

# **Laporan Praktikum Prosedur Tersimpan Database PostgreSQL**



Ahmad Fathan Syakir  
0110217055  
Teknik Informatika 1

Sekolah Tinggi Teknologi Terpadu Nurul Fikiri  
2018

## Percobaan 1 : Membuat Database baru

Cara membuat database baru ini sama dengan cara membuat database yang sebelumnya, yang membedakan adalah pada pembuatan database ini tidak menggunakan user baru(siswa1), melainkan menggunakan user linux yang dibuat sebelumnya(fatan). Setelah membuat database baru lalu masuk kedalam database tersebut.

```
ahmadfathan@X441U:~$ sudo su - fatan
[sudo] password for ahmadfathan:
```

```
fatat@X441U:~$ /home/fatan/pg965/bin/createdb dbpelatihan2 -U
fatat -p5555 -h localhost
password :
```

```
fatat@X441U:~$ /home/fatan/pg965/bin/pg_ctl -D /home/fatan/datapg/
-l logfile start
server starting
```

```
fatat@X441U:~$ /home/fatan/pg965/bin/psql dbpelatihan2 -U fatat
-p5555 -h localhost
Password for user fatat:
psql (9.6.5)
Type "help" for help.
```

```
dbpelatihan2=#
```

Jelaskan Pengertian dari table ini

```
dbpelatihan2=# select * from pg_language;
lanname | lanowner | lanispl | lanpltrusted | lanplcallfoid | laninline | lanvalidator | lanacl
-----+-----+-----+-----+-----+-----+-----+-----
internal |      10 | f       | f             |              0 |          0 |          2246 |
c       |      10 | f       | f             |              0 |          0 |          2247 |
sql     |      10 | f       | t             |              0 |          0 |          2248 |
plpgsql |      10 | t       | t             |         12428 |        12429 |          12430 |
(4 rows)
```

**lanname** : nama language yang support dalam prosedur tersimpan  
**internal** : bahasa internal  
**c** : fungsi yang ditentukan pengguna dapat ditulis dalam C (bahasa yang dapat dibuat kompetibel dengan C, seperti C++)  
**sql** : bahasa sql  
**plpgsql** : merupakan bahasa lain untuk membuat sebuah fungsi

**lanowner** : pemilik language  
**10** : memiliki id bernomor 10

**lanispl** : untuk menunjukan nilai false jika language internal seperti SQL, dan bernilai benar untuk language yang ditentukan oleh user  
**t** : untuk menunjukan nilai true, menunjukan sedang dipakai

**f** : untuk menunjukkan nilai false, menunjukkan sedang tidak pakai

**lanpltrusted** : untuk menampilkan nilai true jika language terpercaya, yang berarti diyakini tidak memberikan akses apapun keluar lingkungan eksekusi SQL.

**t** : untuk menunjukkan nilai true, menunjukkan sedang dipakai

**f** : untuk menunjukkan nilai false, menunjukkan sedang tidak pakai

**Lanplcallfoid** : untuk non-internal language yang menunjuk pada penanganan language

**0** :

**12428** :

**laninline** : ini mereferensikan fungsi yang bertanggung jawab untuk mengeksekusi blok kode anonim(blok) anonim(blok DO). Nol jika blok inline tidak didukung

**0** : tidak ada blok inline yang didukung

**12428** : untuk menunjukan blok inline yang didukung

**lanvalidator** : referensi ini meruoakan fungsi validator language yang bertanggung jawab untuk memeriksa sintaks dan validitas fungsi-fungsi baru ketika mereka dibuat. Nol jika tidak ada validator yang disediakan

**2246** : untuk menunjukan jumlah validator yang disediakan

**2247** : untuk menunjukan jumlah validator yang disediakan

**2248** : untuk menunjukan jumlah validator yang disediakan

**12430** : untuk menunjukan jumlah validator yang disediakan

**lanacl** : untuk menunjukan hak akses istimewa

## Percobaan 2 : Membuat Function

Membuat function dengan menggunakan perintah

```
create or replace function nama_function() returns tipe_data
as
$$
declare
nama_variable tipe_data;
begin
nama_variable = isi_yang_ingin_ditampilkan;
return nama_variable;
end;
$$
language plpgsql;
```

**Note :** "create or replace" jika tidak menggunakan tambahan "replace" maka apabila melakukan perubahan pada function tersebut akan gagal/tidak bisa dilakukan.

