rawatinap
project_iv_pdl-# where idpasien = 101;
 idrawatinap | idpasien | ruangperawatan |   diagnosis    | dokterpenanggungjawab |     validstart      |      validend
-------------+----------+----------------+----------------+-----------------------+---------------------+---------------------
         256 |      101 | ICU            | Penyakit test  | Dokter test           | 2026-02-01 07:00:00 | 2026-02-01 08:00:00
         257 |      101 | ICU 2          | Ralat penyakit | Dokter test           | 2026-02-01 08:00:00 | 2026-02-03 07:00:00
(2 rows)
```

Deskripsi	Insert data TandaVital untuk pasien tersebut
Keterangan tambahan	Pasien cek kesehatan setelah masuk rawat inap
Query	`select InsertTandaVital(256, 36.45, 110, 70, 82, 17, 93, '2026-02-01 07:00:00');`
Hasil	 

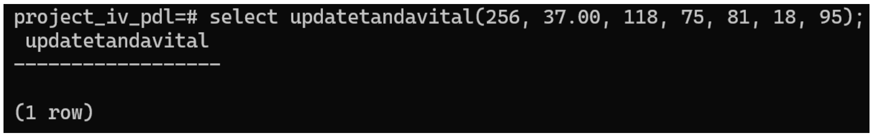
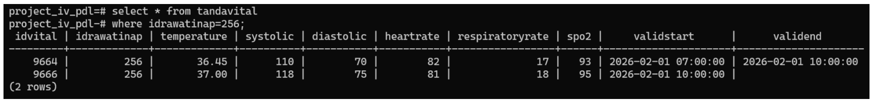
```
project_iv_pdl=# select InsertTandaVital(256, 36.45, 110, 70, 82, 17, 93, '2026-02-01 07:00:00');
 inserttandavital
------------------

(1 row)
```

```
project_iv_pdl=# select * from tandavital
project_iv_pdl-# where idrawatinap=256;
 idvital | idrawatinap | temperature | systolic | diastolic | heartrate | respiratoryrate | spo2 |     validstart      | validend
---------+-------------+-------------+----------+-----------+-----------+-----------------+------+---------------------+----------
    9664 |         256 |       36.45 |      110 |        70 |        82 |              17 |   93 | 2026-02-01 07:00:00 |
(1 row)
```

Deskripsi	Insert data TandaVital baru untuk pasien tersebut
Keterangan tambahan	Pasien kembali dicek tanda vitalnya tiga jam kemudian (asumsi pasien belum pindah sehingga id rawat inap masih 256)
Query	`select InsertTandaVital(256, 37.00, 115, 75, 81, 18, 95, '2026-02-01 10:00:00');`
Hasil	 

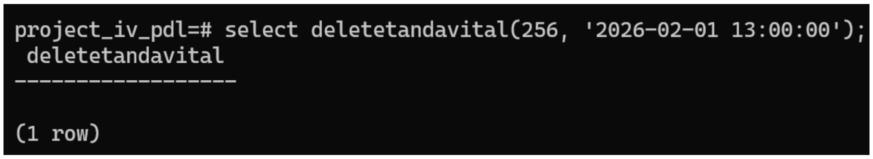
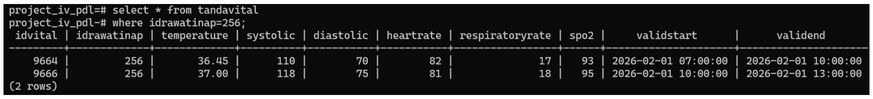
```
project_iv_pdl=# select InsertTandaVital(256, 37.00, 115, 75, 81, 18, 95, '2026-02-01 10:00:00');
 inserttandavital
------------------

(1 row)
```

```
project_iv_pdl=# select * from tandavital
project_iv_pdl-# where idrawatinap=256;
 idvital | idrawatinap | temperature | systolic | diastolic | heartrate | respiratoryrate | spo2 |     validstart      |      validend
---------+-------------+-------------+----------+-----------+-----------+-----------------+------+---------------------+---------------------
    9664 |         256 |       36.45 |      110 |        70 |        82 |              17 |   93 | 2026-02-01 07:00:00 | 2026-02-01 10:00:00
    9666 |         256 |       37.00 |      115 |        75 |        81 |              18 |   95 | 2026-02-01 10:00:00 |
(2 rows)
```

Deskripsi	Update data TandaVital untuk pasien tersebut
Keterangan tambahan	Misalnya saat pengecekkan pukul 10, terdapat kesalahan input sistolik seharusnya 118
Query	`select updatetandavital(256, 37.00, 118, 75, 81, 18, 95);`
Hasil	 

```
project_iv_pdl=# select updatetandavital(256, 37.00, 118, 75, 81, 18, 95);
 updatetandavital
------------------

(1 row)
```

```
project_iv_pdl=# select * from tandavital
project_iv_pdl-# where idrawatinap=256;
 idvital | idrawatinap | temperature | systolic | diastolic | heartrate | respiratoryrate | spo2 |     validstart      |      validend
---------+-------------+-------------+----------+-----------+-----------+-----------------+------+---------------------+---------------------
    9664 |         256 |       36.45 |      110 |        70 |        82 |              17 |   93 | 2026-02-01 07:00:00 | 2026-02-01 10:00:00
    9666 |         256 |       37.00 |      118 |        75 |        81 |              18 |   95 | 2026-02-01 10:00:00 |
(2 rows)
```

Deskripsi	Delete data tanda vital karena pengecekkan sudah tidak dilakukkan
Query	`select deletetandavital(256, '2026-02-01 13:00:00');`
Hasil	 

```
project_iv_pdl=# select deletetandavital(256, '2026-02-01 13:00:00');
 deletetandavital
------------------

(1 row)
```

```
project_iv_pdl=# select * from tandavital
project_iv_pdl-# where idrawatinap=256;
 idvital | idrawatinap | temperature | systolic | diastolic | heartrate | respiratoryrate | spo2 |     validstart      |      validend
---------+-------------+-------------+----------+-----------+-----------+-----------------+------+---------------------+---------------------
    9664 |         256 |       36.45 |      110 |        70 |        82 |              17 |   93 | 2026-02-01 07:00:00 | 2026-02-01 10:00:00
    9666 |         256 |       37.00 |      118 |        75 |        81 |              18 |   95 | 2026-02-01 10:00:00 | 2026-02-01 13:00:00
(2 rows)
```

2. *Allen's* 13 *Relationships*

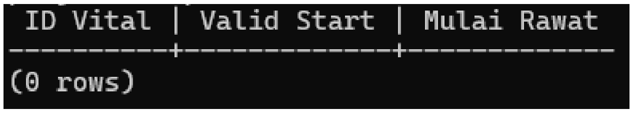
Deskripsi

Mencari data Tanda Vital yang dimasukkan sebelum masa Rawat Inap dimulai (data tidak valid).
BEFORE (X < Y)

Query

```
SELECT tv.IDVital "ID Vital",
       tv.ValidStart AS "Valid Start",
       r.ValidStart AS "Mulai Rawat"
FROM TandaVital tv
JOIN RawatInap r ON tv.IDRawatInap = r.IDRawatInap
WHERE allen_before(tv.ValidStart, tv.ValidEnd,
                   r.ValidStart, r.ValidEnd);
```

Hasil



