DATABASE SQL

Pesantren PeTIK II YBM PLN
Jl. KH. Bisri Syansuri RT/01 RW/05, Plosogeneng,
Kec. Jombang, Kabupaten Jombang, Jawa Timur

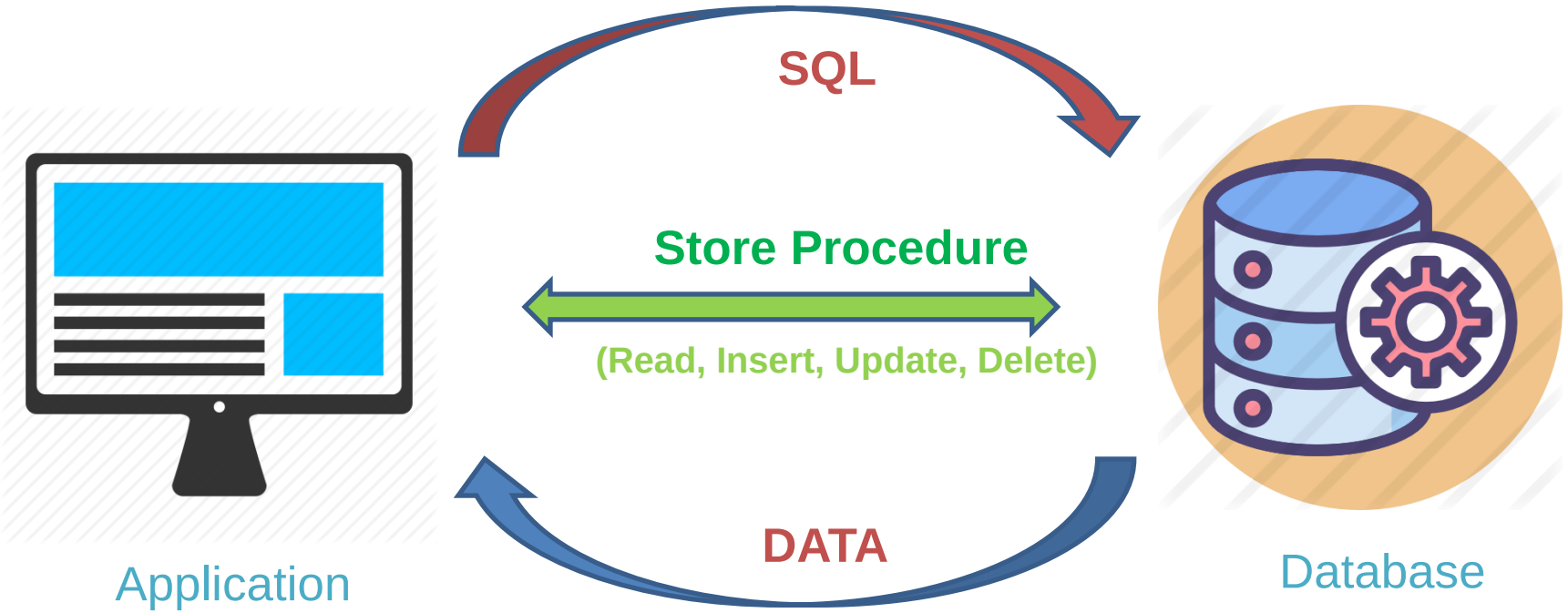
Pertemuan Ke-12

Materi

1. Pengantar Database
2. Pemodelan Data
3. Model Relasional Database
4. Normalisasi Database
5. Pengantar SQL
6. Perintah SQL SELECT 1
7. Perintah SQL SELECT 2
9. Fungsi Aggregate dan Grouping Data
10. Sub Query & SQL Join Table
11. View dan Analisa Query
12. **Store Procedure dan Function**
13. Trigger dan Transaction
14. Manajemen User
15. Backup dan Restore

12. Store Procedure dan Function

Stored Procedure dan Function



Procedure & Function: Rangkaian program SQL yang disimpan dalam database dan dapat dipanggil melalui program lain (aplikasi) atau lewat prompt SQL oleh user database

Stored Procedure & Function

- Fitur yang disupport mulai MySQL versi 5.0
- Stored Procedure adalah sebuah prosedur (pernyataan program yang berisi logika SQL) yang tersimpan dalam database server

Mengapa menggunakan Stored Procedure ?

- Karena berada dalam database server, logika pemrograman pada aplikasi tidak diperlukan lagi, yang harus dilakukan adalah memanggil stored procedure
- Mereduksi trafik pada jaringan yang mengakses ke database server, karena logika pemrograman dilakukan oleh database server bukan oleh komputer client

Keuntungan Stored Procedure/Function

- Sebuah Stored Procedure dikompilasi dan lebih cepat dalam mengeksekusi query
- Memproses data lewat Stored Procedure dilakukan pada Server sehingga mengurangi intensitas lalu lintas data pada jaringan
- Stored procedure adalah program modular yang dapat dipanggil oleh stored procedure lain.

Keuntungan Stored Procedure/Function

- Stored procedure mudah dilakukan perubahan (modifikasi) dan dapat dirasakan perubahan oleh semua user, karena tersimpan terpusat di server database.
- Stored procedure bisa menjadi komponen penting dalam keamanan database. Jika semua user melalui stored procedure maka semua akses table data dapat di kontrol.