### Contoh Function biasa

#### 1. Function Salam()

```
dbpelatihan2=# create or replace function salam() returns text as
dbpelatihan2-# $$
dbpelatihan2$# declare
dbpelatihan2$# pesan text;
dbpelatihan2$# begin
dbpelatihan2$# pesan = 'Welcome to campus 1 juta bisa kuliah';
dbpelatihan2$# return pesan;
dbpelatihan2$# end;
dbpelatihan2$# $$
dbpelatihan2-# language plpgsql;
CREATE FUNCTION
```

untuk menjalankan function jalankan perintah `select nama_function()`

```
dbpelatihan2=# select salam();
salam
-----
Welcome to campus 1 juta bisa kuliah
(1 row)
```

untuk melihat isi function yang sudah dibuat dapat mengetikan perintah `\df nama_function;`

```
dbpelatihan2=# \df salam;
List of functions
 Schema | Name   | Result data type | Argument data types | Type
-----+-----+-----+-----+-----
 public | salam | text              |                      | normal
(1 row)
```

## 2. Function Angka()

```
dbpelatihan2=# create or replace function angka() returns integer as
dbpelatihan2-# $$
dbpelatihan2$# declare
dbpelatihan2$# variable integer;
dbpelatihan2$# begin
dbpelatihan2$# variable = 10;
dbpelatihan2$# return variable;
dbpelatihan2$# end;
dbpelatihan2$# $$
dbpelatihan2-# language plpgsql;
CREATE FUNCTION
```

untuk menjalankan function jalankan perintah `select nama_ function()`

```
dbpelatihan2=# select angka();
angka
-----
      10
(1 row)
```

untuk melihat isi function yang sudah dibuat dapat mengetikkan perintah `\df nama_function;`

```
dbpelatihan2=# \df angka;
List of functions
 Schema | Name  | Result data type | Argument data types | Type
-----+-----+-----+-----+-----
 public | angka | integer          |                     | normal
(1 row)

dbpelatihan2=#
```

## Contoh Function untuk menghitung

### 1. Function Menghitung Luas Segituga

```
dbpelatihan2=# create or replace function luas_segitiga(int,int) returns
double precision as
dbpelatihan2-# $$
dbpelatihan2$# declare
dbpelatihan2$# luas double precision;
dbpelatihan2$# begin
dbpelatihan2$# luas = 0.5 * $1 * $2;
dbpelatihan2$# return luas;
dbpelatihan2$# end;
dbpelatihan2$# $$
dbpelatihan2-# language plpgsql;
CREATE FUNCTION
```

untuk menjalankan function jalankan perintah `select nama_function(masukan_angka_yang_akan_dihitung)`

```
dbpelatihan2=# select luas_segitiga(4,7);
luas_segitiga
-----
14
(1 row)
```

untuk melihat isi function yang sudah dibuat dapat mengetikan perintah `\df nama_function;`

```
dbpelatihan2=# \df luas_segitiga;
List of functions
Schema | Name | Result data type | Argument data types | Type
-----+-----+-----+-----+-----
public | luas_segitiga | double precision | integer, integer | normal
(1 row)
```

## 2. Function Menghitung Luas Trapesium

```
dbpelatihan2=# create or replace function luas_trapesium(int,int,int)
returns double precision as
dbpelatihan2-# $$
dbpelatihan2$# declare
dbpelatihan2$# luas double precision;
dbpelatihan2$# begin
dbpelatihan2$# luas = ($1 + $2) * $3 / 2;
dbpelatihan2$# return luas;
dbpelatihan2$# end;
dbpelatihan2$# $$
dbpelatihan2-# language plpgsql;
CREATE FUNCTION
```

untuk menjalankan function jalankan perintah `select nama_function(masukan_angka_yang_akan_dihitung)`

```
dbpelatihan2=# select luas_trapesium(3,1,3);
luas_trapesium
-----
6
(1 row)
```

untuk melihat isi function yang sudah dibuat dapat mengetikan perintah `\df nama_function;`

```
dbpelatihan2=# \df luas_trapesium;
List of functions
Schema | Name | Result data type | Argument data types | Type
-----+-----+-----+-----+-----
public | luas_trapesium | double precision | integer, integer, integer | normal
(1 row)
```

