



**Program Studi : Teknik Informatika**

**Laporan Praktikum : Basis Data 2**

**Praktikum 4**

Muhammad Azhar Rasyad  
0110217029

**STT Terpadu Nurul Fikri  
Tahun 2018**

## Prosedur Tersimpan – Fungsi 2

### Tugas Pendahuluan

1. Jelaskan fungsi dari prosedur tersimpan !

**Prosedur tersimpan** adalah metode untuk mengenkapsulasi tugas secara berulang. Sehingga memungkinkan kita untuk mendeklarasikan variabel, mengontrol aliran data dan teknik pemrograman yang berguna lainnya.

2. Jelaskan keuntungan dan kerugian dari prosedur tersimpan !

### Kelebihan Prosedur Tersimpan :

- Berbagi logic dengan aplikasi lainnya. Prosedur tersimpan merangkum fungsionalitas untuk memastikan bahwa akses data dan manipulasi koheren antara aplikasi yang berbeda.
- Mengisolasi pengguna dari tabel data. Fitur ini memberi Anda kemampuan untuk memberikan akses ke prosedur tersimpan yang memanipulasi data namun tidak secara langsung ke tabel.
- Berikan mekanisme keamanan. Jika Anda hanya dapat mengakses data menggunakan prosedur tersimpan yang telah ditentukan, tidak ada orang lain yang dapat menjalankan pernyataan SQL DELETE dan menghapus data Anda.
- Untuk meningkatkan kinerja karena mengurangi lalu lintas jaringan. Dengan Prosedur tersimpan, beberapa panggilan bisa disatukan menjadi satu.

### Kekurangan Prosedur Tersimpan :

- Peningkatan beban pada database server. Sebagian besar pekerjaan dilakukan di sisi server, dan kurang pada sisi client.
- Anda harus mempelajari sintaks pernyataan SQL untuk menulis prosedur tersimpan.
- Jika Anda mengulangi logika aplikasi Anda di dua tempat yang berbeda: kode bahasa pemrograman dan kode prosedur tersimpan, membuat pemeliharaan aplikasi akan menjadi lebih sulit.
- Migrasi ke sistem manajemen basis data yang berbeda (DB2, SQL Server, dll) berpotensi menjadi lebih sulit.

3. Pastikan environemnt PostgreSQL anda telah berjalan dengan semestinya (praktikum 3) !

```
apsql@mazharrasyad: ~  
mzharrasyad@mzharrasyad:~$ sudo su - apsql  
[sudo] password for mzharrasyad:  
apsql@mzharrasyad:~$ /home/apsql/pg105/bin/pg_ctl -D /home/apsql/datapg/ -l log  
file start  
waiting for server to start..... done  
server started  
apsql@mzharrasyad:~$
```

```
apsql@mzharrasyad: ~  
apsql@mzharrasyad:~$ netstat -tanp | grep :5555  
(Not all processes could be identified, non-owned process info  
will not be shown, you would have to be root to see it all.)  
tcp        0      0 0.0.0.0:5555          0.0.0.0:*            LISTEN  
11534/postgres  
tcp6       0      0 :::5555             :::*                  LISTEN  
11534/postgres  
apsql@mzharrasyad:~$
```

## Percobaan 1 : Skema Table

1. Buat database dbsdm pada server PostgreSQL anda !!

```
apsql@mazharrasyad: ~  
apsql@mazharrasyad:~$ /home/apsql/pg105/bin/createdb dbsdm -U apsql -p5555 -h localhost  
Password:  
apsql@mazharrasyad:~$
```

```
apsql@mazharrasyad: ~  
apsql@mazharrasyad:~$ /home/apsql/pg105/bin/psql dbsdm -U apsql -p5555 -h localhost  
Password for user apsql:  
psql (10.5)  
Type "help" for help.  
dbsdm=#
```

