

**LAPORAN TUTORIAL LAB 3**  
**BASIS DATA**



**ALVARO AUSTIN**

**2106752180**

**KELAS A**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS INDONESIA**

**DEPOK**

**2022/2023**

# Laporan Tutorial Lab 3

## Basis Data

### Semester Ganjil 2022/2023



#### Latihan

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

- Screenshot contoh 1
  - Screenshot membuat function

```
alvaro.austin=> set search_path to siwanap;
SET
alvaro.austin=> CREATE OR REPLACE FUNCTION SIWANAP.diskon_harga(idkamar VARCHAR(10))
alvaro.austin-> RETURNS INTEGER AS
alvaro.austin-> $$
alvaro.austin$>     DECLARE
alvaro.austin$>     harga_awal INTEGER;
alvaro.austin$>     harga_diskon INTEGER;
alvaro.austin$>     BEGIN
alvaro.austin$>     SELECT harga INTO harga_awal
alvaro.austin$>     FROM KAMAR
alvaro.austin$>     WHERE id_kamar = idkamar;
alvaro.austin$>     harga_diskon := (harga_awal*9/10);
alvaro.austin$>     UPDATE KAMAR SET harga = harga_diskon
alvaro.austin$>     WHERE id_kamar = idkamar;
alvaro.austin$>     RETURN harga_diskon;
alvaro.austin$>     END;
alvaro.austin$> $$
alvaro.austin-> LANGUAGE plpgsql;
CREATE FUNCTION
alvaro.austin=>
```

- Screenshot \df

```
alvaro.austin=> \df

                                List of functions
 Schema |      Name      | Result data type | Argument data types | Type
-----+-----+-----+-----+-----
 siwanap | diskon_harga   | integer          | idkamar character varying | func
(1 row)

alvaro.austin=>
```

- Screenshot \df+

# Laporan Tutorial Lab 3

## Basis Data

Semester Ganjil 2022/2023



UNIVERSITAS  
INDONESIA  
*Veritas, Probitas, Justitia*

FACULTY OF  
COMPUTER  
SCIENCE

Schema		Name	Result data type	Argument data types	Type	Volatility	List of functions			
e		Source code	Description				Owner	Security	Access privileges	Language
		siwanap   diskon_harga	integer	idkamar character varying	func	volatile	alvaro.austin	invoker		plpgsql
			+							
		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;	+							
		(1 row)								
		(END)								

- Screenshot contoh 2

**Laporan Tutorial Lab 3**

**Basis Data**

**Semester Ganjil 2022/2023**



```
alvaro.austin=> SELECT HARGA FROM kamar where id_kamar = 'KA01';
harga
-----
170000
(1 row)

alvaro.austin=> SELECT diskon_harga('KA01')
alvaro.austin-> ;
diskon_harga
-----
153000
(1 row)

alvaro.austin=> SELECT HARGA FROM kamar where id_kamar = 'KA01';
harga
-----
153000
(1 row)

alvaro.austin=> █
```

- Screenshot contoh 3

**Laporan Tutorial Lab 3**

**Basis Data**

**Semester Ganjil 2022/2023**



```
alvaro.austin=> SELECT diskon_harga(id_kamar)
alvaro.austin-> FROM KAMAR;
diskon_harga
-----
153000
153000
153000
153000
270000
270000
270000
270000
270000
270000
630000
630000
630000
630000
630000
990000
990000
990000
990000
990000
990000
990000
990000
990000
990000
990000
990000
990000
990000
990000
990000
1440000
1440000
1440000
1440000
1440000
1440000
1440000
1440000
1440000
1440000
1440000
1440000
1440000
1440000
2790000
2790000
2790000
2790000
```

- Screenshot contoh 4
  - Screenshot buat function

## Laporan Tutorial Lab 3

### Basis Data

Semester Ganjil 2022/2023



```
alvaro.austin=> CREATE OR REPLACE FUNCTION diskon_semua_harga()  
alvaro.austin-> RETURNS void AS  
alvaro.austin-> $$  
alvaro.austin$>     DECLARE  
alvaro.austin$>     temp_row RECORD;  
alvaro.austin$>     harga_diskon INTEGER;  
alvaro.austin$>     BEGIN  
alvaro.austin$>     FOR temp_row IN  
alvaro.austin$>         SELECT *  
alvaro.austin$>         FROM KAMAR  
alvaro.austin$>     LOOP  
alvaro.austin$>         harga_diskon := (temp_row.harga*9/10);  
alvaro.austin$>  
alvaro.austin$>         UPDATE KAMAR SET harga = harga_diskon  
alvaro.austin$>         WHERE id_kamar = temp_row.id_kamar;  
alvaro.austin$>     END LOOP;  
alvaro.austin$>     END;  
alvaro.austin$> $$  
alvaro.austin-> LANGUAGE plpgsql;  
CREATE FUNCTION  
alvaro.austin=>
```

- Screenshot pakai function

```
alvaro.austin=> SELECT diskon_semua_harga();  
diskon_semua_harga  
-----  
  
(1 row)  
  
alvaro.austin=>
```

