

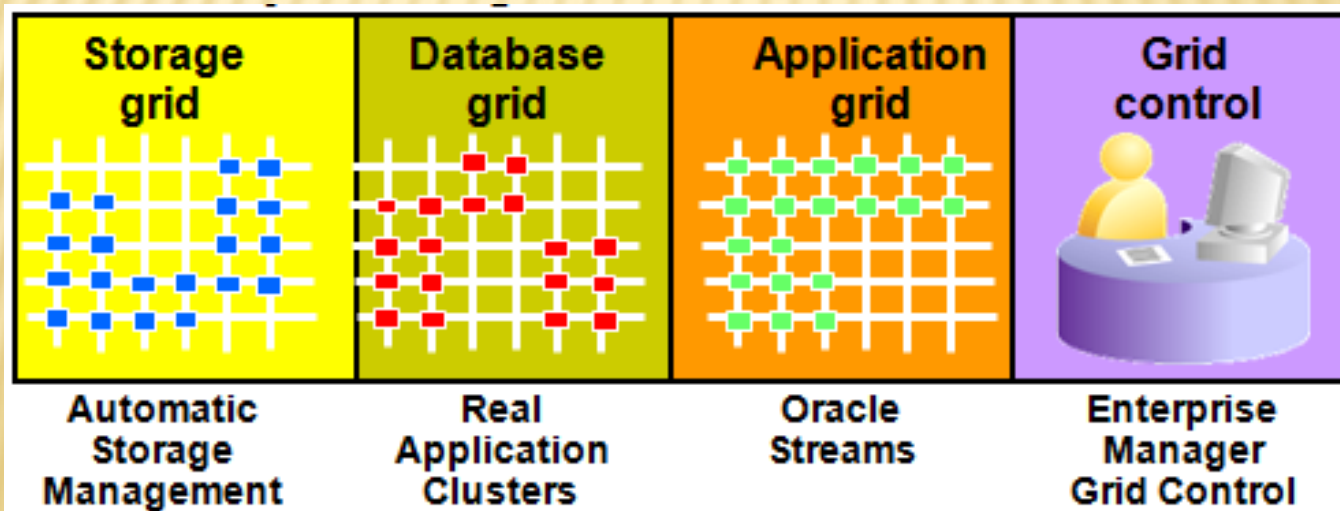
Pada sistem operasi Windows

PHP - ORACLE

Oleh: Ahmad Syauqi Ahsan

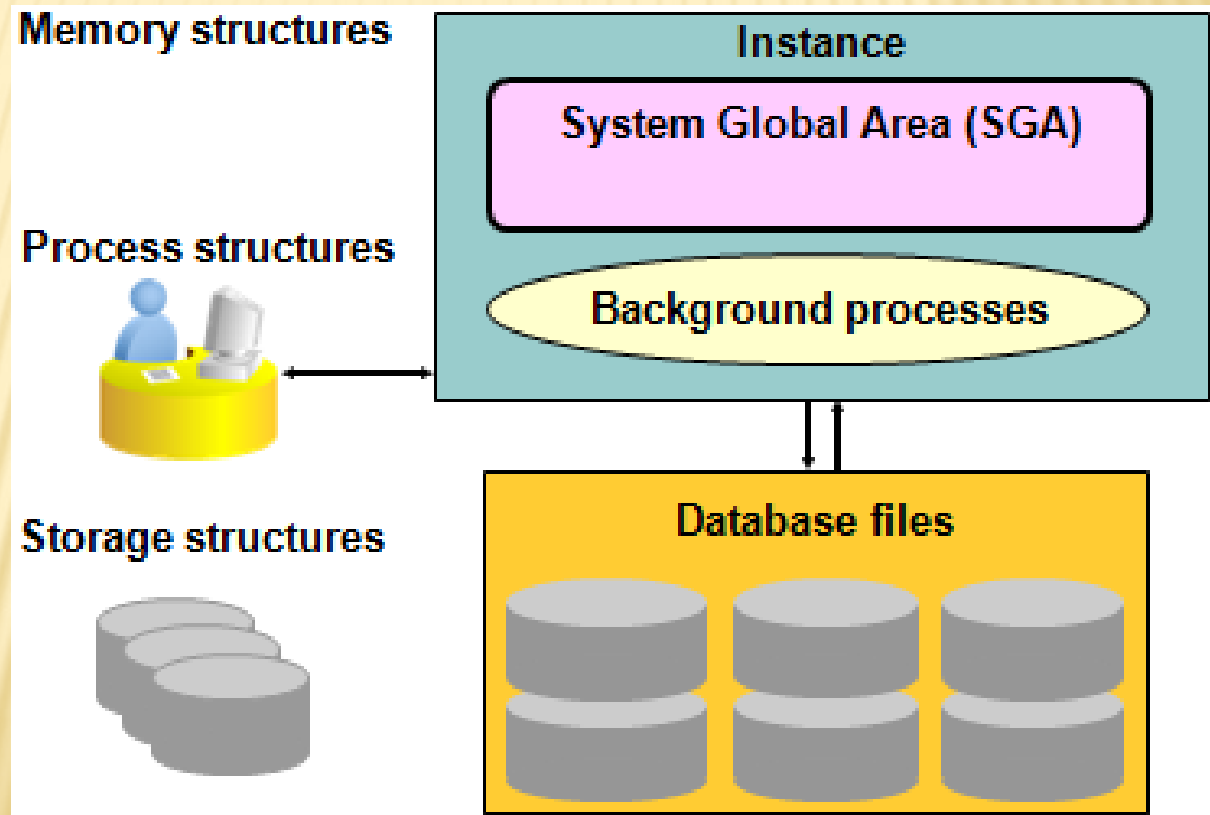
ORACLE DATABASE

- ✘ Oracle Database merupakan multi-user relational database management system (RDBMS) yang dapat berjalan di semua sistem operasi utama (Unix, Linux, windows, dll) baik sebagai server maupun client (atau berfungsi sebagai keduanya).
- ✘ Saat ini Oracle Database mempunyai market share terbesar untuk kategori database server.
- ✘ Oracle Database 10g → “g” diartikan sebagai *Grid*. Keunggulan dari Grid: low cost, high quality of service dan easy to manage.

