Prepared Statement

Ini materi penting jadi tolong benar-benar dipahami!

Mysqli_stmt adalah class khusus yang dipakai untuk menjalankan query MySQL sebagai *prepared statement*.
Prepared statement sendiri merupakan teknik memproses query MySQL dengan memisahkan antara query dengan data yang akan input. Nantinya, query disiapkan terlebih dahulu (proses *prepare*), kemudian data diinput (proses *bind*), dan baru query dijalankan (proses *execute*).
Dengan demikian, di dalam prepared statement terdapat 3 langkah yang harus di proses secara berurutan: **prepare**, **bind**, dan **execute**.
Seperti yang akan kita lihat nanti, sebenarnya prepared statement ini tampak lebih ribet daripada menjalankan query seperti yang sudah kita lakukan seperti sebelumnya. Namun prepared statement membawa beberapa keunggulan:

1. Query yang ditulis dengan prepared statement otomatis ter-validasi. Artinya, prepared statement kebal terhadap MySQL injection.
2. Untuk query yang berulang, kita hanya perlu menulis 1 perintah query saja (proses prepare), kemudian menginput data beberapa. Cara ini lebih efisien dibandingkan menjalankan query satu per satu.

INGAT INI !!!
Inti dari pembuatan *prepared statement* ada di proses **prepare** dan **bind**. Dalam proses *prepare*, kita membuat query seperti biasa namun untuk bagian data input diganti dengan tanda tanya " ? ", seperti contoh berikut:

```
"SELECT * FROM barang WHERE id_barang = ?"
```

atau

```
"INSERT INTO barang VALUES (?,?,?,?)"
```

Coba contoh berikut simpan dengan nama **prepared1.php** simpan di folder **pertemuan11**

Kita akan bahas detail setelahnya.

```php
<?php
mysqli_report(MYSQLI_REPORT_STRICT);
try {
 $mysqli = new mysqli("localhost", "root", "","ilkoom");
 // Proses prepare
 $stmt = $mysqli->prepare("SELECT * FROM barang WHERE id_barang = ?");

 // Proses bind
 $id_barang = 5;
 $stmt->bind_param("i", $id_barang);
```

1

2

```php
 // Proses execute
 $stmt->execute();

 // Proses menampilkan hasil query
 $result = $stmt->get_result();
 if ($mysqli->error){
  throw new Exception($mysqli->error, $mysqli->errno);
 }
 else {
  while ($row = $result->fetch_assoc()){
    echo $row['id_barang'];      echo " | ";
    echo $row['nama_barang'];    echo " | ";
    echo $row['jumlah_barang'];  echo " | ";
    echo $row['harga_barang'];   echo " | ";
    echo $row['tanggal_update'];
    echo "<br>";
  }
 }

 // Hapus memory dan tutup prepared statement
 $stmt->free_result();
 $stmt->close();
}
catch (Exception $e) {
 echo "Koneksi / Query bermasalah: ".$e->getMessage(). " (".$e->getCode().")";
}
finally {
 if (isset($mysqli)) {
   $mysqli->close();
 }
}
```

Hasil kode program:

```
5 | Smartphone Xiaomi Pocophone F1 | 25 | 4750000 | 2019-01-07 21:09:41
```

PENJELASAN PROGRAM

1. Dalam **mysqli** object, proses *prepared* ini menggunakan method **mysqli::prepare()**. Method ini butuh **sebuah argument bertipe string yang berisi query** kemudian mengembalikan sebuah **mysqli_stmt** object. Dalam program di atas:

```
$stmt = $mysqli->prepare("SELECT * FROM barang WHERE id_barang = ?");
```

Dengan perintah ini, variabel `$stmt` akan berisi sebuah **mysqli_stmt** object. Sepanjang proses pembuatan prepared statement, variabel inilah yang akan terus kita akses. Nama variabel penampung **mysqli_stmt** object ini boleh bebas, tidak harus `$stmt`.

2. Langkah selanjutnya adalah proses **bind**, yakni mengisi tanda tanya yang sudah kita siapkan pada saat *prepare*. Proses *bind* dijalankan dengan method `mysqli_stmt::bind_param()`.

Perhatikan bahwa method `bind_param()` ini adalah "kepunyaan" dari **mysqli_stmt** object. Method `mysqli_stmt::bind_param()` **butuh minimal 2 argument**, yakni jenis tipe data dan variabel yang akan diinput. Disebut "**minimal**" karena argument method bisa lebih dari 2 tergantung jumlah tanda tanya yang terdapat di query.

