

Store Procedure

Gold - Chapter 4 - Topic 4

Selamat datang di **Chapter 4 Topic 4**
online course **Back End Java** dari
Binar Academy!

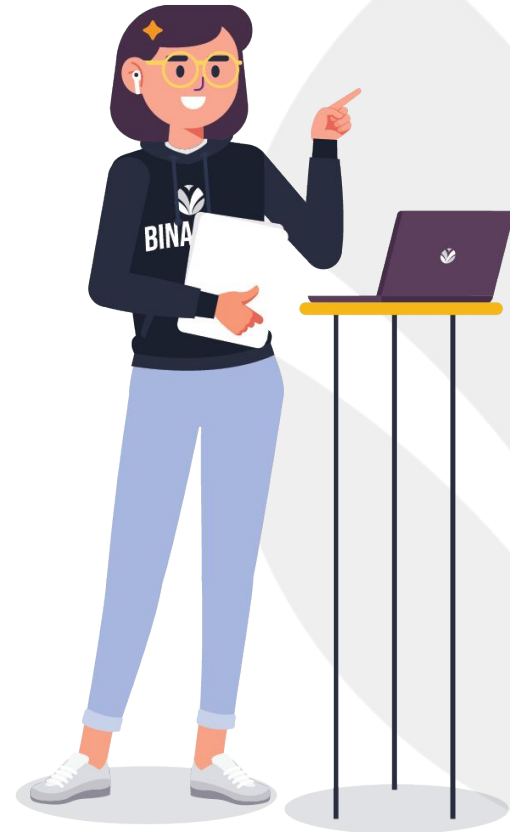


Belajar lagi, progress lagi! ✨

Binarian, kamu keren banget udah sampai di chapter ini. Setelah Database operation udah beres sekarang kita move on ke Store Procedure.

Berhubung istilah ini baru kita kenal, gimana kalau kita mulai materi dengan kenalan sama konsep Store Procedure dulu?

Penasaran mau belajar apa aja? Yaudah yuk, kita buka tirai selanjutnya □



Dari sesi ini, kita bakal bahas hal-hal berikut:

- Konsep Store Procedure
- Hands on Store Procedure pada PostgreSQL



Dalam pembuatan sebuah database tentu aja kita pernah menemui sebuah kesalahan data.

Yepp, bahasan **Store Procedure** hari ini ada kaitannya dengan solusi untuk mengatasi kesalahan itu.

Kita coba jelasin setelah ini, ya.



Aku siap menghadapi masalah database dengan Store Procedure ☐☐♀

Tadi udah disebutin, bahwa dalam pembuatan sebuah database bisa terjadi kesalahan dalam memuat sebuah data yang disediakan di dalam database.

Nggak cuma itu, terkadang di dalam sebuah database data yang kita panggil menjadi tidak teratur.

Bahkan kita membutuhkan penggunaan memori yang besar karena adanya query database yang berulang-ulang.



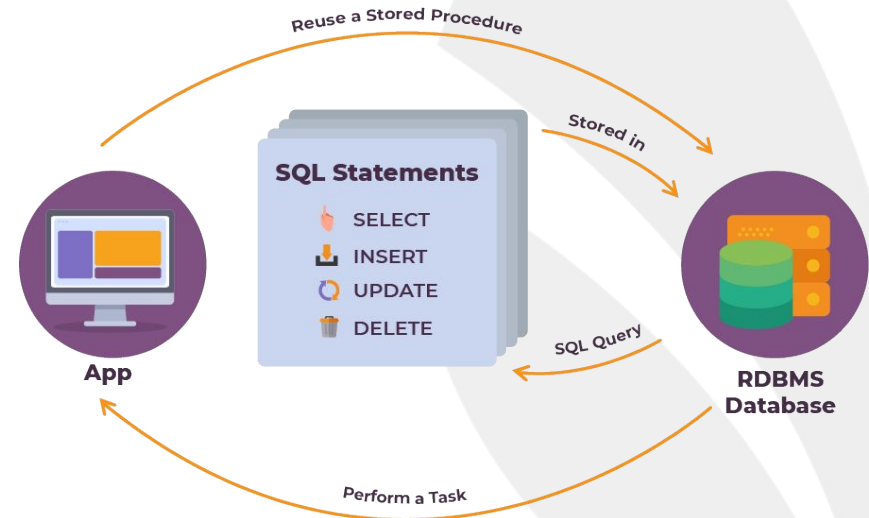
STORED PROCEDURES

Store Procedure hadir merubah segalanya~

Masalah-masalah tadi itu membutuhkan store procedure untuk menyelesaikannya.