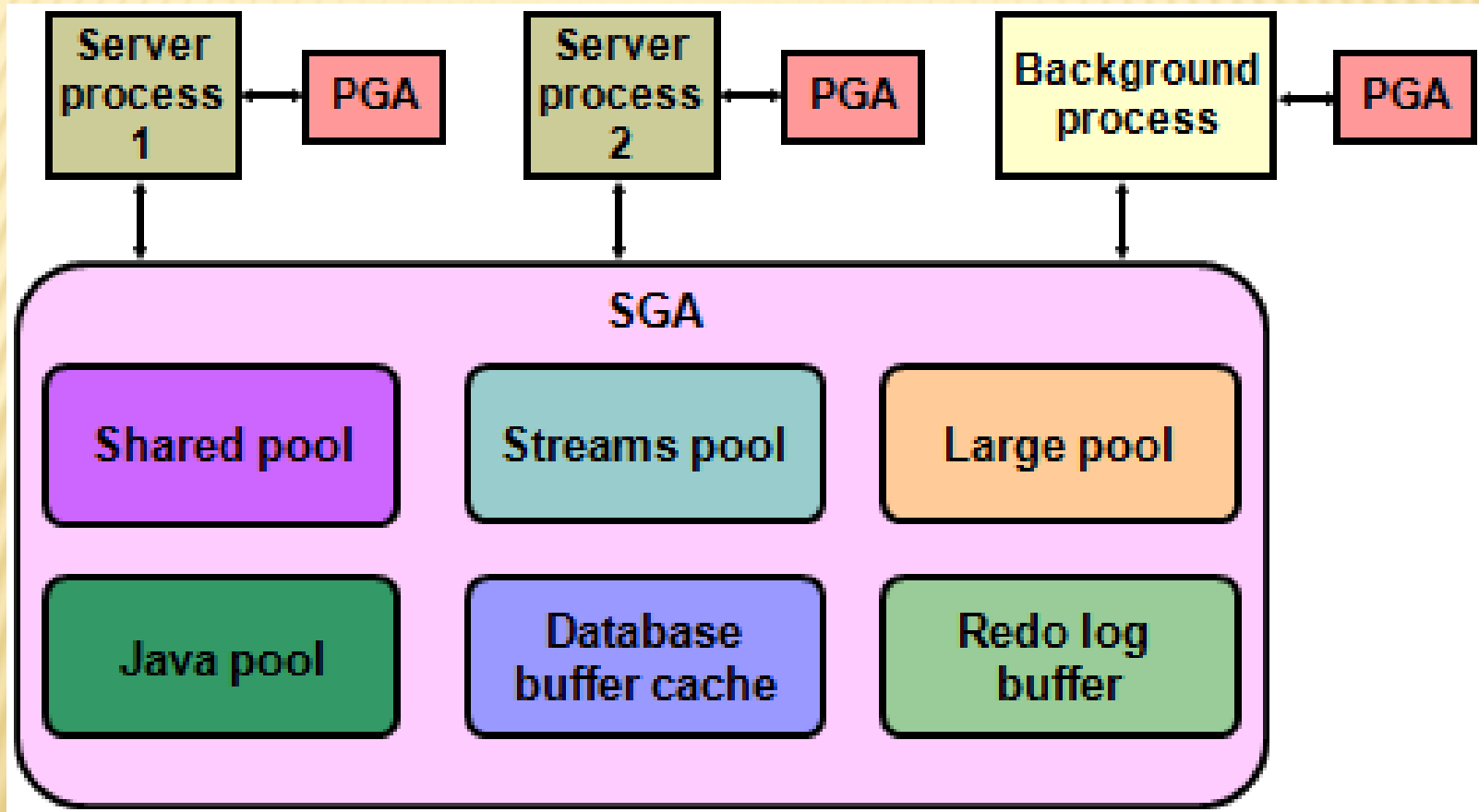


ORACLE DATABASE ARCHITECTURE

- ✘ Oracle database terdiri dari **Oracle instance** dan **Oracle database**



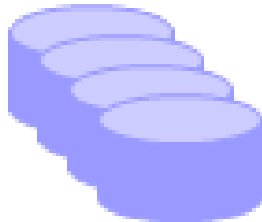
ORACLE INSTANCE (MEMORY STRUCTURE)



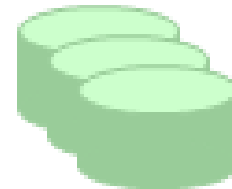
ORACLE PHYSICAL DATABASE STRUCTURE



Control files



Data files



Online redo log files



Parameter file



Backup files



Archive log files



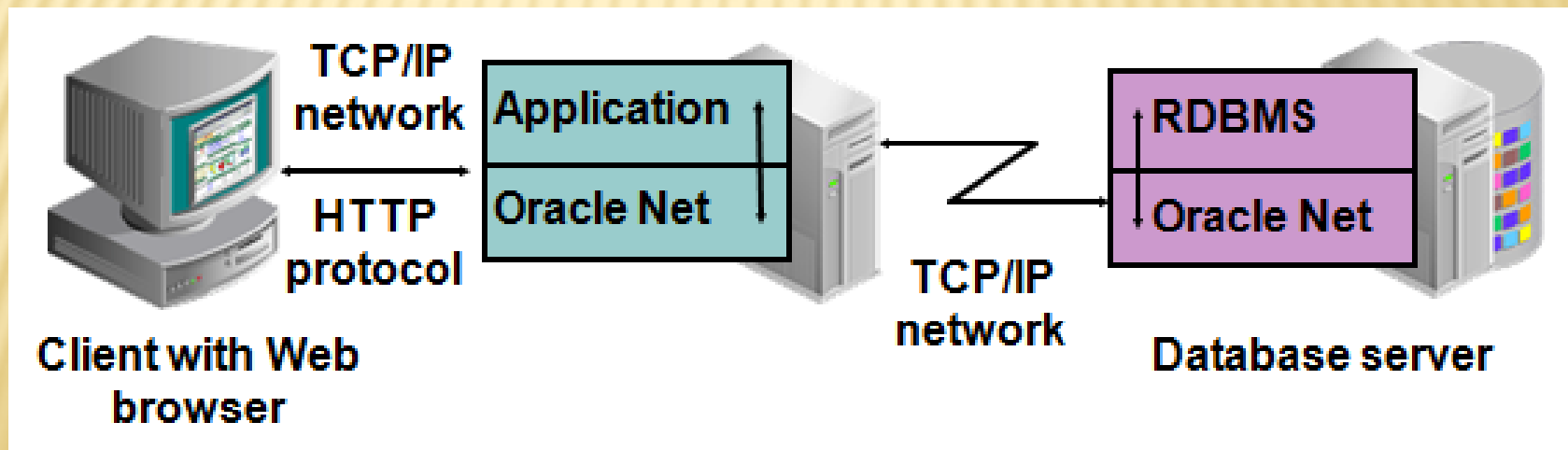
Password file



Alert and trace log files

ORACLE NET

- ✘ Untuk dapat terkoneksi dengan Oracle Database Server, sebuah komputer harus mempunyai Oracle Net Client
- ✘ Oracle Net Client dapat diperoleh dari:
 - + Oracle database software installation
 - + Oracle Net Client
 - + Oracle Instant Client



ORACLE INSTANT CLIENT

- ✘ Memungkinkan koneksi ke Oracle Database Server tanpa harus menginstall Oracle Client standart (yang ukurannya cukup besar).
- ✘ Dengan Oracle Instant Client, suatu aplikasi dapat terhubung ke Oracle Database Server melalui OCI, OCCI, Pro*C, ODBC, ataupun JDBC.
- ✘ For production use and Free.