Dalam contoh di atas karena hanya ada satu attribute yang dibutuhkan dalam query yaitu **id_barang** maka Berikut penulisan dari `mysqli_stmt::bind_param()`:

```
1 $stmt = $mysqli->prepare("SELECT * FROM barang WHERE id_barang = ?");
2 $id_barang = 5;
3 $stmt->bind_param("i", $id_barang);
```

Terdapat 4 buah pilihan tipe data pada saat proses *bind*, yakni:
- `i` = integer
- `d` = double
- `s` = string
- `b` = blob (binary)

Oya perlu dipahami juga knapa **`$id_barang kita isi nilai sample 5.`** `Karena kita belum memiliki form tampil data barang, sebenarnya dalam implementasi nilai 5 tersebut di input melalui form oleh user dan akan ditangkap oleh variable $id_barang.`

DI BAWAH INI PENTING PERHATIKAN !!!

LALU bagaimana cara penulisan bind_param bila attribute query lebih dari satu??? Begini contoh lain potongan perintah prepare dan bind untuk query INSERT ke tabel barang:

```
// Buat prepared statement untuk input data barang
 $stmt = $mysqli->prepare("INSERT INTO barang (nama_barang,
jumlah_barang, harga_barang, tanggal_update) VALUES (?,?,?,?)");

 // Proses bind
 $stmt->bind_param("siis", $nama_barang, $jumlah_barang,
            $harga_barang, $tanggal_update);
```

Terlihat jelas bedanya ya??? Dengan 4 attribute berarti ada 4 tanda tanya, dan berarti juga terdapat 4 argument dalam proses BIND. PAHAM YA?? JANGAN SAMPAI GA PAHAM. OYA perhatikan juga cara pemberian tipe data contoh di atas ditulis "siis" artinya argument yang dikirim memiliki tipe "**string, integer, integer, string". HARUS URUT**

3 Setelah *prepare* dan *bind*, langkah terakhir adalah *execute* yakni menjalankan query tersebut dengan method `mysqli_stmt::execute()`.

4 *Proses menampilkan hasil query*
perintah `$result = $stmt->get_result()`. Variabel **`$result`** di sini akan berisi **mysqli_result** object hasil pemrosesan *prepared statement*. Jika anda masih ingat (**harus ingat**), `mysqli_result` object ini sudah kita bahas cukup detail sebelumnya. Tidak ada perbedaan antara `mysqli_result` object hasil *prepared statement* dengan `mysqli_result` object hasil `mysqli::query()` biasa. Untuk menampilkan hasilnya saya menggunakan perulangan `while ($row = $result->fetch_assoc())`

CATATAN PENTING, SEPERTI YANG SUDAH KITA PELAJARI SEBELUMNYA, PROSES EKSTRAK DATA INI ADA BEBERAPA CARA YANG TELAH KITA PELAJARI SEBELUMNYAM DAN SEMUA DAPAT DIGUNAKAN DI SINI, PADA CONTOH PERTAMA DI ATAS KITA GUNAKAN FETCH_ASSOC(), CEK LAGI ADA BERAPA CARA SELAIN INI DI MATERI SEBELUMNYA.

Ok berikut contoh lain agar lebih paham lagi, simpan dengan nama **prepared2.php**

```php
<?php
mysqli_report(MYSQLI_REPORT_STRICT);
try {
 $mysqli = new mysqli("localhost", "root", "","ilkoom");

 // Proses prepare
 $stmt = $mysqli->prepare("SELECT * FROM barang WHERE id_barang = ?
       OR nama_barang = ?");

 // Proses bind
 $id_barang = 5;
 $nama_barang = "TV Samsung 43NU7090 4K";
 $stmt->bind_param("is", $id_barang, $nama_barang);

 // Proses execute
 $stmt->execute();
```

```
 // Proses menampilkan hasil query dengan fetch_assoc()
 $result = $stmt->get_result();
 if ($mysqli->error){
  throw new Exception($mysqli->error, $mysqli->errno);
 }
 else {
  while ($row = $result->fetch_assoc()){
   echo $row['id_barang'];      echo " | ";
   echo $row['nama_barang'];    echo " | ";
   echo $row['jumlah_barang'];  echo " | ";
   echo $row['harga_barang'];   echo " | ";
   echo $row['tanggal_update'];
   echo "<br>";
  }
 }
 // Hapus memory dan tutup prepared statement
 $stmt->free_result();
 $stmt->close();
}
catch (Exception $e) {
 echo "Koneksi / Query bermasalah: ".$e->getMessage(). " (".$e->getCode().")";
}
finally {
 if (isset($mysqli)) {
  $mysqli->close();
 }
}
```

Hasil kode program:

```
1 | TV Samsung 43NU7090 4K | 5 | 5399000 | 2019-01-07 21:09:41
5 | Smartphone Xiaomi Pocophone F1 | 25 | 4750000 | 2019-01-07 21:09:41
```

Sedikit perbedaan dari program sebelumnya terletak pada sisi prepare dan bind

```
// Proses prepare
```

```
$stmt = $mysqli->prepare("SELECT * FROM barang WHERE id_barang = ?
      OR nama_barang = ?");
```

Query di atas menambahkan sebuah kondisi untuk pencarian berdasarkan id_barang atau nama_barang, yang artinya ada 2 attribut, artinya lagi pada bagian proses bind harus disiapkan 2 buah argument untuk mengisi nilai pada query

```
// Proses bind
$id_barang = 5;
$nama_barang = "TV Samsung 43NU7090 4K";
$stmt->bind_param("is", $id_barang, $nama_barang);
```

Bagian menampilkan data masih manggunaan fetch_assoc()

Dicontoh selanjutnya kita akan gunakan fungsi yang lain.
Jelas ya??