2. Buat table departemen dengan data berikut ini:

```
GNU nano 2.5.3      File: /tmp/psql.edit.4280.sql      Modified  
  
create table departemen(  
iddep serial primary key,  
nama varchar(50) not null  
);  
  
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos  
^X Exit      ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

```
apsql@mazharrasyad: ~
GNU nano 2.5.3      File: /tmp/psql.edit.4280.sql      Modified

insert into departemen values
(1, 'SDM'),
(2, 'Keuangan'),
(3, 'Sales');

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

```
apsql@mazharrasyad: ~
dbstdm=# select * from departemen;
 iddep |  nama
-----+-----
      1 |  SDM
      2 | Keuangan
      3 |  Sales
(3 rows)

dbstdm=#
```

3. Buat table pegawai dengan data berikut ini: perhatikan pada table pegawai terdapat field iddep yang merupakan foreign key ke tabel departemen

```
apsql@mazharrasyad: ~
GNU nano 2.5.3      File: /tmp/psql.edit.4280.sql      Modified

create table pegawai(
nip serial primary key,
nama varchar(50) not null,
tgl_lahir date not null,
gaji integer not null,
iddep integer references departemen(iddep)
);

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

```
apsql@mazharrasyad: ~
GNU nano 2.5.3      File: /tmp/psql.edit.4280.sql      Modified

insert into pegawai values
(20141, 'Rosalie Naurah', '20030308', 3000000, 1),
(20142, 'Rawa Wulan', '20020414', 2400000, 1),
(20143, 'Glagah Putih', '20010910', 4000000, 2),
(20144, 'Sambungsari', '20040918', 3000000, 2),
(20145, 'Agung Sedayu', '20010220', 2500000, 3);

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

```
apsql@mazharrasyad: ~
dbsdm=# select * from pegawai;
  nip |      nama      | tgl_lahir | gaji  | iddep
-----+-----+-----+-----+-----
 20141 | Rosalie Naurah | 2003-03-08 | 3000000 |      1
 20142 | Rawa Wulan    | 2002-04-14 | 2400000 |      1
 20143 | Glagah Putih  | 2001-09-10 | 4000000 |      2
 20144 | Sambungsari   | 2004-09-18 | 3000000 |      2
 20145 | Agung Sedayu  | 2001-02-20 | 2500000 |      3
(5 rows)

dbsdm=#
```

4. Buat table gaji\_departemen yang menyimpan informasi total budget gaji untuk setiap departemen dengan perintah query berikut:

```
apsql@mazharrasyad: ~
GNU nano 2.5.3      File: /tmp/psql.edit.4280.sql      Modified

create table gaji_departemen as
select iddep, 0 as total_gaji from departemen;
```

```
apsql@mazharrasyad: ~  
dbstdm=# select * from gaji_departemen;  
iddep | total_gaji  
-----+-----  
1 | 0  
2 | 0  
3 | 0  
(3 rows)  
  
dbstdm=#
```