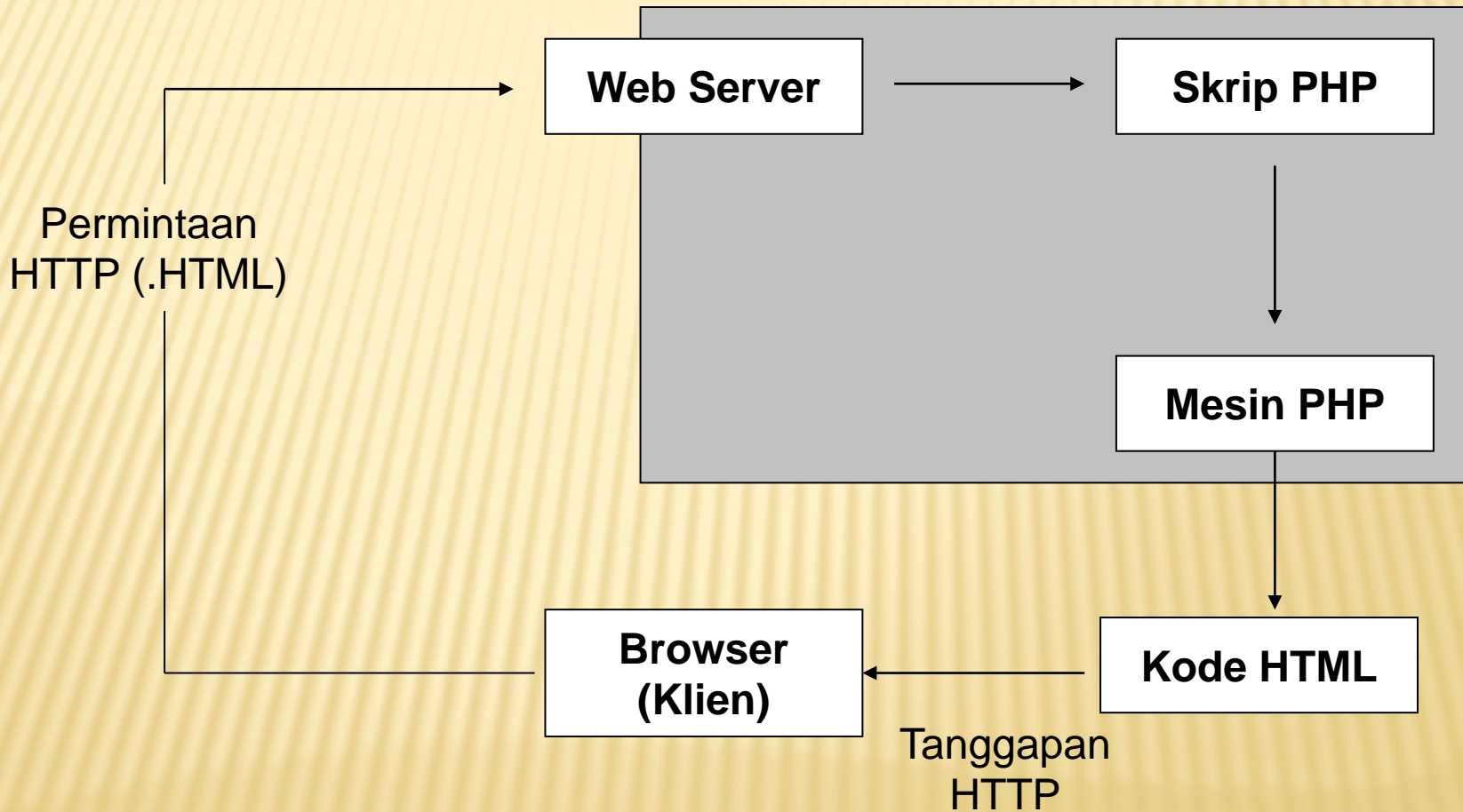
INSTALASI ORACLE INSTANT CLIENT

- ✘ Download paket Instant Client yang sesuai dengan platform/OS.
- ✘ Extract paket tersebut kedalam satu directory. Misal di “c:\instantclient”
- ✘ Edit environment variable pada windows, dan tambahkan lokasi dari instantclient (Misal: “c:\instantclient”) kedalam parameter “PATH”.
- ✘ Oracle Instant Client siap digunakan.