Simpan program dengan nama **prepared3.php untuk contoh penggunaan fetch_row()**

```
<?php
mysqli_report(MYSQLI_REPORT_STRICT);

try {
 $mysqli = new mysqli("localhost", "root", "","ilkoom");

 // Buat prepared statement untuk ambil data barang
 $query = "SELECT * FROM barang WHERE id_barang = ?";
 $stmt = $mysqli->prepare($query);

 // Proses bind
 $stmt->bind_param("i", $id_barang);

 //proses menampilkan data dengan fetch_row()
 $id_barang = 2;
 $stmt->execute();
 $result = $stmt->get_result();
 $row = $result->fetch_row(); // bedanya di sini !!! bandingkan dengan sebelumnya
 echo $row[0]." | ".$row[1]. " | ".$row[2]. " | ".$row[3]. " | ".$row[4];
```

```
 $stmt->free_result();
 $stmt->close();
}
catch (Exception $e) {
 echo "Koneksi / Query bermasalah: ".$e->getMessage(). " (".$e->getCode().")";
}
finally {
 if (isset($mysqli)) {
  $mysqli->close();
 }
}
```

Simpan program di bawah dengan nama **prepared4.php** untuk contoh menampilkan data dengan fetch_object()

```
<?php
mysqli_report(MYSQLI_REPORT_STRICT);

try {
 $mysqli = new mysqli("localhost", "root", "","ilkoom");

 // Buat prepared statement untuk ambil data barang
 $stmt = $mysqli->prepare("SELECT * FROM barang WHERE id_barang = ?");

 // Proses bind
 $stmt->bind_param("i", $id_barang);
 $id_barang = 3;

 // Proses execute
 $stmt->execute();

 // Proses menampilkan hasil query
 $result = $stmt->get_result();
 $row = $result->fetch_object(); //bedanya di sini !!! bandingkan dengan sebelumnya
```

```
 echo $row->id_barang;      echo " | "; //ini juga beda
 echo $row->nama_barang;    echo " | ";
 echo $row->jumlah_barang;  echo " | ";
 echo $row->harga_barang;   echo " | ";
 echo $row->tanggal_update;

 $stmt->free_result();
 $stmt->close();
}
catch (Exception $e) {
 echo "Koneksi / Query bermasalah: ".$e->getMessage(). " (".$e->getCode().")";
}
finally {
 if (isset($mysqli)) {
   $mysqli->close();
 }
}
```

Hasil kode program:

```
3 | Laptop ASUS ROG GL503GE | 7 | 16200000 | 2019-01-07 21:09:41
```

OK YAP.

Nah berikut ini ada cara yang lebih simple dalam proses menampikan data yaitu dengan konsep chaining method, cukup mudah jadi jangan bingung, silakan coba di impementasikan pada program2 yang telah dibuat di atas, **khusus untuk program yang tidak menggunakan perulangan while**

Ubah bagian dibawah ini saja dari program sebelumnya

```
// Proses menampilkan hasil query cara biasa
 $result = $stmt->get_result();
 $row = $result->fetch_object();
```

```
// Proses menampilkan hasil query dengan chaining method
 $row = $stmt->get_result()->fetch_object(); //ini penting chain method
```

OK LANJUT

DARI awal kita membuat program untuk query SELECT, LALU BAGAIMANA dengan QUERY insert, update dan delete?????