**Note :** "(int,int)" ini digunakan untuk memasukan angka  
"\$" digunakan untuk menunjuk angka yang dimasukan

## Contoh Function Dengan Logika IF Else

### 1. Function Hasil Ujian()

```
dbpelatihan2=# create or replace function hasil_ujian(double precision)
returns text as
dbpelatihan2-# $$
dbpelatihan2$# declare
dbpelatihan2$# nilai alias for $1;
dbpelatihan2$# begin
dbpelatihan2$# if nilai > 55 then return 'LULUS';
dbpelatihan2$# else return 'TIDAK LULUS';
dbpelatihan2$# end if;
dbpelatihan2$# end;
dbpelatihan2$# $$
dbpelatihan2-# language plpgsql;
CREATE FUNCTION
```

untuk menjalankan function jalankan perintah `select nama_`  
`function(masukan_angka/nilai)`

```
dbpelatihan2=# select hasil_ujian(55);
 hasil_ujian
-----
TIDAK LULUS
(1 row)
```

```
dbpelatihan2=# select hasil_ujian(55.1);
 hasil_ujian
-----
LULUS
(1 row)
```

### 2. Function Grade()

Range Nilai	Grade
0 s/d 30	E
>30 s/d 55	D
>55 s/d 69	C
>69 s/d 84	B
>84 s/d 100	A

```
dbpelatihan2=# create or replace function grade(double precision) returns
text as
dbpelatihan2-# $$
dbpelatihan2$# declare
dbpelatihan2$# nilai alias for $1;
dbpelatihan2$# begin
dbpelatihan2$# if nilai > 100 then return 'N/A';
dbpelatihan2$# elseif nilai > 84 then return 'A';
dbpelatihan2$# elseif nilai > 69 then return 'B';
dbpelatihan2$# elseif nilai > 55 then return 'C';
dbpelatihan2$# elseif nilai > 30 then return 'D';
dbpelatihan2$# elseif nilai >= 0 then return 'E';
dbpelatihan2$# end if;
dbpelatihan2$# end;
dbpelatihan2$# $$
dbpelatihan2-# language plpgsql;
```

untuk menjalankan function jalankan perintah `select nama_`  
`function(masukan_angka/nilai)`

```
dbpelatihan2=# select grade(100);
grade
-----
A
(1 row)
```

```
dbpelatihan2=# select grade(100.1);
grade
-----
N/A
(1 row)
```

```
dbpelatihan2=# select grade(85);
grade
-----
A
(1 row)
```

```
dbpelatihan2=# select grade(84);
grade
-----
B
(1 row)
```

```
dbpelatihan2=# select grade(50);
grade
-----
D
(1 row)
```

```
dbpelatihan2=# select grade(0);
grade
-----
E
(1 row)
```



```
dbpelatihan2=# select grade(-1);
ERROR:  control reached end of function without RETURN
CONTEXT:  PL/pgSQL function grade(double precision)
dbpelatihan2=#
```

**Note :** apabila memasukan nilai yang melewati batas yang ditentukan akan menampilkan pesan error atau N/A

untuk melihat semua function yang sudah dibuat didatabase dapat mengetikan perintah **\df**

```
dbpelatihan2=# \df
```

List of functions				
Schema	Name	Result data type	Argument data types	Type
public	angka	integer		normal
public	grade	text	double precision	normal
public	hasil_ujian	text	double precision	normal
public	luas_segitiga	double precision	integer, integer	normal
public	luas_trapesium	double precision	integer, integer, integer	normal
public	salam	text		normal

(6 rows)