

**LAPORAN TUTORIAL LAB 3
BASIS DATA**



**EDUARDUS TJTIRAHARDJA
2106653602
KELAS A**

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS INDONESIA
DEPOK
2022/2023**

Laporan Tutorial Lab

Basis Data

Semester Ganjil 2022/2023

Latihan

1. [SQL] Jalankan seluruh contoh 1 hingga contoh 8 di atas!

- Contoh 1

```
eduardus.tjitrahardja.siwana> -- Contoh 1
CREATE OR REPLACE FUNCTION SIWANAP.diskon_harga(idkamar VARCHAR(10))
    RETURNS INTEGER AS
$$
DECLARE
    harga_awal    INTEGER;
    harga_diskon  INTEGER;
BEGIN
    SELECT harga
    INTO harga_awal
    FROM KAMAR
    WHERE id_kamar = idkamar;

    harga_diskon := (harga_awal * 9 / 10);

    UPDATE KAMAR
    SET harga = harga_diskon
    WHERE id_kamar = idkamar;

    RETURN harga_diskon;
END;
$$
LANGUAGE plpgsql

[2022-11-09 21:12:07] completed in 48 ms
```

```
eduardus.tjitrahardja=> \df
```

Schema	Name	Result data type	Argument data types	Type
siwana	diskon_harga	integer	idkamar character varying	func

(1 row)

- Contoh 2

```
eduardus.tjitrahardja.siwana> SELECT diskon_harga('KA01')
```

[2022-11-09 21:15:59] 1 row retrieved starting from 1 in 251 ms (6)

	diskon_harga
1	153000

Laporan Tutorial Lab

Basis Data

Semester Ganjil 2022/2023



- Contoh 3

The screenshot shows a PostgreSQL query client interface. The left pane displays the query: `SELECT diskon_harga(id_kamar) FROM KAMAR`. The right pane shows the results of the query, which is a table with 26 rows. The first column is an index from 1 to 26, and the second column is the value of the `diskon_harga` function for each `id_kamar`.

	diskon_harga
1	153000
2	153000
3	153000
4	153000
5	270000
6	270000
7	270000
8	270000
9	270000
10	630000
11	630000
12	630000
13	630000
14	630000
15	990000
16	990000
17	990000
18	990000
19	990000
20	990000
21	990000
22	990000
23	990000
24	990000
25	990000
26	990000

Laporan Tutorial Lab

Basis Data

Semester Ganjil 2022/2023



- Contoh 4

```
Output
eduardus.tjitrahardja.sivanap> -- Contoh 4
                                CREATE OR REPLACE FUNCTION diskon_semua_harga()
                                RETURNS void AS
                                $$
                                DECLARE
                                    temp_row    RECORD;
                                    harga_diskon INTEGER;
                                BEGIN
                                    FOR temp_row IN
                                        SELECT *
                                        FROM KAMAR
                                        LOOP
                                            harga_diskon := (temp_row.harga * 9 / 10);

                                            UPDATE KAMAR
                                            SET harga = harga_diskon
                                            WHERE id_kamar = temp_row.id_kamar;
                                        END LOOP;
                                END;
                                $$
                                LANGUAGE plpgsql

[2022-11-09 21:45:22] completed in 88 ms
```

```
Output
eduardus.tjitrahardja.sivanap> SELECT diskon_semua_harga()
[2022-11-09 21:45:44] 1 row retrieved starting from 1 in 109 ms (execution: 53 ms, fetching: 56 ms)

diskon_semua_harga()void
1
```

- Contoh 5

```
Output
eduardus.tjitrahardja.sivanap> DROP FUNCTION diskon_harga(idkamar VARCHAR(10))
[2022-11-09 21:46:04] completed in 39 ms

diskon_semua_harga()void
1
```