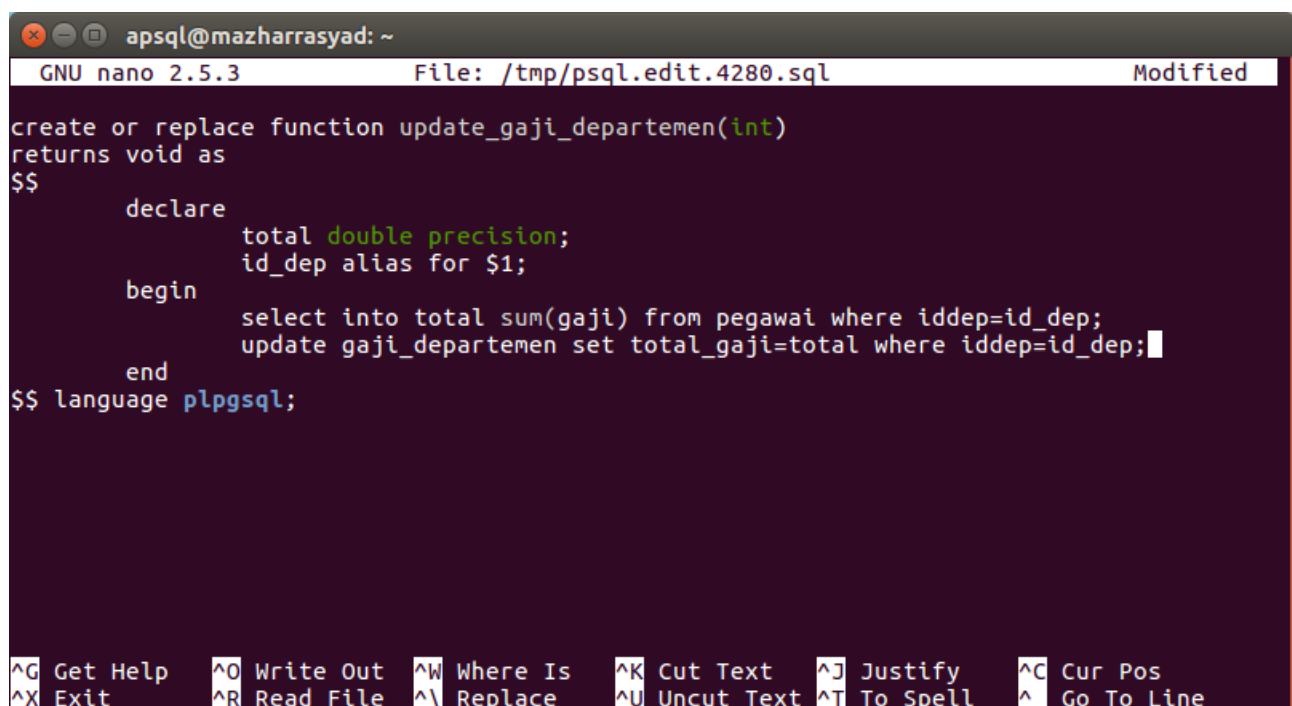
## Membuat Fungsi Update Table gaji\_departemen

1. Fungsi ini digunakan untuk mendapatkan budget gaji yang dibutuhkan oleh sebuah departemen. Fungsi memiliki argumen iddepartemen.

**Penjelasan return void**, Fungsi yang void sering disebut juga prosedur. Disebut void karena fungsi tersebut **tidak mengembalikan suatu nilai keluaran yang didapat dari hasil proses fungsi tersebut**.

### Ciri-Ciri return void :

- Tidak adanya keyword return.
- Tidak adanya tipe data di dalam deklarasi fungsi.
- Menggunakan keyword void.
- Tidak dapat langsung ditampilkan hasilnya.
- Tidak memiliki nilai kembalian fungsi.



```
apsql@mazharrasyad: ~
GNU nano 2.5.3 File: /tmp/psql.edit.4280.sql Modified
create or replace function update_gaji_departemen(int)
returns void as
$$
    declare
        total double precision;
        id_dep alias for $1;
    begin
        select into total sum(gaji) from pegawai where iddep=id_dep;
        update gaji_departemen set total_gaji=total where iddep=id_dep;
    end
$$ language plpgsql;
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^\_ Go To Line



## 2. Melihat informasi fungsi update\_gaji\_departemen

```
apsql@mazharrasyad: ~
dbstdm=# \df

```

		List of functions		
Schema	Name	Result data type	Argument data types	Type
public	update_gaji_departemen	void	integer	normal