PHP

- ✘ PHP (Hypertext Preprocessor) merupakan scripting language yang digunakan untuk memproduksi website yang dinamis.
- ✘ Kode2 PHP dapat ditanamkan dalam dokumen HTML (HyperText Markup Language). Begitu pula sebaliknya, tag2 HTML juga bisa ditanamkan didalam kode2 PHP.
- ✘ Umumnya, PHP digunakan sebagai processor untuk hampir semua jenis web server modern(Apache, IIS, dll).
Walaupun sebenarnya PHP juga dapat digunakan sebagai stand-alone interpreter pada hampir semua sistem operasi (Windows, Linux, dll).
- ✘ Pada kesempatan ini, kita akan membahas penggunaan PHP pada Apache web server.

CARA KERJA PHP



APACHE WEB SERVER

- ✘ Merupakan open source webserver yang dibangun dan dikelola oleh komunitas developer dibawah naungan Apache Software Foundation.
- ✘ Tersedia untuk berbagai macam sistem operasi: Unix, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X, Microsoft Windows, OS/2, dll.
- ✘ Pada February 2010, Apache digunakan oleh 54,46% dari seluruh website yang ada.

XAMPP

- ✘ XAMPP merupakan suatu paket yang terdiri dari Apache web server, MySQL database, PHP, dan Perl).
- ✘ XAMPP adalah singkatan dari: X (cross platform), A (Apache), M (MySQL), P (PHP), dan P (Perl).
- ✘ XAMPP diupdate secara regular untuk mengakomodir versi terbaru dari Apache/MySQL/PHP/Perl. Juga terdapat beberapa modul tambahan seperti OpenSSL dan phpMyAdmin.
- ✘ Dengan menggunakan XAMPP, proses instalasi dan konfigurasi akan lebih mudah dan lebih mudah dibandingkan dengan menginstall masing2 software secara terpisah.

ENABLING ORACLE CONNECTION FROM PHP

- ✘ Secara default, koneksi ke Oracle dari PHP dinonaktifkan.
- ✘ Cara meng-enable-kan koneksi dari PHP ke Oracle Database Server adalah sebagai berikut:
 1. Buka file php.ini (ini merupakan file konfigurasi PHP)
 2. Ubah baris
`;extension=php_oci8.dll`
menjadi (hilangkan karakter `;`)
`extension=php_oci8.dll`
 3. Restart Apache Web Server

OCI (ORACLE CALL INTERFACE)

- ✘ PHP memanfaatkan OCI untuk terhubung dengan Oracle Database Server
- ✘ Berikut fungsi2 PHP yang sering digunakan untuk pengaksesan data pada Oracle Database Server:
 - + `oci_connect()`
 - + `oci_close()`
 - + `oci_parse()`
 - + `oci_free_statement()`
 - + `oci_execute()`
 - + `oci_fetch()`
 - + `oci_result()`

OCI_CONNECT ()

- ✘ Berfungsi untuk membuka koneksi dengan Oracle Database Server
- ✘ Syntax penulisan:

```
resource oci_connect(  
string username, string password  
[,string database])
```

CONTOH PENGGUNAAN OCI_CONNECT ()

```
<?php
    $con=oci_connect("hr","hr","10.252.44.102/orcl");
    if (!$con)
        echo "Koneksi Gagal";
    else
        echo "Koneksi Berhasil";
?>
```

- ✘ Script diatas berusaha membuka koneksi menggunakan username “hr” dengan password “hr” ke Oracle Database Server di mesin dengan IP “10.252.44.102” dengan nama database “orcl”
- ✘ Script akan menampilkan “Koneksi Berhasil” jika bisa terhubung dan akan menampilkan “Koneksi Gagal” jika sebaliknya.

OCI_CLOSE ()

- ✘ Berfungsi untuk menutup koneksi dengan Oracle Database Server
- ✘ Syntax penulisan:
`bool oci_close(resource connection)`
- ✘ `connection` adalah parameter yang didapat pada pemanggilan `oci_connect ()`
- ✘ Sangat direkomendasikan untuk selalu menutup koneksi dengan Oracle Database Server yang sudah tidak digunakan, karena ini akan membebaskan resource pada server untuk dapat digunakan oleh user yang lain.