Laporan Tutorial Lab

Basis Data

Semester Ganjil 2022/2023



- Contoh 6

```
Output
eduardus.tjitrahardja.siwonap> -- Contoh 6
CREATE OR REPLACE FUNCTION cek_jumlah_shift()
    RETURNS trigger AS
$$
DECLARE
    shift_count integer;
BEGIN
    IF (TG_OP = 'INSERT') THEN
        SELECT COUNT(*)
        INTO shift_count
        FROM SHIFT_PERAWAT
        WHERE id_perawat = NEW.id_perawat
        GROUP BY id_perawat;
        IF (shift_count >= 5) THEN
            RAISE EXCEPTION 'Maaf, perawat tidak boleh memiliki shift melebihi 5';
        END IF;
        RETURN NEW;
    END IF;
END;
$$
LANGUAGE plpgsql

[2022-11-09 21:46:38] completed in 30 ms
```

- Contoh 7

```
Output
eduardus.tjitrahardja.siwonap> -- Contoh 7
CREATE TRIGGER trigger_cek_jumlah_shift
    BEFORE INSERT
    ON SHIFT_PERAWAT
    FOR EACH ROW
    EXECUTE PROCEDURE cek_jumlah_shift()

[2022-11-09 21:47:05] completed in 31 ms
```

- Contoh 8

```
Output
eduardus.tjitrahardja.siwonap> INSERT INTO SHIFT_PERAWAT (id_shift_perawat, id_perawat, id_rawat_inap, waktu_mulai, waktu_akhir)
VALUES ('SP101', 'PE13', 'RI20', '2020-11-30 00:00', '2020-11-30 12:00')
[2022-11-09 21:47:43] [P0001] ERROR: Maaf, perawat tidak boleh memiliki shift melebihi 5
eduardus.tjitrahardja.siwonap> INSERT INTO SHIFT_PERAWAT (id_shift_perawat, id_perawat, id_rawat_inap, waktu_mulai, waktu_akhir)
VALUES ('SP101', 'PE11', 'RI20', '2020-11-30 00:00', '2020-11-30 12:00')
[2022-11-09 21:48:05] 1 row affected in 35 ms
```

Laporan Tutorial Lab

Basis Data

Semester Ganjil 2022/2023



2. [SQL] Buatlah function/stored procedure dengan nama **check_validity** dan trigger dengan nama **trigger_check_validity** untuk setiap INSERT pada tabel **RAWAT_INAP** untuk memastikan bahwa **tgl_masuk** terjadi sebelum **tgl_keluar** (**tgl_masuk dan tgl_keluar pada hari yang sama juga tidak boleh**). Berikan exception message seperti berikut 'Input tidak valid pastikan bahwa tanggal masuk sebelum tanggal keluar' atau disesuaikan dengan kreativitas kalian tetapi masih dalam pengertian yang sesuai.

Jawaban:

```
eduardus.tjitrahardja.siwonap> -- 2
CREATE OR REPLACE FUNCTION siwanap.check_validity()
    RETURNS trigger AS
$$
BEGIN
    IF (NEW.tgl_masuk >= NEW.tgl_keluar) THEN
        RAISE EXCEPTION 'Input tidak valid pastikan bahwa tanggal masuk sebelum tanggal keluar';
    END IF;
    RETURN NEW;
END;
$$
LANGUAGE plpgsql
[2022-11-09 22:27:58] completed in 60 ms
eduardus.tjitrahardja.siwonap> CREATE TRIGGER trigger_check_validity
    BEFORE INSERT
    ON rawat_inap
    FOR EACH ROW
    EXECUTE PROCEDURE check_validity()
[2022-11-09 22:27:58] completed in 53 ms

eduardus.tjitrahardja.siwonap> INSERT INTO RAWAT_INAP
    VALUES ('RI51', 'KA01', 'PA03', '2022-11-06', '2022-11-08')
[2022-11-09 22:25:44] 1 row affected in 100 ms
eduardus.tjitrahardja.siwonap> INSERT INTO RAWAT_INAP
    VALUES ('RI52', 'KA05', 'PA18', '2022-11-10', '2022-11-08')
[2022-11-09 22:25:46] [P0001] ERROR: Input tidak valid pastikan bahwa tanggal masuk sebelum tanggal keluar
eduardus.tjitrahardja.siwonap> INSERT INTO RAWAT_INAP
    VALUES ('RI53', 'KA01', 'PA38', '2022-11-11', '2022-11-11')
[2022-11-09 22:25:50] [P0001] ERROR: Input tidak valid pastikan bahwa tanggal masuk sebelum tanggal keluar
```