```
 ID Vital | Valid Start | Mulai Rawat
----------+-------------+-------------
(0 rows)
```

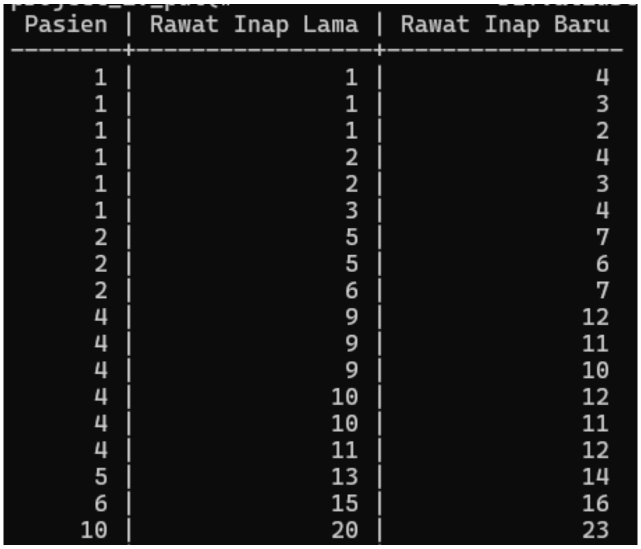
Deskripsi

Mencari jadwal Rawat Inap baru yang terjadi setelah rawat inap sebelumnya selesai.
AFTER (Y > X)

Query

```
SELECT r1.IDPasien as "Pasien",
       r1.IDRawatInap AS "Rawat Inap Lama",
       r2.IDRawatInap AS "Rawat Inap Baru"
FROM RawatInap r1
JOIN RawatInap r2 ON r1.IDPasien = r2.IDPasien
WHERE r1.IDRawatInap != r2.IDRawatInap
  AND allen_after(r2.ValidStart, r2.ValidEnd,
                  r1.ValidStart, r1.ValidEnd);
```

Hasil



```
 Pasien | Rawat Inap Lama | Rawat Inap Baru
--------+-----------------+-----------------
      1 |               1 |               4
      1 |               1 |               3
      1 |               1 |               2
      1 |               2 |               4
      1 |               2 |               3
      1 |               3 |               4
      2 |               5 |               7
      2 |               5 |               6
      2 |               6 |               7
      4 |               9 |              12
      4 |               9 |              11
      4 |               9 |              10
      4 |              10 |              12
      4 |              10 |              11
      4 |              11 |              12
      5 |              13 |              14
      6 |              15 |              16
     10 |              20 |              23
```

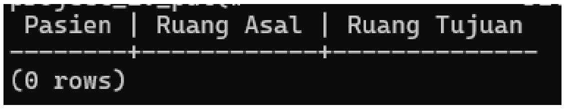
Deskripsi

Mencari rawat inap dengan perpindahan kamar/unit. Rawat Inap awalnya di kamar A lalu dipindahkan ke kamar B.
MEETS (X m Y)

Query

```
SELECT r1.IDPasien AS “Pasien”,
       r1.RuangPerawatan AS “Ruang Asal”,
       r2.RuangPerawatan AS “Ruang Tujuan”
FROM RawatInap r1
JOIN RawatInap r2 ON r1.IDPasien = r2.IDPasien
WHERE allen_meets(r1.ValidStart, r1.ValidEnd,
                  r2.ValidStart, r2.ValidEnd);
```

Hasil



```
 Pasien | Ruang Asal | Ruang Tujuan
--------+------------+--------------
(0 rows)
```

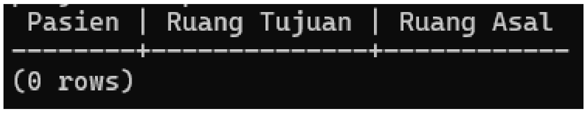
Deskripsi

Kebalikan dari Meets.
MET BY (Y mi X)

Query

```
SELECT r1.IDPasien AS “Pasien”,
       r2.RuangPerawatan AS “Ruang Tujuan”,
       r1.RuangPerawatan AS “Ruang Asal”
FROM RawatInap r1
JOIN RawatInap r2 ON r1.IDPasien = r2.IDPasien
WHERE allen_met_by(r2.ValidStart, r2.ValidEnd,
                   r1.ValidStart, r1.ValidEnd);
```

Hasil



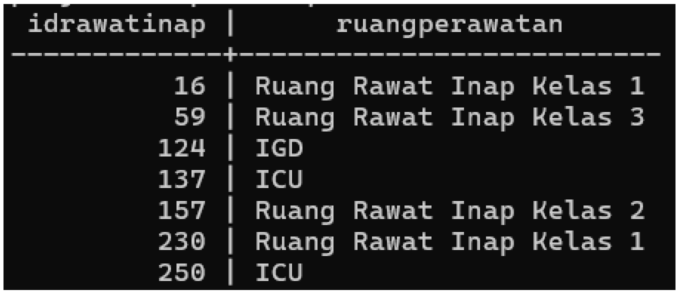
```
 Pasien | Ruang Tujuan | Ruang Asal
--------+--------------+------------
(0 rows)
```

Deskripsi

Mencari jadwal Rawat Inap yang beririsan dengan kejadian tertentu (misal renovasi rumah sakit).
OVERLAPS (X o Y)

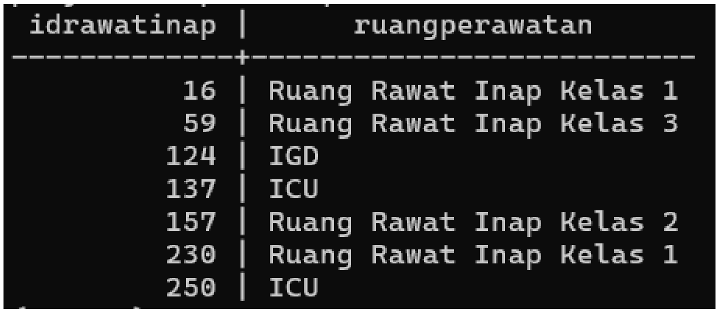
Query

```
SELECT IDRawatInap, RuangPerawatan
FROM RawatInap
WHERE allen_overlaps(
    ValidStart, ValidEnd,
    '2024-03-01 08:00:00', '2024-04-02 08:00:00' -- Jadwal
);
```

Hasil	
	` idrawatinap |      ruangperawatan`
	`-------------+-------------------------`
	`          16 | Ruang Rawat Inap Kelas 1`
	`          59 | Ruang Rawat Inap Kelas 3`
	`         124 | IGD`
	`         137 | ICU`
	`         157 | Ruang Rawat Inap Kelas 2`
	`         230 | Ruang Rawat Inap Kelas 1`
	`         250 | ICU`

Deskripsi	Kebalikan Overlaps. Apakah jadwal tertentu "overlap" dengan jadwal pasien tertentu. OVERLAPPED BY (Y oi X)

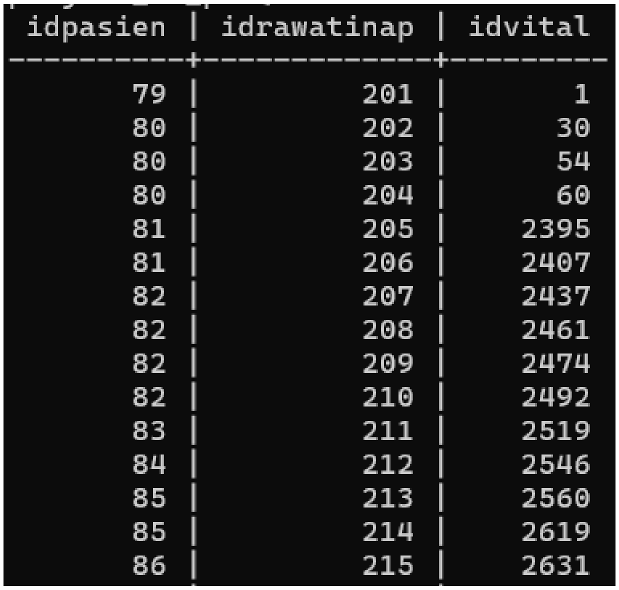
Query	`SELECT IDRawatInap, RuangPerawatan`
	`FROM RawatInap`
	`WHERE allen_overlapped_by(`
	`    '2024-03-01 08:00:00', '2024-04-02 08:00:00', -- Jadwal`
	`    ValidStart, ValidEnd`
	`);`

Hasil	
	` idrawatinap |      ruangperawatan`
	`-------------+-------------------------`
	`          16 | Ruang Rawat Inap Kelas 1`
	`          59 | Ruang Rawat Inap Kelas 3`
	`         124 | IGD`
	`         137 | ICU`
	`         157 | Ruang Rawat Inap Kelas 2`
	`         230 | Ruang Rawat Inap Kelas 1`
	`         250 | ICU`

Deskripsi	Mencari pemeriksaan Tanda Vital yang dilakukan tepat saat pasien baru saja masuk. STARTS (X s Y)

Query	`SELECT r.IDPasien, tv.IDRawatInap, tv.IDVital`
	`FROM TandaVital tv`
	`JOIN RawatInap r ON tv.IDRawatInap = r.IDRawatInap`
	`WHERE allen_starts(tv.ValidStart, tv.ValidEnd,`
	`                   r.ValidStart, r.ValidEnd);`

<table>
<tr><td>Hasil</td><td>



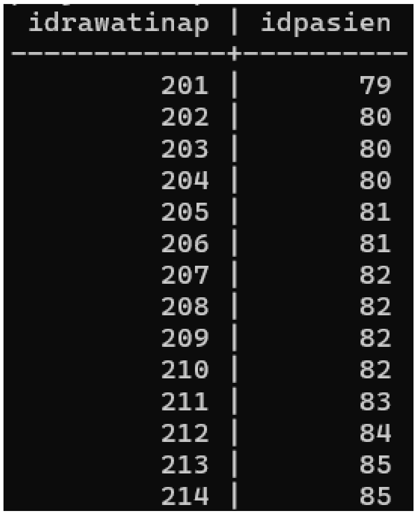
```
 idpasien | idrawatinap | idvital
----------+-------------+---------
       79 |         201 |       1
       80 |         202 |      30
       80 |         203 |      54
       80 |         204 |      60
       81 |         205 |    2395
       81 |         206 |    2407
       82 |         207 |    2437
       82 |         208 |    2461
       82 |         209 |    2474
       82 |         210 |    2492
       83 |         211 |    2519
       84 |         212 |    2546
       85 |         213 |    2560
       85 |         214 |    2619
       86 |         215 |    2631
```

</td></tr>
</table>

<table>
<tr><td>Deskripsi</td><td>Kebalikan Starts. Mencari Rawat Inap yang diawali dengan proses pemeriksaan tertentu.
STARTED BY (Y si X)</td></tr>
<tr><td>Query</td><td>