```
(1 row)
dbstdm=#
```

## 3. Memanggil fungsi

```
apsql@mazharrasyad: ~
dbstdm=# select update_gaji_departemen(1);
update_gaji_departemen
-----
(1 row)

dbstdm=# select update_gaji_departemen(2);
update_gaji_departemen
-----
(1 row)

dbstdm=# select update_gaji_departemen(3);
update_gaji_departemen
-----
(1 row)

dbstdm=# select update_gaji_departemen(4);
update_gaji_departemen
-----
(1 row)

dbstdm=#
```

## 4. Cek output tabel gaji\_departemen

```
apsql@mazharrasyad: ~
dbstdm=# select * from gaji_departemen;
iddep | total_gaji
-----+-----
1 | 5400000
2 | 7000000
3 | 2500000
(3 rows)

dbstdm=#
```

## Praktikum Mandiri

1. Tambahkan filed jabatan varchar(30) pada table pegawai

```
apsql@mazharrasyad: ~
dbstdm=# alter table pegawai add jabatan varchar(30);
ALTER TABLE
dbstdm=# \d pegawai

Table "public.pegawai"
Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
nip    | integer               |           | not null | nextval('pegawai_ni
p_seq'::regclass)
nama   | character varying(50) |           | not null |
tgl_lahir | date                  |           | not null |
gaji   | integer               |           | not null |
iddep  | integer               |           |          |
jabatan | character varying(30) |           |          |
Indexes:
    "pegawai_pkey" PRIMARY KEY, btree (nip)
Foreign-key constraints:
    "pegawai_iddep_fkey" FOREIGN KEY (iddep) REFERENCES departemen(iddep)

dbstdm=#
```

2. Buat fungsi update\_jabatan\_pegawai dengan argumen NIP pegawai yang otomatis mengupdate jabatan pegawai dengan ketentuan :

Gaji >= 3000000 sebagai 'SUPERVISOR' dan selainnya adalah 'STAFF'

```
GNU nano 2.5.3      File: /tmp/psql.edit.8577.sql      Modified
create or replace function update_jabatan_pegawai(int)
returns void as
$$
    declare
        total double precision;
        vnip alias for $1;
        vjabatan text;
    begin
        select into total gaji from pegawai where nip=vnip;

        if total >= 3000000 then
            vjabatan = 'SUPERVISOR';
        else
            vjabatan = 'STAFF';
        end if;

        update pegawai set jabatan = vjabatan where nip=vnip;
    end
$$ language plpgsql;

^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify
^X Exit          ^R Read File     ^\ Replace       ^U Uncut Text    ^T To Spell
```

### 3. Hasil akhir table pegawai

```
apsql@mazharrasyad: ~
dbstdm=# select update_jabatan_pegawai('20141');
update_jabatan_pegawai
-----
(1 row)

dbstdm=# select update_jabatan_pegawai('20142');
update_jabatan_pegawai
-----
(1 row)

dbstdm=# select update_jabatan_pegawai('20143');
update_jabatan_pegawai
-----
(1 row)

dbstdm=# select update_jabatan_pegawai('20144');
update_jabatan_pegawai
-----
(1 row)

dbstdm=# select update_jabatan_pegawai('20145');
update_jabatan_pegawai
-----
(1 row)

dbstdm=#
```

```
apsql@mazharrasyad: ~
dbstdm=# select * from pegawai order by jabatan desc;
nip |      nama      | tgl_lahir | gaji | iddep | jabatan
-----+-----+-----+-----+-----+-----
20141 | Rosalie Naurah | 2003-03-08 | 3000000 | 1 | SUPERVISOR
20143 | Glagah Putih  | 2001-09-10 | 4000000 | 2 | SUPERVISOR
20144 | Sambungsari   | 2004-09-18 | 3000000 | 2 | SUPERVISOR
20142 | Rawa Wulan    | 2002-04-14 | 2400000 | 1 | STAFF
20145 | Agung Sedayu  | 2001-02-20 | 2500000 | 3 | STAFF
(5 rows)

dbstdm=#
```

----- Selesai -----

## Referensi

- <https://www.yudana.id/mengenal-stored-procedures-di-database-mysql/>
- <http://yuliedwia.blogspot.com/2016/05/fungsi-void-non-void-borland-c.html>