Gimana enggak? Store procedure ini merupakan kumpulan pernyataan/procedure SQL yang disimpan dalam database.

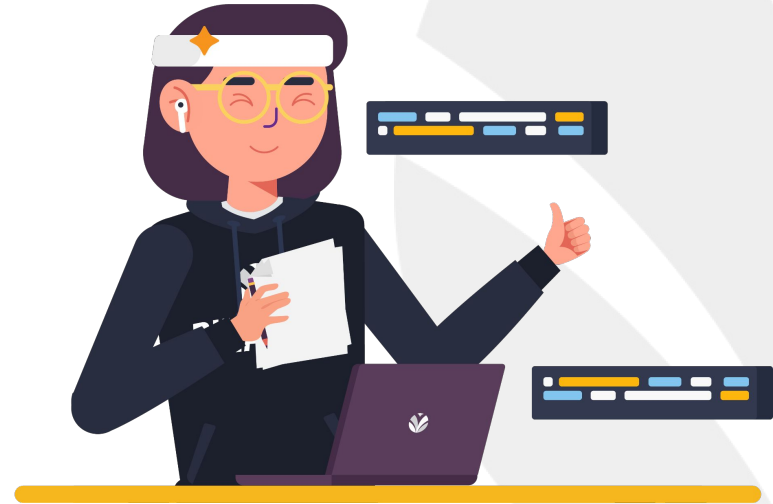
Stored procedure dapat menerima parameter, dan dapat mengatur variabel, menulis pernyataan IF, dan lainnya dalam stored procedure.



Kelebihan Stored Procedure

Kalau direkomendasiin, berarti ada alasannya dong. Beberapa kelebihan yang ditawarkan stored procedure antara lain:

- Meningkatkan performa, berhubung syntax SQL pada stored procedure tersimpan pada server, maka pemanggilan bisa lebih cepat.
- Mereduksi trafik jaringan, hal ini karena semua proses bisnis dilakukan di database dan aplikasi akan menerima hasil prosesnya saja



- Reusable, iya soalnya cukup ditulis sekali tapi bisa dipakai berkali-kali.
- Meningkatkan kontrol security, misalnya ketika user ABCD login di aplikasi dan nggak bisa akses tabel-tabel tertentu secara langsung, maka dia akan diarahkan melalui store procedure.

Ini artinya, dengan store procedure maka akan meningkatkan ke-valid-an data.



Kekurangan Stored Procedure

Unfortunately, ada kelebihan maka ada juga kekurangannya. Kita harus hati-hati nih sebagaimana catatan dibawah ini:

- Beban kerja server meningkat karena prosedur yang seharusnya bisa dikerjakan di aplikasi pemanggil harus dikerjakan pada server.
- Lebih sulit dalam debugging ketika stored procedure yang dibuat semakin banyak.