```
SELECT r.IDRawatInap, r.IDPasien
FROM RawatInap r
JOIN TandaVital tv ON r.IDRawatInap = tv.IDRawatInap
WHERE allen_started_by(r.ValidStart, r.ValidEnd,
                       tv.ValidStart, tv.ValidEnd);
```

</td></tr>
<tr><td>Hasil</td><td>



```
 idrawatinap | idpasien
-------------+----------
         201 |       79
         202 |       80
         203 |       80
         204 |       80
         205 |       81
         206 |       81
         207 |       82
         208 |       82
         209 |       82
         210 |       82
         211 |       83
         212 |       84
         213 |       85
         214 |       85
```

</td></tr>
</table>

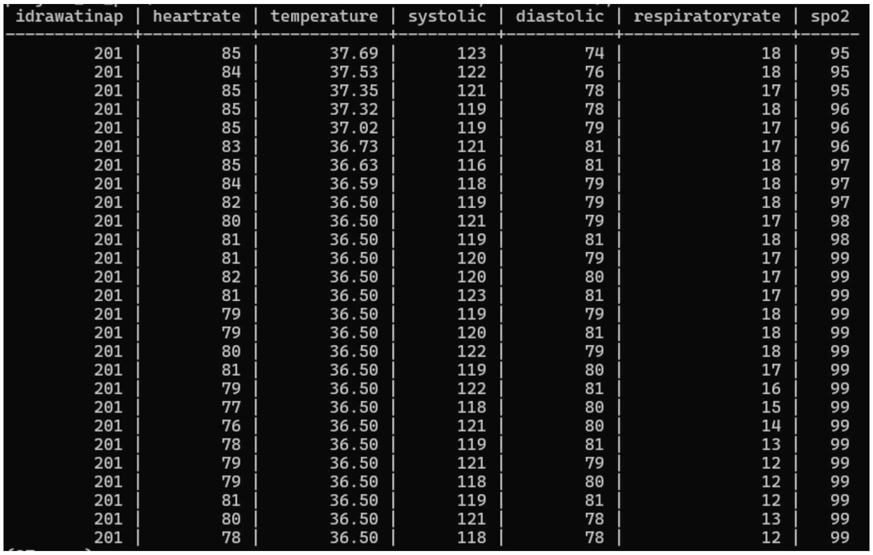
Deskripsi	Mengambil semua data Tanda Vital yang validitasnya berada sepenuhnya di dalam masa Rawat Inap pasien.

DURING (X d Y)

Query

```
SELECT r.IDRawatInap,
       tv.HeartRate,
       tv.Temperature,
       tv.Systolic,
       tv.Diastolic,
       tv.RespiratoryRate,
       tv.SPO2
FROM RawatInap r
JOIN TandaVital tv ON r.IDRawatInap = tv.IDRawatInap
WHERE r.IDPasien = 79
  AND allen_during(tv.ValidStart, tv.ValidEnd,
                   r.ValidStart, r.ValidEnd);
```

Hasil



```
 idrawatinap | heartrate | temperature | systolic | diastolic | respiratoryrate | spo2
-------------+-----------+-------------+----------+-----------+-----------------+------
         201 |        85 |       37.69 |      123 |        74 |              18 |   95
         201 |        84 |       37.53 |      122 |        76 |              18 |   95
         201 |        85 |       37.35 |      121 |        78 |              17 |   95
         201 |        85 |       37.32 |      119 |        78 |              18 |   96
         201 |        85 |       37.02 |      119 |        79 |              17 |   96
         201 |        83 |       36.73 |      121 |        81 |              17 |   96
         201 |        85 |       36.63 |      116 |        81 |              18 |   97
         201 |        84 |       36.59 |      118 |        79 |              18 |   97
         201 |        82 |       36.50 |      119 |        79 |              18 |   97
         201 |        80 |       36.50 |      121 |        79 |              17 |   98
         201 |        81 |       36.50 |      119 |        81 |              18 |   98
         201 |        81 |       36.50 |      120 |        79 |              17 |   99
         201 |        82 |       36.50 |      120 |        80 |              17 |   99
         201 |        81 |       36.50 |      123 |        81 |              17 |   99
         201 |        79 |       36.50 |      119 |        79 |              18 |   99
         201 |        79 |       36.50 |      120 |        81 |              18 |   99
         201 |        80 |       36.50 |      122 |        79 |              18 |   99
         201 |        81 |       36.50 |      119 |        80 |              17 |   99
         201 |        79 |       36.50 |      122 |        81 |              16 |   99
         201 |        77 |       36.50 |      118 |        80 |              15 |   99
         201 |        76 |       36.50 |      121 |        80 |              14 |   99
         201 |        78 |       36.50 |      119 |        81 |              13 |   99
         201 |        79 |       36.50 |      121 |        79 |              12 |   99
         201 |        79 |       36.50 |      118 |        80 |              12 |   99
         201 |        81 |       36.50 |      119 |        81 |              12 |   99
         201 |        80 |       36.50 |      121 |        78 |              13 |   99
         201 |        78 |       36.50 |      118 |        78 |              12 |   99
```

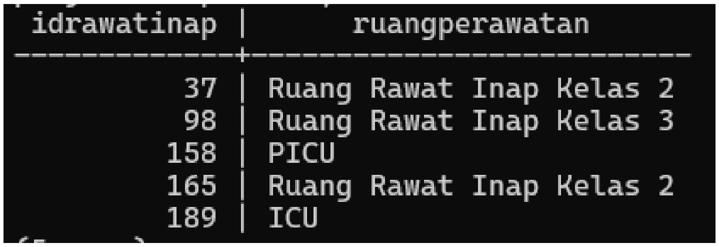
Deskripsi

Mencari Rawat Inap yang didalamnya terjadinya kejadian tertentu (misal mati lampu RS).
CONTAINS (Y di X)

Query