Syntax:

```
CREATE [DEFINER = { user | CURRENT_USER }]
PROCEDURE sp_name ([proc_parameter[,...]])
[characteristic ...] routine_body
proc_parameter: [ IN | OUT | INOUT ] param_name type
type:
Any valid MySQL data type
characteristic:
COMMENT 'string'
| LANGUAGE SQL
| [NOT] DETERMINISTIC
| { CONTAINS SQL | NO SQL | READS SQL DATA
| MODIFIES SQL DATA }
| SQL SECURITY { DEFINER | INVOKER }
routine_body:
Valid SQL routine statement
```

Delimiter

- Delimiter adalah karakter atau string yang digunakan dibagian akhir eksekusi perintah SQL
- Default Delimiter adalah titik koma (;)
- Pada program SQL dengan Stored Procedure, tanda titik koma digunakan sebagai bagian dari program
- Delimiter titik koma (;) harus diganti agar dapat CREATE Stored Procedure atau Function

Mengganti Delimiter

- Delimiter titik koma akan diganti dengan $$

```
DELIMITER $$ ;

SELECT id,nama FROM pelanggan $$;
```

```
+----+---------------+
| id | nama          |
+----+---------------+
|  1 | Agung Sedayu  |
|  9 | Ahmad Hasan   |
|  8 | Andre Haru    |
| 10 | Cassanndra    |
|  7 | Dewi Gyat     |
|  6 | Gayatri Dwi   |
|  2 | Pandan Wangi  |
|  5 | Pradabashu    |
|  3 | Sekar Mirah   |
|  4 | Swandaru Geni |
+----+---------------+
```

Stored Procedure

- Buat stored procedure SelectAllPelanggan

```
DELIMITER $$ ;
CREATE PROCEDURE SelectAllPelanggan()
-> BEGIN
-> SELECT * FROM pelanggan;
-> END
-> $$
```

- Eksekusi stored procedure: call nama_procedure

```
call selectAllPelanggan();
```

Stored Procedure

- Lihat stored procedure

```
SHOW PROCEDURE status \G
```

```
                  Db: dbkoperasi
                Name: SelectAllPelanggan
                Type: PROCEDURE
             Definer: root@localhost
            Modified: 2021-06-18 08:23:08
             Created: 2021-06-18 08:23:08
       Security_type: DEFINER
             Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: latin1_swedish_ci
```

- Hapus stored procedure SelectAllPelanggan

```
DROP procedure IF EXISTS SelectAllPelanggan;
```

Contoh 1: procedure

- Buat procedure untuk menampilkan jumlah data produk

```
mysql> delimiter //
mysql> create procedure
    pro_hitung_produk(out param1 INT)
 -> begin
 -> select count(*) into param1 from produk;
 -> end;
 -> //
mysql> delimiter ;
```

- Eksekusi Stored Procedure

```
mysql> call pro_hitung_produk(@jumlah);

mysql> select @jumlah;
+---------+
| @jumlah |
+---------+
| 4       |
+---------+
```

Contoh 2: procedure

- Buat procedure untuk menampilkan jumlah data produk berdasarkan inputan idjenis produknya

```
mysql> delimiter //
mysql> create procedure pro_hitung_produk (out param1 INT, in
 param2 INT)
  -> begin
  -> select count(*) into param1 from produk
  -> WHERE jenis_produk_id=param2;
  -> end;
  -> //
mysql> delimiter ;
```

- Eksekusi Store Procedure untuk idjenis 1

```
mysql> call pro_hitung_produk(@jumlah,1);

mysql> select @jumlah;
+----------+
| @jumlah |
+----------+
| 3        |
+----------+
```

Contoh 3: procedure update data

- Buat procedure untuk update stok, data stok akan ditambahkan sesuai dengan inputan id produk dan jumlah stok yang akan ditambahkan

```
mysql> delimiter //
mysql> create procedure pro_update_stok (in idprod INT, in jumlah INT)
 -> begin
 -> UPDATE produk SET stok = stok + jumlah WHERE id=idprod;
 -> end;
 -> //

mysql> delimiter ;
```

- Eksekusi Store Procedure dan amati perubahan datanya

```
mysql>SELECT * FROM produk;

mysql>call pro_update_stok (1,20); // update stok tambahkan 20 untuk id produk 1

mysql>SELECT * FROM produk;
```

Function

- Buat fungsi hello yang mengembalikan text “selamat belajar”

```
DELIMITER $$ ;
CREATE FUNCTION hello ()
RETURNS TEXT
BEGIN

RETURN "SELAMAT BELAJAR";
END
```

- Melihat fungsi yang ada di database

```
show function status \G
```

- Menghapus fungsi

```
DROP function IF EXISTS hello;
```

Contoh 5: fungsi

- Buat gender dengan argumen tipe data char , jika inputan argument P cetak “Perempuan” jika inputan argument L cetak “Laki-Laki”

```
1  CREATE FUNCTION gender (jk char(1))
2  RETURNS varchar(20)
3  BEGIN
4    DECLARE string_gender varchar(20);
5    IF jk = 'P' THEN
6       SET string_gender = 'Perempuan';
7    ELSEIF jk='L' THEN
8       SET string_gender = 'Laki-Laki';
9    END IF;
10 RETURN string_gender;
11 END
```

- **SELECT nama,gender(jk) as gender FROM pelanggan ;**

TERIMA KASIH
ATAS SEGALA PERHATIAN
SEMOGA BERMANFAAT...