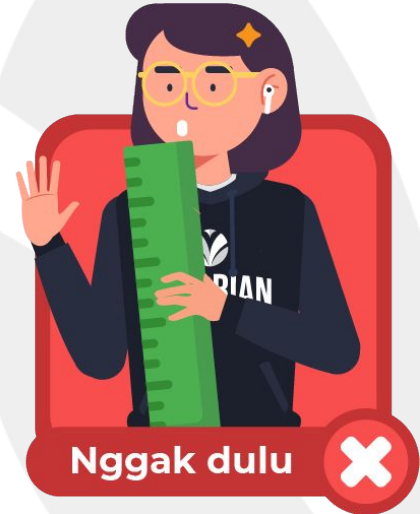
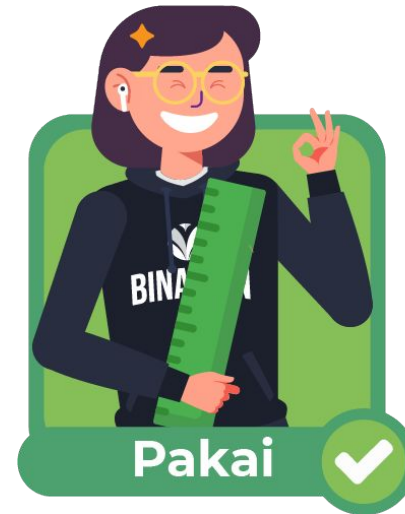
Walaupun begitu, dalam implementasi Store Procedure akan sering melibatkan parameter.

Parameter dalam Stored Procedure merupakan suatu komunikasi antara **pemanggilan stored procedure** dengan **stored procedure yang dibuat dengan pendeklarasian** (yang dilakukan setelah nama Stored Procedure).



Masih ngomongin parameter, pada pembuatan stored procedure kamu bisa pilih antara pakai parameter atau nggak pakai sama sekali.

Jumlah parameter yang digunakan itu tuh aslinya nggak ditentukan. Kamu bisa cuma menggunakan satu parameter atau bahkan ratusan sekaligus asalkan sesuai dengan kebutuhan yang akan digunakan.



Berhubung tadi ada banyak pilihan parameter, sebenarnya parameter Stored Procedure itu dibedakan menjadi tiga mode:

- **IN**

Parameter yang merupakan mode default ini mengindikasikan bahwa sebuah parameter dapat di-pass ke dalam stored procedure dengan catatan si nilainya nggak bisa diubah (dari dalam stored procedure)



- **OUT**

Mode ini mengindikasikan bahwa stored procedure dapat mengubah parameter dan mengirimkan kembali ke program pemanggil.

- **INOUT**

Mode ini pada dasarnya merupakan kombinasi dari mode IN dan OUT.



OUT



IN OUT

Gimana kenalan kamu sama Store Procedure?

Supaya gambaran kita nggak cuma teori aja, kita coba praktek dulu yuk.

Kita coba **Hands on Store Procedure on PostgreSQL**. Let's go!



Create Stored Procedure

Pertama banget nih, untuk menentukan Store Procedure baru, kamu perlu membuat procedure statement.

Syntax di samping adalah syntax dasar dari membuat store procedure di PostgreSQL.

```
create [or replace] procedure procedure_name(parameter_list)
language plpgsql
as $$
declare
-- variable declaration
begin
-- stored procedure body
end; $$
```


Belum selesai, guys.

Di samping ini ada contoh membuat store procedure bernama “transfer”.

Yaitu, kamu akan mentransfer sejumlah uang tertentu dari satu account ke account lainnya.

```
create or replace procedure transfer(  
    sender int,  
    receiver int,  
    amount dec  
)  
language plpgsql  
as $$  
begin  
    -- subtracting the amount from the sender's account  
    update accounts  
    set balance = balance - amount  
    where id = sender;  
  
    -- adding the amount to the receiver's account  
    update accounts  
    set balance = balance + amount  
    where id = receiver;  
  
    commit;  
end;$$
```

Calling a Stored Procedure

Next, untuk memanggil store procedure, kamu bisa menggunakan statement **call**.

Contoh di samping adalah statement yang meminta stored procedure untuk mentransfer \$1.000 dari satu account ke account lain.



```
call stored_procedure_name(argument_list);
```



```
call transfer(1,2,1000);
```

Yuhuuu, gimana nih gengs impresimu setelah belajar materi ini?

Terus, kira-kira kapan ya kita perlu untuk menggunakan Store Procedure?



Nah, selesai sudah pembahasan kita di Chapter 4 Topic 4 ini.

Selanjutnya, kita bakal bahas tentang **NoSQL**.

Penasaran kayak gimana? Yuk, langsung ke topik selanjutnya~