```
SELECT IDRawatInap, RuangPerawatan
FROM RawatInap
WHERE allen_contains(
    ValidStart, ValidEnd,
    '2024-05-01 08:00:00', '2024-05-02 08:00:00' -- Waktu
);
```

<table>
<tr><td>Hasil</td><td>



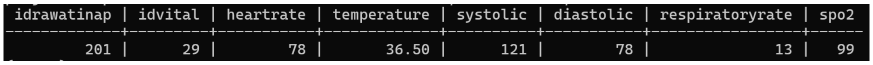
```
 idrawatinap |      ruangperawatan
-------------+------------------------
          37 | Ruang Rawat Inap Kelas 2
          98 | Ruang Rawat Inap Kelas 3
         158 | PICU
         165 | Ruang Rawat Inap Kelas 2
         189 | ICU
```

</td></tr>
</table>

<table>
<tr><td>Deskripsi</td><td>Mencari pemeriksaan Tanda Vital terakhir yang selesai tepat bersamaan dengan waktu pasien pulang.
FINISHES (X f Y)</td></tr>
<tr><td>Query</td><td>

```
SELECT r.IDRawatInap,
       tv.IDVital,
       tv.HeartRate,
       tv.Temperature,
       tv.Systolic,
       tv.Diastolic,
       tv.RespiratoryRate,
       tv.SPO2
FROM TandaVital tv
JOIN RawatInap r ON tv.IDRawatInap = r.IDRawatInap
WHERE r.IDPasien = 79
  AND allen_finishes(tv.ValidStart, tv.ValidEnd,
                     r.ValidStart, r.ValidEnd);
```

</td></tr>
<tr><td>Hasil</td><td>



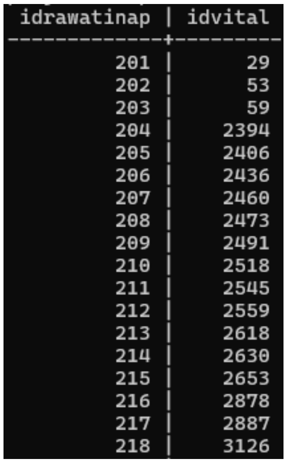
```
 idrawatinap | idvital | heartrate | temperature | systolic | diastolic | respiratoryrate | spo2
-------------+---------+-----------+-------------+----------+-----------+-----------------+------
         201 |      29 |        78 |       36.50 |      121 |        78 |              13 |   99
```

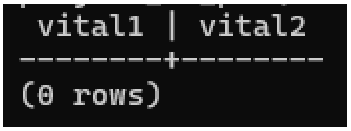
</td></tr>
</table>

<table>
<tr><td>Deskripsi</td><td>Kebalikan Finishes. Rawat Inap selesai bersamaan dengan selesainya pemantauan tertentu.
FINISHED BY (Y fi X)</td></tr>
<tr><td>Query</td><td>

```
SELECT r.IDRawatInap, tv.IDVital
FROM RawatInap r
JOIN TandaVital tv ON r.IDRawatInap = tv.IDRawatInap
WHERE allen_finished_by(r.ValidStart, r.ValidEnd,
                        tv.ValidStart, tv.ValidEnd);
```

</td></tr>
</table>

<table>
<tr><td>Hasil</td><td>
<pre>
 idrawatinap | idvital
-------------+---------
         201 | 29
         202 | 53
         203 | 59
         204 | 2394
         205 | 2406
         206 | 2436
         207 | 2460
         208 | 2473
         209 | 2491
         210 | 2518
         211 | 2545
         212 | 2559
         213 | 2618
         214 | 2630
         215 | 2653
         216 | 2878
         217 | 2887
         218 | 3126
</pre></td></tr>
</table>

<table>
<tr><td>Deskripsi</td><td>Mencari duplikasi data. Misal mencari dua data Tanda Vital yang memiliki rentang validitas waktu yang persis sama (kemungkinan entri ganda).
EQUALS (X = Y)</td></tr>
<tr><td>Query</td><td><pre>
SELECT tv1.IDVital AS Vital1,
       tv2.IDVital AS Vital2
FROM TandaVital tv1
JOIN TandaVital tv2 ON tv1.IDRawatInap = tv2.IDRawatInap
WHERE tv1.IDVital < tv2.IDVital -- Hindari self-join
  AND allen_equals(tv1.ValidStart, tv1.ValidEnd,
                   tv2.ValidStart, tv2.ValidEnd);
</pre></td></tr>
<tr><td>Hasil</td><td>
<pre>
 vital1 | vital2
--------+--------
(0 rows)
</pre></td></tr>
</table>

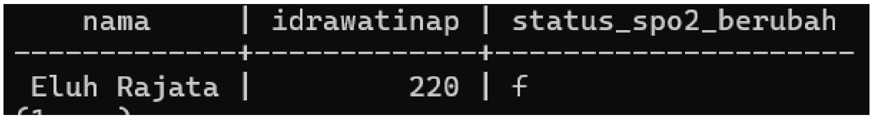
3. Predikat Temporal

<table>
<tr><td>Deskripsi</td><td>(<i>Changed</i>)
Dokter ingin tahu apakah kondisi oksigen (SPO2) pasien stabil atau naik-turun selama satu hari. Jika hasil FALSE, berarti pasien sangat stabil (nilai konstan). Jika TRUE, berarti terjadi perubahan.</td></tr>
<tr><td>Query</td><td><pre>
-- Contoh 1
SELECT
    p.Nama,
    r.IDRawatInap,
    temporal_changed(
        'tandavital', 'idrawatinap', r.IDRawatInap, 'spo2',
        '2024-07-05 00:00:00', '2024-07-05 23:59:59'
</pre></td></tr>
</table>

```
    ) AS status_spo2_berubah
FROM Pasien p
JOIN RawatInap r ON p.IDPasien = r.IDPasien
WHERE r.IDRawatInap = 220;

-- Contoh 2
SELECT
    p.Nama,
    r.IDRawatInap,
    temporal_changed(
        'tandavital', 'idrawatinap', r.IDRawatInap, 'spo2',
        '2024-06-21 00:00:00', '2024-06-21 23:59:59'
    ) AS status_spo2_berubah
FROM Pasien p
JOIN RawatInap r ON p.IDPasien = r.IDPasien
WHERE r.IDRawatInap = 220;
```

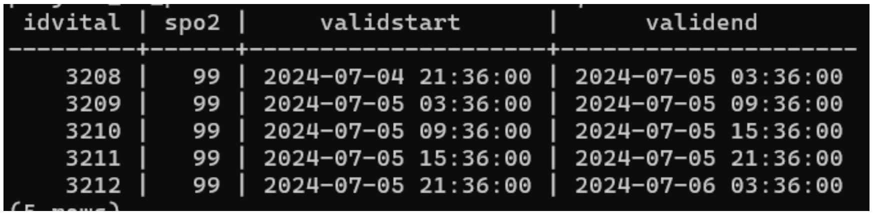
Hasil

Hasil contoh 1



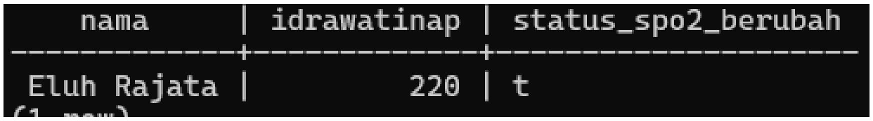
```
     nama     | idrawatinap | status_spo2_berubah
--------------+-------------+---------------------
 Eluh Rajata  |         220 | f
```

Data asli



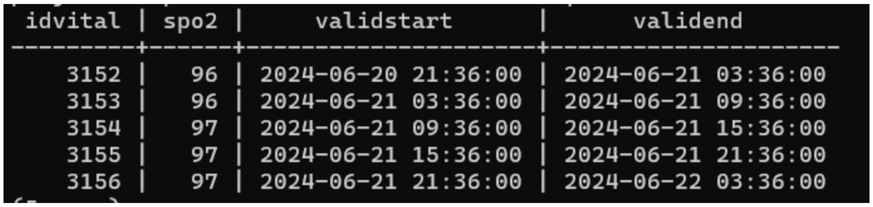
```
 idvital | spo2 |      validstart      |       validend
---------+------+---------------------+---------------------
    3208 |   99 | 2024-07-04 21:36:00 | 2024-07-05 03:36:00
    3209 |   99 | 2024-07-05 03:36:00 | 2024-07-05 09:36:00
    3210 |   99 | 2024-07-05 09:36:00 | 2024-07-05 15:36:00
    3211 |   99 | 2024-07-05 15:36:00 | 2024-07-05 21:36:00
    3212 |   99 | 2024-07-05 21:36:00 | 2024-07-06 03:36:00
```

Hasil contoh 2