NAH prinsipnya sama saja, bahkan bisa dibilang lebih sederhana karena tidak perlu proses menampilkan hasil query seperti yang dilakukan pada query select. Berikut contoh program query insert, simpan dengan nama **prepared5.php**

```php
<?php
mysqli_report(MYSQLI_REPORT_STRICT);

try {
 $mysqli = new mysqli("localhost", "root", "","ilkoom");

 // Buat format tanggal hari ini
 $sekarang = new DateTime('now', new DateTimeZone('Asia/Jakarta'));
 $timestamp = $sekarang->format("Y-m-d H:i:s");

 // Buat prepared statement untuk input data barang
 $stmt = $mysqli->prepare("INSERT INTO barang (nama_barang,
 jumlah_barang, harga_barang, tanggal_update) VALUES (?,?,?,?)");

 // Proses bind
 $stmt->bind_param("siis", $nama_barang, $jumlah_barang,
                $harga_barang, $tanggal_update);

 $nama_barang = "Sharp Microwave Oven R-728(K)";
 $jumlah_barang = 20;
 $harga_barang = 1250500;
 $tanggal_update = $timestamp;

 // Proses execute
 $stmt->execute();
 echo "Terdapat ".$mysqli->affected_rows." baris yang ditambah <br>";
```

```
  $stmt->close();
}
catch (Exception $e) {
  echo "Koneksi / Query bermasalah: ".$e->getMessage(). " (".$e->getCode().")";
}
finally {
  if (isset($mysqli)) {
    $mysqli->close();
  }
}
```

Hasil kode program:

```
Terdapat 1 baris yang ditambah
```

Nah perhatikan program diatas lebih simple dari program sebelumnya, kalian hanya cukup memahami query insert saja,

Ok sekarang bagaimana bila setelah data di insert, kita juga ingin menampilkan datanya pada browser??? Ya cukup simple kalian tinggal membuat prepared statement untuk menampilkan data barang, dibawah proses insert, berikut contoh programnya simpan dengan nama **prepared6.php**

```
<?php
mysqli_report(MYSQLI_REPORT_STRICT);

try {
  $mysqli = new mysqli("localhost", "root", "","ilkoom");

  // Buat format tanggal hari ini
  $sekarang = new DateTime('now', new DateTimeZone('Asia/Jakarta'));
  $timestamp = $sekarang->format("Y-m-d H:i:s");

  // Buat prepared statement untuk input data barang
  $stmt = $mysqli->prepare("INSERT INTO barang (nama_barang,
  jumlah_barang, harga_barang, tanggal_update) VALUES (?,?,?,?)");

  $stmt->bind_param("siis", $nama_barang, $jumlah_barang,
                $harga_barang, $tanggal_update);
```

```php
// Input data
$nama_barang = "Cosmos CRJ-8229 - Rice Cooker";
$jumlah_barang = 4;
$harga_barang = 299000;
$tanggal_update = $timestamp;

$stmt->execute();
echo "Terdapat ".$mysqli->affected_rows." baris yang ditambah <br>";
$stmt->close();

// Proses prepare untuk menampilkan semua isi tabel barang
$stmt = $mysqli->prepare("SELECT * FROM barang WHERE id_barang");

// Proses execute
$stmt->execute();

// Proses menampilkan hasil query
$result = $stmt->get_result();
  while ($row = $result->fetch_assoc()){
    echo $row['id_barang'];      echo " | ";
    echo $row['nama_barang'];    echo " | ";
    echo $row['jumlah_barang'];  echo " | ";
    echo $row['harga_barang'];   echo " | ";
    echo $row['tanggal_update'];
    echo "<br>";
  }

// Hapus memory dan tutup prepared statement
$stmt->free_result();
$stmt->close();
}
catch (Exception $e) {
 echo "Koneksi / Query bermasalah: ".$e->getMessage(). " (".$e->getCode().")";
```

```
}
finally {
  if (isset($mysqli)) {
    $mysqli->close();
  }
}
```

Hasil kode program:

```
Terdapat 1 baris yang ditambah
1 | TV Samsung 43NU7090 4K | 5 | 5399000 | 2019-01-07 21:09:41
2 | Kulkas LG GC-A432HLHU | 10 | 7600000 | 2019-01-07 21:09:41
3 | Laptop ASUS ROG GL503GE | 7 | 16200000 | 2019-01-07 21:09:41
4 | Printer Epson L220 | 14 | 2099000 | 2019-01-07 21:09:41
5 | Smartphone Xiaomi Pocophone F1 | 25 | 4750000 | 2019-01-07 21:09:41
6 | Sharp Microwave Oven R-728(K) | 20 | 1250500 | 2019-01-09 17:08:40
7 | Cosmos CRJ-8229 - Rice Cooker | 4 | 299000 | 2019-01-09 17:56:27
```

ok materi kita untuk mysqli sudah selesai.

Agar lebih mentap buat program untuk soal berikut, tipe data attribute silakan disesuaikan, simpan dalam folder yang sama dengan latihan diatas. Masing masing soal disimpan dalam file berbeda, beri nama file sesuai nomor soal, misal **nomor1.php**

1. Buat database SIMAK
2. Buat tabel mahasiswa
   Property tabel mahasiswa
   - Nim
   - Nama
   - Program_studi
   - Alamat
   - Tanggal_lahir
   - Jenis-kelamin
3. Tambah 5 buah data mahasiswa bebas
4. Program untuk update data nama mahasiswa
5. Program untuk hapus data mahasiswa

Selamat mengerjakan