Laporan Tutorial Lab

Basis Data

Semester Ganjil 2022/2023



UNIVERSITAS
INDONESIA

Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

```
eduardus.tjitrahardja=> \df
```

Schema	Name	Result data type	Argument data types	Type
siwanap	cek_jumlah_shift	trigger		trigger
siwanap	check_validity	trigger		trigger
siwanap	diskon_semua_harga	void		func

(3 rows)

```
eduardus.tjitrahardja=> \d RAWAT_INAP
```

Column	Type	Collation	Nullable	Default
id_rawat_inap	character varying(10)		not null	
id_kamar	character varying(10)		not null	
id_pasien	character varying(10)		not null	
tgl_masuk	date		not null	
tgl_keluar	date			

Indexes:

"rawat_inap_pkey" PRIMARY KEY, btree (id_rawat_inap)

Foreign-key constraints:

"rawat_inap_id_kamar_fkey" FOREIGN KEY (id_kamar) REFERENCES kamar(id_kamar) ON DELETE CASCADE

"rawat_inap_id_pasien_fkey" FOREIGN KEY (id_pasien) REFERENCES pasien(id_pasien) ON DELETE CASCADE

Referenced by:

TABLE "dokter_rawat_inap" CONSTRAINT "dokter_rawat_inap_id_rawat_inap_fkey" FOREIGN KEY (id_rawat_inap) REFERENCES rawat_inap(id_rawat_inap) ON DELETE CASCADE

TABLE "shift_perawat" CONSTRAINT "shift_perawat_id_rawat_inap_fkey" FOREIGN KEY (id_rawat_inap) REFERENCES rawat_inap(id_rawat_inap) ON DELETE CASCADE

Triggers:

trigger_check_validity BEFORE INSERT ON rawat_inap FOR EACH ROW EXECUTE PROCEDURE check_validity()

3. [SQL] Terlebih dahulu lakukan penambahan kolom pada tabel **RAWAT_INAP** dengan nama **jml_biaya** dengan tipe integer dan nilai default = 0.

Buatlah function/stored procedure dengan nama **calculate_cost** dan trigger dengan nama **trigger_calculate_cost** untuk setiap INSERT dan UPDATE pada tabel **RAWAT_INAP**. Function bertujuan untuk menghitung **jml_biaya** yang perlu dibayarkan oleh pasien untuk rawat inap. Perhitungan matematisnya: Jumlah malam dirawat * harga kamar (kolom **harga** pada tabel **KAMAR**). Contoh apabila seorang pasien menginap pada kamar KA02 dengan harga 137700 dan dia dirawat dari 2022-11-06 dan keluar pada 2022-11-08 (2 malam) maka jumlah biayanya adalah $2 * 137700 = 275400$. Perlu diperhatikan bahwa kolom **tgl_keluar** pada tabel **RAWAT_INAP** bisa kosong (null), untuk kasus ini maka **calculate_cost** tidak akan dijalankan (Hint: gunakan if pada function untuk menhandle ini).