```
     nama     | idrawatinap | status_spo2_berubah
--------------+-------------+---------------------
 Eluh Rajata  |         220 | t
```

Data asli



```
 idvital | spo2 |      validstart      |       validend
---------+------+---------------------+---------------------
    3152 |   96 | 2024-06-20 21:36:00 | 2024-06-21 03:36:00
    3153 |   96 | 2024-06-21 03:36:00 | 2024-06-21 09:36:00
    3154 |   97 | 2024-06-21 09:36:00 | 2024-06-21 15:36:00
    3155 |   97 | 2024-06-21 15:36:00 | 2024-06-21 21:36:00
    3156 |   97 | 2024-06-21 21:36:00 | 2024-06-22 03:36:00
```

<table>
<tr>
<td>Deskripsi</td>
<td>

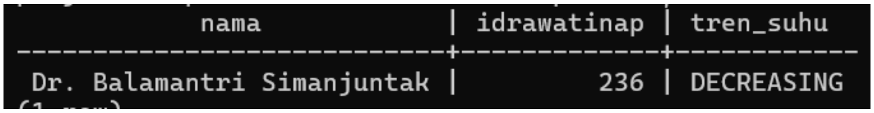
(*Trend*)
Perawat ingin mencari tren suhu tubuh pasien pada hari tertentu (*Temperature*).

</td>
</tr>
<tr>
<td>Query</td>
<td>

```
-- Contoh 1
SELECT
    p.Nama,
    r.IDRawatInap,
    temporal_trend(
        'tandavital', 'idrawatinap', r.IDRawatInap,
'temperature',
        '2024-03-27 00:00:00', '2024-03-27 23:59:59'
    ) AS tren_suhu
FROM RawatInap r
JOIN Pasien p ON r.IDPasien = p.IDPasien
WHERE r.IDRawatInap = 236;

-- Contoh 2
SELECT
    p.Nama,
    r.IDRawatInap,
    temporal_trend(
        'tandavital', 'idrawatinap', r.IDRawatInap,
'temperature',
        '2024-03-29 00:00:00', '2024-03-29 23:59:59'
    ) AS tren_suhu
FROM RawatInap r
JOIN Pasien p ON r.IDPasien = p.IDPasien
WHERE r.IDRawatInap = 236;
```

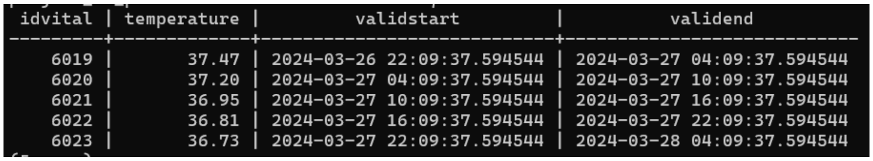
</td>
</tr>
<tr>
<td>Hasil</td>
<td>

Hasil contoh 1



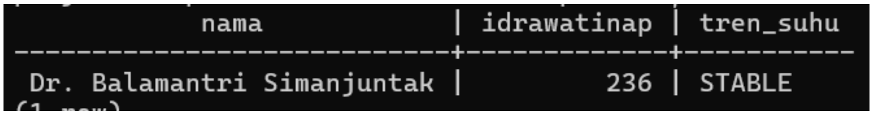
```
           nama              | idrawatinap | tren_suhu
-----------------------------+-------------+------------
 Dr. Balamantri Simanjuntak |         236 | DECREASING
```

Data asli



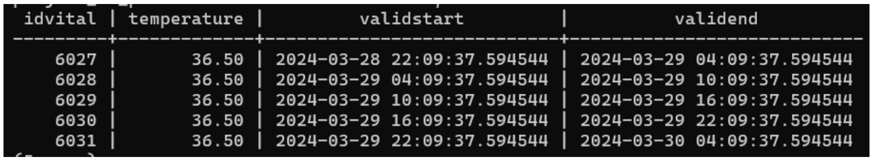
```
 idvital | temperature |         validstart         |          validend
---------+-------------+----------------------------+----------------------------
    6019 |       37.47 | 2024-03-26 22:09:37.594544 | 2024-03-27 04:09:37.594544
    6020 |       37.20 | 2024-03-27 04:09:37.594544 | 2024-03-27 10:09:37.594544
    6021 |       36.95 | 2024-03-27 10:09:37.594544 | 2024-03-27 16:09:37.594544
    6022 |       36.81 | 2024-03-27 16:09:37.594544 | 2024-03-27 22:09:37.594544
    6023 |       36.73 | 2024-03-27 22:09:37.594544 | 2024-03-28 04:09:37.594544
```

Hasil contoh 2



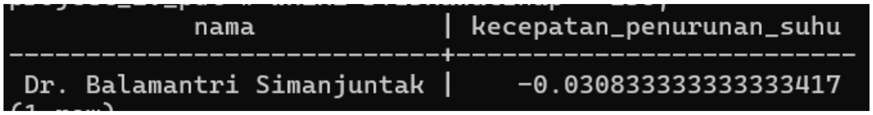
```
           nama              | idrawatinap | tren_suhu
-----------------------------+-------------+-----------
 Dr. Balamantri Simanjuntak |         236 | STABLE
```

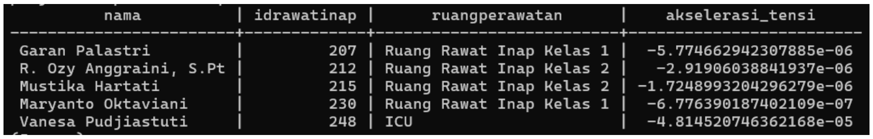
Data asli