- Screenshot contoh 5

# Laporan Tutorial Lab 3

## Basis Data

Semester Ganjil 2022/2023



```
alvaro.austin=> DROP FUNCTION diskon_harga(idkamar VARCHAR(10));
DROP FUNCTION
alvaro.austin=>
```

- Screenshot contoh 6
  - Screenshot membuat function trigger

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

- Screenshot contoh 7

```
alvaro.austin=> CREATE TRIGGER trigger_cek_jumlah_shift
alvaro.austin-> BEFORE INSERT ON SHIFT_PERAWAT
alvaro.austin-> FOR EACH ROW
alvaro.austin-> EXECUTE PROCEDURE cek_jumlah_shift();
CREATE TRIGGER
alvaro.austin=>
```

- Screenshot contoh 8
  - Screenshot trigger berfungsi

```
alvaro.austin=> INSERT INTO SHIFT_PERAWAT (id_shift_perawat, id_perawat, id_rawat_inap, waktu_mulai, waktu_akhir)
alvaro.austin-> VALUES ('SP101', 'PE13', 'RI20', '2020-11-30 00:00', '2020-11-30 12:00');
ERROR:  Maaf, perawat          tidak boleh memiliki shift melebihi          5
alvaro.austin=>
alvaro.austin=>
```

# Laporan Tutorial Lab 3

## Basis Data

### Semester Ganjil 2022/2023



- Screenshot trigger tidak jalan

```
alvaro.austin=> INSERT INTO SHIFT_PERAWAT (id_shift_perawat, id_perawat, id_rawat_inap, waktu_mulai, waktu_akhir)
alvaro.austin-> VALUES ('SP101', 'PE11', 'RI20', '2020-11-30 00:00', '2020-11-30 12:00');
INSERT 0 1
```

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.

- Create function

```
alvaro.austin=> CREATE OR REPLACE FUNCTION check_validity()
alvaro.austin-> RETURNS trigger AS
alvaro.austin-> $$
alvaro.austin$> BEGIN
alvaro.austin$>   if (TG_OP = 'INSERT') THEN
alvaro.austin$>     IF (NEW.tgl_masuk >= NEW.tgl_keluar) THEN
alvaro.austin$>       RAISE EXCEPTION 'Input tidak valid pastikan bahwa tanggal masuk sebelum tanggal keluar';
alvaro.austin$>     END IF;
alvaro.austin$>     RETURN NEW;
alvaro.austin$>   end if;
alvaro.austin$> END;
alvaro.austin$> $$
alvaro.austin-> LANGUAGE plpgsql;
CREATE FUNCTION
alvaro.austin=> █
```

- \df

```
alvaro.austin=> \df
```

List of functions				
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)

```
alvaro.austin=> █
```



# Laporan Tutorial Lab 3

## Basis Data

### Semester Ganjil 2022/2023



- Create Trigger

```
alvaro.austin=> CREATE TRIGGER trigger_check_validity
alvaro.austin-> BEFORE INSERT ON RAWAT_INAP
alvaro.austin-> FOR EACH ROW EXECUTE PROCEDURE check_validity();
CREATE TRIGGER
alvaro.austin=>
```

- Insert 1

```
alvaro.austin=> INSERT INTO RAWAT_INAP VALUES ('RI51', 'KA01', 'PA03', '2022-11-06', '2022-11-08');
INSERT 0 1
alvaro.austin=>
```

- Insert 2

```
alvaro.austin=> INSERT INTO RAWAT_INAP VALUES ('RI52', 'KA05', 'PA18', '2022-11-10', '2022-11-08');
ERROR:  Input tidak valid pastikan bahwa tanggal masuk sebelum tanggal keluar
alvaro.austin=>
```

- Insert 3

```
alvaro.austin=> INSERT INTO RAWAT_INAP VALUES ('RI53', 'KA01', 'PA38', '2022-11-11', '2022-11-11');
ERROR:  Input tidak valid pastikan bahwa tanggal masuk sebelum tanggal keluar
alvaro.austin=>
```

- \d RAWAT\_INAP

```
alvaro.austin=> \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                  |           |          |
Indexes:
    "rawat_inap_pkey" PRIMARY KEY, btree (id_rawat_inap)
    "index_tgl_keluar_rawat_inap" btree (tgl_keluar)
Foreign-key constraints:
    "rawat_inap_id_kamar_fkey" FOREIGN KEY (id_kamar) REFERENCES kamar(id_kamar)
    "rawat_inap_id_pasien_fkey" FOREIGN KEY (id_pasien) REFERENCES pasien(id_pasien)
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)
    TABLE "shift_perawat" CONSTRAINT "shift_perawat_id_rawat_inap_fkey" FOREIGN KEY (id_rawat_inap) REFERENCES rawat_inap(id_rawat_inap)
Triggers:
    trigger_check_validity BEFORE INSERT ON rawat_inap FOR EACH ROW EXECUTE PROCEDURE check_validity()
alvaro.austin=>
```

**Laporan Tutorial Lab 3**  
**Basis Data**  
**Semester Ganjil 2022/2023**



3. 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 handle ini).