Jawaban:

Laporan Tutorial Lab

Basis Data

Semester Ganjil 2022/2023



```
eduardus.tjitrahardja.siwanap> ALTER TABLE rawat_inap
                                ADD jml_biaya INTEGER
                                DEFAULT (0)

[2022-11-09 23:10:59] completed in 248 ms
eduardus.tjitrahardja.siwanap> CREATE OR REPLACE FUNCTION siwanap.calculate_cost()
                                RETURNS trigger AS
                                $$
                                DECLARE
                                    jml_biaya INTEGER;
                                BEGIN
                                    IF (NEW.tgl_keluar IS NOT NULL) THEN
                                        jml_biaya := (NEW.tgl_keluar - NEW.tgl_masuk) *
                                                    (SELECT k.harga
                                                     FROM kamar k
                                                     WHERE k.id_kamar = NEW.id_kamar);

                                        NEW.jml_biaya := jml_biaya;

                                    RETURN NEW;
                                END IF;
                                END;
                                $$

                                LANGUAGE plpgsql

[2022-11-09 23:10:59] completed in 29 ms
eduardus.tjitrahardja.siwanap> CREATE TRIGGER trigger_calculate_cost
                                BEFORE INSERT OR UPDATE
                                ON rawat_inap
                                FOR EACH ROW
                                EXECUTE PROCEDURE calculate_cost()

[2022-11-09 23:10:59] completed in 64 ms
```

```
eduardus.tjitrahardja.siwanap> INSERT INTO RAWAT_INAP
                                VALUES ('RI52', 'KA05', 'PA18', '2022-11-10', '2022-11-12')

[2022-11-09 23:12:51] 1 row affected in 45 ms
```

```
eduardus.tjitrahardja.siwanap> SELECT *
                                FROM RAWAT_INAP
                                WHERE id_rawat_inap = 'RI52'

[2022-11-09 23:21:37] 1 row retrieved starting from 1 in 176 ms (execution: 28 ms, fetching: 148 ms)
```

1	id_rawat_inap	id_kamar	id_pasien	tgl_masuk	tgl_keluar	jml_biaya
1	RI52	KA05	PA18	2022-11-10	2022-11-12	275400

Laporan Tutorial Lab

Basis Data

Semester Ganjil 2022/2023



UNIVERSITAS
INDONESIA

Veritas, Probatum, Justitia

FACULTY OF
COMPUTER
SCIENCE

```
eduardus.tjitrahardja=> \df
```

List of functions				
Schema	Name	Result data type	Argument data types	Type
siwanap	calculate_cost	trigger		trigger
siwanap	cek_jumlah_shift	trigger		trigger
siwanap	check_validity	trigger		trigger
siwanap	diskon_semua_harga	void		func
(4 rows)				

```
eduardus.tjitrahardja=> \d RAWAT_INAP
```

Table "siwanap.rawat_inap"				
Column	Type	Collation	Nullable	Default
id_rawat_inap	character varying(10)		not null	
id_kamar	character varying(10)		not null	
id_pasien	character varying(10)		not null	
tgl_masuk	date		not null	
tgl_keluar	date		not null	
jml_biaya	integer			0
Indexes:				
"rawat_inap_pkey" PRIMARY KEY, btree (id_rawat_inap)				
Foreign-key constraints:				
"rawat_inap_id_kamar_fkey" FOREIGN KEY (id_kamar) REFERENCES kamar(id_kamar) ON DELETE CASCADE				
"rawat_inap_id_pasien_fkey" FOREIGN KEY (id_pasien) REFERENCES pasien(id_pasien) ON DELETE CASCADE				
Referenced by:				
TABLE "dokter_rawat_inap" CONSTRAINT "dokter_rawat_inap_id_rawat_inap_fkey" FOREIGN KEY (id_rawat_inap) REFERENCES rawat_inap(id_rawat_inap) ON DELETE CASCADE				
TABLE "shift_perawat" CONSTRAINT "shift_perawat_id_rawat_inap_fkey" FOREIGN KEY (id_rawat_inap) REFERENCES rawat_inap(id_rawat_inap) ON DELETE CASCADE				
Triggers:				
trigger_calculate_cost BEFORE INSERT OR UPDATE ON rawat_inap FOR EACH ROW EXECUTE PROCEDURE calculate_cost()				
trigger_check_validity BEFORE INSERT ON rawat_inap FOR EACH ROW EXECUTE PROCEDURE check_validity()				