```
 idvital | temperature |         validstart         |          validend
---------+-------------+----------------------------+----------------------------
    6027 |       36.50 | 2024-03-28 22:09:37.594544 | 2024-03-29 04:09:37.594544
    6028 |       36.50 | 2024-03-29 04:09:37.594544 | 2024-03-29 10:09:37.594544
    6029 |       36.50 | 2024-03-29 10:09:37.594544 | 2024-03-29 16:09:37.594544
    6030 |       36.50 | 2024-03-29 16:09:37.594544 | 2024-03-29 22:09:37.594544
    6031 |       36.50 | 2024-03-29 22:09:37.594544 | 2024-03-30 04:09:37.594544
```

</td>
</tr>
</table>

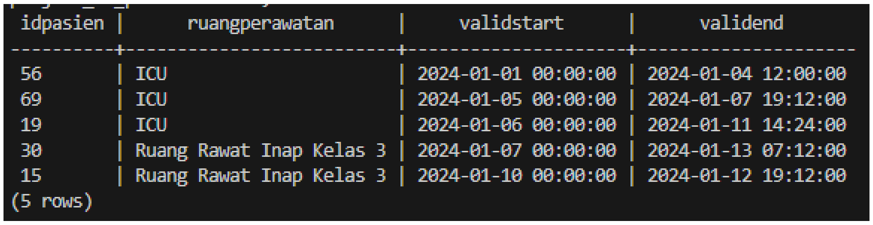
<table>
<tr><td>Deskripsi</td><td>(<em>Speed</em>)
Dokter ingin tahu seberapa cepat suhu pasien turun/naik per satuan waktunya pada hari tertentu.</td></tr>
<tr><td>Query</td><td><pre>
SELECT
    p.Nama,
    temporal_speed(
        'tandavital', 'idrawatinap',
        r.IDRawatInap, 'systolic',
        '2024-03-27 00:00:00', '2024-03-27 23:59:59'
) AS kecepatan_penurunan_suhu
FROM RawatInap r
JOIN Pasien p ON r.IDPasien = p.IDPasien
WHERE r.IDRawatInap = 236;
</pre></td></tr>
<tr><td>Hasil</td><td>
<pre>
            nama | kecepatan_penurunan_suhu
----------------------------+--------------------------
 Dr. Balamantri Simanjuntak | -0.030833333333333417
</pre></td></tr>
</table>

<table>
<tr><td>Deskripsi</td><td>(<em>Acceleration</em>)
Dokter ingin mencari tahu pasien yang pernah mengalami penurunan tekanan darah yang semakin cepat (akselerasi negatif).</td></tr>
<tr><td>Query</td><td><pre>
SELECT
    p.Nama,
    r.IDRawatInap,
    r.RuangPerawatan,
    temporal_acceleration(
        'tandavital', 'idrawatinap', r.IDRawatInap,
'systolic',
        '2024-01-01 00:00:00', '2024-12-31 23:59:59'
) AS akselerasi_tensi
FROM RawatInap r
JOIN Pasien p ON r.IDPasien = p.IDPasien
WHERE temporal_acceleration(
    'tandavital', 'idrawatinap', r.IDRawatInap, 'systolic',
    '2024-01-01 00:00:00', '2024-12-31 23:59:59'
) < -5
</pre></td></tr>
<tr><td>Hasil</td><td>
<pre>
         nama | idrawatinap | ruangperawatan | akselerasi_tensi
-----------------------+-------------+--------------------------+------------------------
 Garan Palastri | 207 | Ruang Rawat Inap Kelas 1 | -5.774662942307885e-06
 R. Ozy Anggraini, S.Pt| 212 | Ruang Rawat Inap Kelas 2 | -2.91906038841937e-06
 Mustika Hartati | 215 | Ruang Rawat Inap Kelas 2 | -1.7248993204296279e-06
 Maryanto Oktaviani | 230 | Ruang Rawat Inap Kelas 1 | -6.776390187402109e-07
 Vanesa Pudjiastuti | 248 | ICU | -4.814520746362168e-05
</pre></td></tr>
</table>

4. *Temporal Operations*

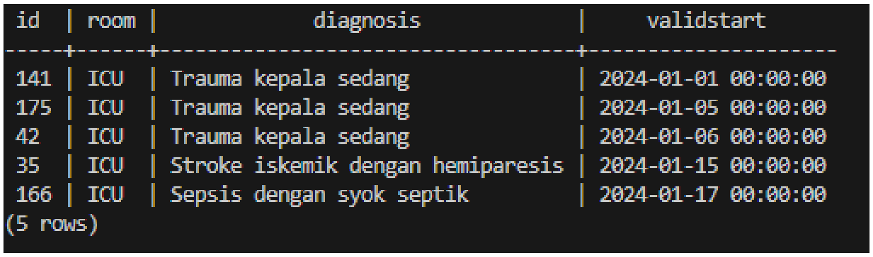
Deskripsi	Menampilkan daftar pasien rawat inap beserta ruang perawatannya, lengkap dengan periode waktu rawat inap.

Query	SELECT
	    data->>'idpasien' as IDPasien,
	    data->>'ruangperawatan' as RuangPerawatan,
	    validstart,
	    validend
	FROM temporal_projection('rawatinap', ARRAY['idpasien',
	'ruangperawatan'])
	LIMIT 5;

Hasil	 idpasien | ruangperawatan | validstart | validend ----------+-------------------------+---------------------+--------------------- 56 | ICU | 2024-01-01 00:00:00 | 2024-01-04 12:00:00 69 | ICU | 2024-01-05 00:00:00 | 2024-01-07 19:12:00 19 | ICU | 2024-01-06 00:00:00 | 2024-01-11 14:24:00 30 | Ruang Rawat Inap Kelas 3 | 2024-01-07 00:00:00 | 2024-01-13 07:12:00 15 | Ruang Rawat Inap Kelas 3 | 2024-01-10 00:00:00 | 2024-01-12 19:12:00 (5 rows)

Deskripsi	Mencari seluruh pasien yang pernah dirawat di ruang ICU, beserta diagnosis dan waktu masuknya.

Query	SELECT
	    data->>'idrawatinap' as ID,
	    data->>'ruangperawatan' as Room,
	    data->>'diagnosis' as Diagnosis,
	    validstart
	FROM temporal_selection('rawatinap', 'ruangperawatan =
	''ICU''')
	LIMIT 5;

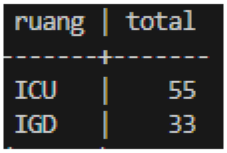
Hasil	 id | room | diagnosis | validstart -----+------+-----------------------------------+--------------------- 141 | ICU | Trauma kepala sedang | 2024-01-01 00:00:00 175 | ICU | Trauma kepala sedang | 2024-01-05 00:00:00 42 | ICU | Trauma kepala sedang | 2024-01-06 00:00:00 35 | ICU | Stroke iskemik dengan hemiparesis | 2024-01-15 00:00:00 166 | ICU | Sepsis dengan syok septik | 2024-01-17 00:00:00 (5 rows)

Deskripsi	Berapa total rawat inap ICU + IGD per ruang?

Query	SELECT
	  data->>'ruangperawatan' AS ruang,
	  COUNT(*) AS total
	FROM temporal_union(

```
  'rawatinap',
  'ruangperawatan = ''ICU''',
  'ruangperawatan = ''IGD'''
)
GROUP BY data->>'ruangperawatan';
```

Hasil



```
ruang | total
-------+-------
ICU   |    55
IGD   |    33
```

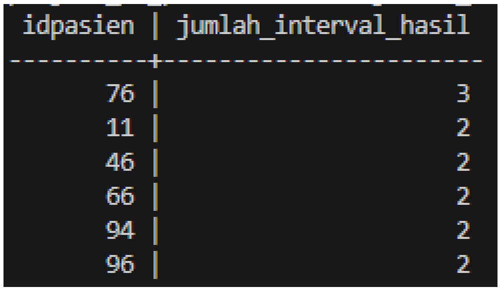
Deskripsi

Apakah ada periode ICU pasien yang terpotong atau terbelah karena adanya perawatan IGD di tengahnya?

Query

```
SELECT
  (data->>'idpasien')::int AS idpasien,
  COUNT(*) AS jumlah_interval_hasil
FROM temporal_set_difference(
  'rawatinap',
  'idpasien',
  'ruangperawatan = ''ICU''',
  'ruangperawatan = ''IGD'''
)
GROUP BY (data->>'idpasien')::int
HAVING COUNT(*) > 1
ORDER BY jumlah_interval_hasil DESC, idpasien;
```

Hasil