- Alter table

```
alvaro.austin=> ALTER TABLE RAWAT_INAP
alvaro.austin-> ADD COLUMN jml_biaya INTEGER DEFAULT 0;
ALTER TABLE
alvaro.austin=> █
```

- Create function

## Laporan Tutorial Lab 3

### Basis Data

Semester Ganjil 2022/2023



UNIVERSITAS  
INDONESIA  
*Veritas, Probitas, Justitia*

FACULTY OF  
COMPUTER  
SCIENCE

```
alvaro.austin=> CREATE OR REPLACE FUNCTION calculate_cost()
alvaro.austin-> RETURNS TRIGGER AS
alvaro.austin-> $$
alvaro.austin$> DECLARE
alvaro.austin$>     jmlh_biaya INTEGER;
alvaro.austin$>     harga_kamar INTEGER;
alvaro.austin$>     hari_stay INTEGER;
alvaro.austin$> BEGIN
alvaro.austin$>     if(NEW.tgl_keluar is not NULL) THEN
alvaro.austin$>         IF(TG_OP = 'INSERT') THEN
alvaro.austin$>             SELECT harga INTO harga_kamar
alvaro.austin$>             FROM KAMAR
alvaro.austin$>             WHERE KAMAR.id_kamar = NEW.id_kamar;
alvaro.austin$>
alvaro.austin$>             hari_stay := NEW.tgl_keluar - NEW.tgl_masuk;
alvaro.austin$>             jmlh_biaya := hari_stay*harga_kamar;
alvaro.austin$>             NEW.jml_biaya = jmlh_biaya;
alvaro.austin$>
alvaro.austin$>         ELSIF(TG_OP = 'UPDATE') THEN
alvaro.austin$>             SELECT harga INTO harga_kamar
alvaro.austin$>             FROM KAMAR
alvaro.austin$>             WHERE KAMAR.id_kamar = NEW.id_kamar;
alvaro.austin$>
alvaro.austin$>             hari_stay := NEW.tgl_keluar - NEW.tgl_masuk;
alvaro.austin$>             jmlh_biaya := hari_stay*harga_kamar;
alvaro.austin$>             NEW.jml_biaya = jmlh_biaya;
alvaro.austin$>
alvaro.austin$>         END IF;
alvaro.austin$>         RETURN NEW;
alvaro.austin$>     END IF;
alvaro.austin$> END;
alvaro.austin$> $$
alvaro.austin-> LANGUAGE plpgsql;
CREATE FUNCTION
alvaro.austin=>
```

# Laporan Tutorial Lab 3

## Basis Data

### Semester Ganjil 2022/2023



- Create trigger

```
alvaro.austin=> CREATE TRIGGER trigger_calculate_cost
alvaro.austin-> BEFORE INSERT OR UPDATE ON RAWAT_INAP
alvaro.austin-> FOR EACH ROW EXECUTE PROCEDURE calculate_cost();
CREATE TRIGGER
alvaro.austin=>
```

- Insert

```
alvaro.austin=> INSERT INTO RAWAT_INAP VALUES ('RI52', 'KA05', 'PA18', '2022-11-10', '2022-11-12');
INSERT 0 1
alvaro.austin=>
```

- Select

```
alvaro.austin=> SELECT * FROM RAWAT_INAP WHERE id_rawat_inap='RI52';
 id_rawat_inap | id_kamar | id_pasien | tgl_masuk | tgl_keluar | jml_biaya
-----+-----+-----+-----+-----+-----
 RI52          | KA05     | PA18      | 2022-11-10 | 2022-11-12 | 275400
(1 row)

alvaro.austin=>
```

- \df

```
alvaro.austin=> \df

Schema | Name | List of functions
-----+-----+-----
siwanap | calculate_cost | trigger | | trigger
siwanap | cek_jumlah_shift | trigger | | trigger
siwanap | check_validity | trigger | | trigger
siwanap | diskon_semua_harga | void | | func
(4 rows)
```

- \d RAWAT\_INAP

# Laporan Tutorial Lab 3

## Basis Data

Semester Ganjil 2022/2023



UNIVERSITAS  
INDONESIA  
*Veritas, Probitas, Justitia*

FACULTY OF  
COMPUTER  
SCIENCE

```
alvaro.austin=> \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            |           |           |
jml_biaya     | integer         |           |           | 0
Indexes:
    "rawat_inap_pkey" PRIMARY KEY, btree (id_rawat_inap)
    "index_tgl_keluar_rawat_inap" btree (tgl_keluar)
Foreign-key constraints:
    "rawat_inap_id_kamar_fkey" FOREIGN KEY (id_kamar) REFERENCES kamar(id_kamar)
    "rawat_inap_id_pasien_fkey" FOREIGN KEY (id_pasien) REFERENCES pasien(id_pasien)
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)
    TABLE "shift_perawat" CONSTRAINT "shift_perawat_id_rawat_inap_fkey" FOREIGN KEY (id_rawat_inap) REFERENCES rawat_inap(id_rawat_inap)
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()
alvaro.austin=>
```