```
 idpasien | jumlah_interval_hasil
----------+-----------------------
       76 |                     3
       11 |                     2
       46 |                     2
       66 |                     2
       94 |                     2
       96 |                     2
```

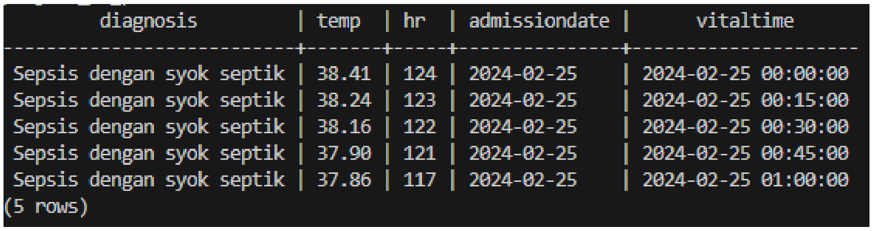
Deskripsi

Menampilkan tanda vital pasien ICU atau pasien dengan diagnosis sepsis selama masa rawat inapnya.

Query

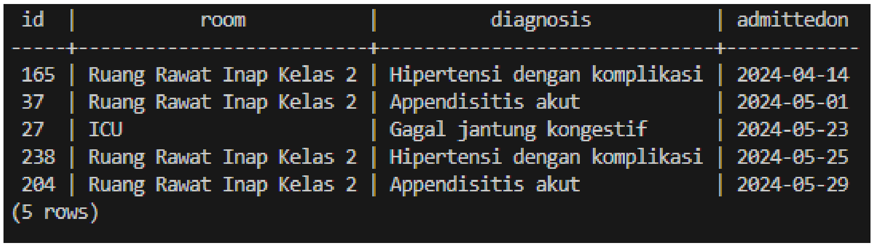
```
SELECT
    data->>'diagnosis' as Diagnosis,
    data->>'temperature' as Temp,
    data->>'heartrate' as HR,
    validstart_t1::DATE as AdmissionDate,
    validstart_t2 as VitalTime
FROM temporal_join(
```

	'rawatinap',
	    'tandavital',
	    'idrawatinap',
	    ARRAY['diagnosis'],
	    ARRAY['temperature', 'heartrate']
	)
	WHERE data->>'diagnosis' LIKE '%ICU%'
	   OR data->>'diagnosis' LIKE '%Sepsis%'
	LIMIT 5;

Hasil	
	`       diagnosis          | temp  | hr  | admissiondate |      vitaltime`
	`--------------------------+-------+-----+---------------+---------------------`
	` Sepsis dengan syok septik | 38.41 | 124 | 2024-02-25    | 2024-02-25 00:00:00`
	` Sepsis dengan syok septik | 38.24 | 123 | 2024-02-25    | 2024-02-25 00:15:00`
	` Sepsis dengan syok septik | 38.16 | 122 | 2024-02-25    | 2024-02-25 00:30:00`
	` Sepsis dengan syok septik | 37.90 | 121 | 2024-02-25    | 2024-02-25 00:45:00`
	` Sepsis dengan syok septik | 37.86 | 117 | 2024-02-25    | 2024-02-25 01:00:00`
	`(5 rows)`

Deskripsi	Menampilkan daftar pasien yang sedang dirawat saat ini.

Query	SELECT
	    data->>'idrawatinap' as ID,
	    data->>'ruangperawatan' as Room,
	    data->>'diagnosis' as Diagnosis,
	    validstart::DATE as AdmittedOn
	FROM temporal_timeslice('rawatinap', CURRENT_TIMESTAMP)
	LIMIT 5;

Hasil	
	` id  |           room           |           diagnosis            | admittedon`
	`-----+--------------------------+--------------------------------+------------`
	` 165 | Ruang Rawat Inap Kelas 2 | Hipertensi dengan komplikasi | 2024-04-14`
	` 37  | Ruang Rawat Inap Kelas 2 | Appendisitis akut            | 2024-05-01`
	` 27  | ICU                      | Gagal jantung kongestif      | 2024-05-23`
	` 238 | Ruang Rawat Inap Kelas 2 | Hipertensi dengan komplikasi | 2024-05-25`
	` 204 | Ruang Rawat Inap Kelas 2 | Appendisitis akut            | 2024-05-29`
	`(5 rows)`

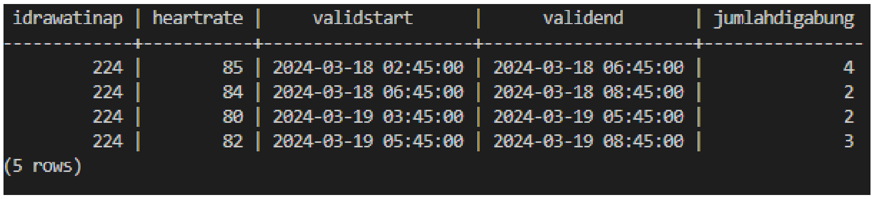
5. *Coalesce*

Deskripsi	Menggabungkan record tanda vital yang memiliki nilai HeartRate sama dan interval waktu yang berurutan (adjacent) menjadi satu record dengan interval yang lebih panjang

Query	SELECT
	    (data->>'idrawatinap')::INT as IDRawatInap,
	     (data->>'heartrate')::INT as HeartRate,
	     validstart as ValidStart,
	     validend as ValidEnd,

```
    coalesced_count as JumlahDigabung
FROM temporal_coalesce(
    'tandavital',
    ARRAY['idrawatinap', 'heartrate'],
    'idrawatinap = 224'
)
)
WHERE coalesced_count > 1
LIMIT 5;"
```

Hasil



```
 idrawatinap | heartrate |     validstart      |       validend      | jumlahdigabung
-------------+-----------+---------------------+---------------------+----------------
         224 |        85 | 2024-03-18 02:45:00 | 2024-03-18 06:45:00 |              4
         224 |        84 | 2024-03-18 06:45:00 | 2024-03-18 08:45:00 |              2
         224 |        80 | 2024-03-19 03:45:00 | 2024-03-19 05:45:00 |              2
         224 |        82 | 2024-03-19 05:45:00 | 2024-03-19 08:45:00 |              3
(5 rows)
```

Deskripsi

Menampilkan hasil coalesce beserta ID Vital mana saja yang digabungkan, berguna untuk audit trail dan verifikasi.

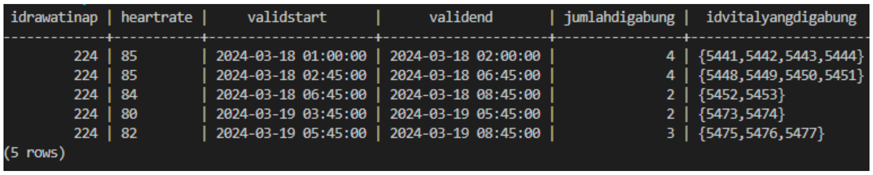
Untuk membuktikan query tersebut, maka ditampilkan juga data asli sebelum di-coalesce untuk membuktikan bahwa 4 record dengan HeartRate=85 memiliki interval yang adjacent (ValidEnd record sebelumnya = ValidStart record berikutnya).

Query

```
Query Utama:
SELECT
    idrawatinap as IDRawatInap,
    vital_data->>'heartrate' as HeartRate,
    validstart as ValidStart,
    validend as ValidEnd,
    coalesced_count as JumlahDigabung,
    merged_ids as IDVitalYangDigabung
FROM coalesce_tandavital(224, ARRAY['heartrate'])
WHERE coalesced_count > 1
LIMIT 5;

Query Mengambil Data Asli:
SELECT idvital, heartrate, validstart, validend
FROM tandavital
WHERE idvital IN (5441, 5442, 5443, 5444)
ORDER BY validstart;
```

Hasil

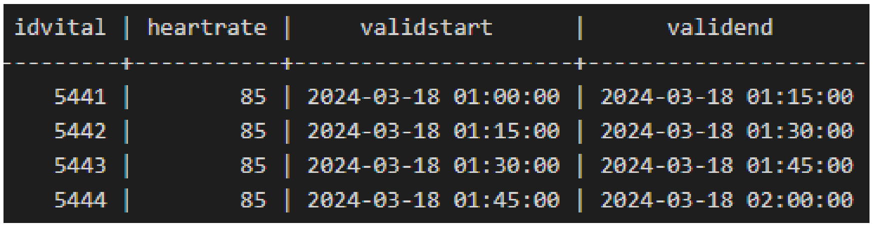
```
Hasil Coalesce:
```



```
 idrawatinap | heartrate |     validstart      |       validend      | jumlahdigabung |   idvitalyangdigabung
-------------+-----------+---------------------+---------------------+----------------+-------------------------
         224 | 85        | 2024-03-18 01:00:00 | 2024-03-18 02:00:00 |              4 | {5441,5442,5443,5444}
         224 | 85        | 2024-03-18 02:45:00 | 2024-03-18 06:45:00 |              4 | {5448,5449,5450,5451}
         224 | 84        | 2024-03-18 06:45:00 | 2024-03-18 08:45:00 |              2 | {5452,5453}
         224 | 80        | 2024-03-19 03:45:00 | 2024-03-19 05:45:00 |              2 | {5473,5474}
         224 | 82        | 2024-03-19 05:45:00 | 2024-03-19 08:45:00 |              3 | {5475,5476,5477}
(5 rows)
```

<table>
<tr>
<td></td>
<td>

Data Asli:



```
 idvital | heartrate |      validstart      |       validend
---------+-----------+---------------------+---------------------
    5441 |        85 | 2024-03-18 01:00:00 | 2024-03-18 01:15:00
    5442 |        85 | 2024-03-18 01:15:00 | 2024-03-18 01:30:00
    5443 |        85 | 2024-03-18 01:30:00 | 2024-03-18 01:45:00
    5444 |        85 | 2024-03-18 01:45:00 | 2024-03-18 02:00:00
```

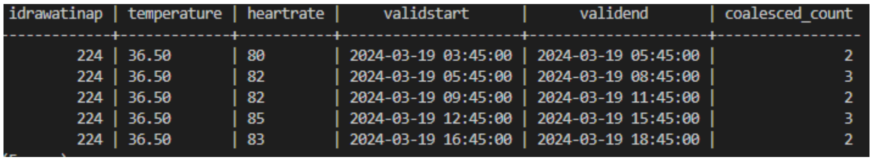
*Terbukti adjacent

</td>
</tr>
</table>

<table>
<tr>
<td>Deskripsi</td>
<td>Menggabungkan record yang memiliki nilai Temperature DAN HeartRate yang sama sekaligus. Record hanya akan digabung jika KEDUA nilai identik.</td>
</tr>
<tr>
<td>Query</td>
<td>

```
SELECT
    idrawatinap,
    vital_data->>'temperature' as Temperature,
    vital_data->>'heartrate' as HeartRate,
    validstart,
    validend,
    coalesced_count
FROM coalesce_tandavital(224, ARRAY['temperature',
'heartrate'])
WHERE coalesced_count > 1
LIMIT 5;
```

</td>
</tr>
<tr>
<td>Hasil</td>
<td>



```
 idrawatinap | temperature | heartrate |      validstart      |       validend       | coalesced_count
-------------+-------------+-----------+---------------------+---------------------+-----------------
         224 | 36.50       | 80        | 2024-03-19 03:45:00 | 2024-03-19 05:45:00 |               2
         224 | 36.50       | 82        | 2024-03-19 05:45:00 | 2024-03-19 08:45:00 |               3
         224 | 36.50       | 82        | 2024-03-19 09:45:00 | 2024-03-19 11:45:00 |               2
         224 | 36.50       | 85        | 2024-03-19 12:45:00 | 2024-03-19 15:45:00 |               3
         224 | 36.50       | 83        | 2024-03-19 16:45:00 | 2024-03-19 18:45:00 |               2
```

</td>
</tr>
</table>

4. KESIMPULAN DAN *LESSON LEARNED*

Implementasi basis data temporal pada PostgreSQL berhasil menunjukkan bahwa konsep valid time dapat diterapkan secara eksplisit melalui penambahan atribut `ValidStart` dan `ValidEnd` pada tabel fakta seperti RawatInap dan TandaVital. Di samping itu, rancangan fungsi-fungsi `Insert`, `Update`, dan `Delete` yang memodifikasi `ValidEnd` alih-alih menghapus atau menimpa baris memungkinkan penerapan mekanisme versioning secara manual di atas PostgreSQL. Hal ini diperlukan sebab PostgreSQL tidak memiliki system-versioned table bawaan.

Tidak hanya itu, penerapan Allen's 13 Temporal Relationships dan berbagai predikat temporal (Changed, Trend, Speed, Acceleration) menunjukkan bahwa PostgreSQL cukup fleksibel untuk mendukung query-query temporal tingkat lanjut dengan memanfaatkan kombinasi tipe waktu, fungsi PL/pgSQL, dan operator interval. Terakhir, operasi temporal seperti *projection, selection, union, set difference, join, timeslice,* dan *coalesce* memperlihatkan bahwa data temporal dapat dimanipulasi dengan cara yang mendekati model teoritis, misalnya untuk menggabungkan interval adjacent atau mengambil snapshot pada suatu waktu tertentu.

Dengan demikian, sistem dapat menjawab kebutuhan analisis temporal untuk mendukung kebutuhan medis praktis, seperti mengecek tren kenaikan suhu, kestabilan SPO2, atau percepatan penurunan tekanan darah. Hal ini menegaskan bahwa basis data temporal bukan hanya konsep teoretis, tetapi dapat langsung mendukung pengambilan keputusan klinis melalui query yang lebih kaya makna terhadap dimensi waktu.

Di samping itu, kami juga mendapatkan beberapa pelajaran penting:

1. Perancangan skema temporal membutuhkan pemikiran jauh lebih matang dibandingkan skema relasional biasa, khususnya dalam menentukan dimensi waktu apa yang ingin direkam (valid time vs transaction time) dan di tabel mana atribut temporal diletakkan. Kesalahan pada tahap desain,

misalnya tidak konsisten mengisi `ValidStart`/`ValidEnd` atau tidak memberikan constraint waktu, dapat berujung pada data historis yang sulit dianalisis.

2. Implementasi manual system versioning di PostgreSQL melalui fungsi `Insert`/`Update`/`Delete` menunjukkan bahwa logika bisnis temporal sebaiknya dipusatkan di level *database*, bukan di aplikasi, agar konsistensi historis lebih terjamin.
3. *Allen's 13 Relationship* sangat kuat untuk mengekspresikan berbagai skenario waktu, tetapi penerapannya dalam bentuk fungsi PL/pgSQL cukup *verbose* dan rawan kesalahan jika tidak terdokumentasi dengan baik. Pengalaman ini menekankan pentingnya membuat *helper* generik (seperti `getEndTime`) dan menamai fungsi secara konsisten agar *query* yang menggunakan predikat temporal tetap *readable* dan tidak mengulang-ulang mengetik kode yang sama.
4. Implementasi generik berbasis dynamic SQL menunjukkan *trade-off* antara fleksibilitas dan kompleksitas, sehingga pada sistem produksi perlu dipertimbangkan aspek performa, *indexing*, dan batasan domain nilai untuk menjaga *query* tetap efisien.
5. Operasi temporal lanjutan seperti *temporal join*, *timeslice*, dan *coalesce* menunjukkan bahwa desain API/fungsi yang generik dan *reusable* sangat membantu ketika jumlah *query* analitis mulai bertambah. Di sisi lain, pengalaman ini juga menekankan bahwa tanpa *naming convention* dan dokumentasi yang konsisten, fungsi-fungsi temporal yang banyak dan kompleks dapat dengan mudah membingungkan *developer* lain yang akan memelihara sistem di kemudian hari.

5. PEMBAGIAN KERJA

NIM	TUGAS
13522006	- Fungsi Allen’s 13 predicate - Fungsi predikat temporal - Query terkait allen’s 13 predicate dan predikat temporal
13522033	- Eksplorasi DBMS - Membuat skema - Fungsi insert, update, dan delete - Query terkait insert, update, dan delete
13522040	- Mencari studi kasus - Fungsi coalesce - Query terkait coalesce - Kesimpulan dan lesson learned
13522061	- Mencari studi kasus - Membuat skema - Seeding data - Fungsi temporal operations - Query terkait temporal operations - Mengedit video

6. LAMPIRAN

Tautan video: https://youtu.be/kcaqewipCsQ?si=loNVNNrAjwXbZrKU

Sample data:

https://github.com/BryanLauw/Project-IV-IF4040-PDL/tree/main/data_sample