CONTOH PENGGUNAAN OCI_CLOSE ()

```
<?php
    $con=oci_connect("hr","hr","10.252.44.102/orcl");
    if (!$con)
        echo "Koneksi Gagal";
    else
        echo "Koneksi Berhasil";
    oci_close($con);
?>
```

OCI_PARSE ()

- ✘ Berfungsi untuk menyiapkan `sql_text` menggunakan `connection` yang akan mengembalikan satu `statement identifier` untuk dapat digunakan oleh fungsi `oci_execute ()` maupun fungsi2 yang lain.
- ✘ Syntax penulisan:

```
resource oci_parse(resource  
connection, string sql_text)
```

OCI_FREE_STATEMENT ()

- ✘ Disarankan untuk memanggil fungsi ini untuk membebaskan resource yang digunakan oleh suatu statement yang dihasilkan oleh fungsi `oci_parse ()`
- ✘ Syntax penulisan:

```
bool oci_free_statement(  
resource statement)
```

OCI_EXECUTE ()

- ✘ Berfungsi untuk mengeksekusi statemen SQL yang sebelumnya telah didefinisikan menggunakan fungsi `oci_parse ()`

- ✘ Syntax penulisan:

```
bool oci_execute(resource statement  
[, int mode])
```

- ✘ mode adalah parameter opsional yang dapat digunakan untuk mengontrol apakah statemen yang dieksekusi akan langsung di-commit atau tidak. Nilai defaultnya: statemen akan langsung di-commit.

CONTOH OCI_PARSE () & OCI_EXECUTE ()

```
<?php
```

```
    $con=oci_connect("hr","hr","10.252.44.102/orcl");  
    $stid=oci_parse($con,"select * from employees");  
    oci_execute($stid);  
    oci_free_statement($stid);  
    oci_close($con);
```

```
?>
```

- ✘ Script diatas menggunakan `oci_parse ()` untuk menyiapkan query “**select * from employees**” menggunakan koneksi `$con` dan meyimpan statement identifiernya dalam variabel `$stid`.
- ✘ Kemudian statement identifier `$stid` tersebut dieksekusi menggunakan `oci_execute ()`
- ✘ Script diatas tidak menampilkan data.

OCI_FETCH ()

- ✘ Berfungsi untuk mengambil data dari query yang telah dieksekusi dan menyimpannya kedalam internal buffer. Setiap pemanggilan `oci_fetch()` hanya mengambil satu baris data.

Data pada internal buffer ini bisa diakses dengan fungsi `oci_result()`

- ✘ Syntax penulisan:

```
bool oci_fetch(resource $statement)
```

OCI_RESULT ()

- ✘ Berfungsi untuk mengambil data dari internal buffer (yang dihasilkan oleh fungsi `oci_fetch()`), untuk selanjutnya dapat ditampilkan pada browser.

- ✘ Syntax penulisan:

```
mixed oci_result(  
resource statement, mixed field)
```

- ✘ Parameter `field` dapat diisi dengan nomor kolom maupun nama kolom.

CONTOH OCI_FETCH () & OCI_RESULT ()

```
<?php
    $con=oci_connect("hr","hr","10.252.44.102/orcl");
    $stid=oci_parse($con,"select last_name, salary from
employees");
    oci_execute($stid);
    oci_fetch($stid);
    echo oci_result($stid, 1);
    echo oci_result($stid, "SALARY");
    oci_free_statement($stid);
    oci_close($con);
?>
```

CONTOH #2 OCI_FETCH() & OCI_RESULT()

```
<?php
    $con=oci_connect("hr","hr","10.252.44.102/orcl");
    $stid=oci_parse($con,"select last_name, salary from
employees");
    oci_execute($stid);
    while (oci_fetch($stid)){
        echo oci_result($stid, 1);
        echo oci_result($stid, "SALARY");
    }
    oci_free_statement($stid);
    oci_close($con);
?>
```

CONTOH #3 OCI_FETCH() & OCI_RESULT()

```
<?php
    $con=oci_connect("hr","hr","10.252.44.102/orcl");
    $stid=oci_parse($con,"select last_name, salary from
employees");
    oci_execute($stid);
    echo "<table border=\"1\" align=\"left\">";
    echo "<tr><th>Nama</th><th>Gaji</th></tr>";
    while (oci_fetch($stid)){
        echo "<tr><td>".oci_result($stid, 1)."</td>";
        echo "<td>".oci_result($stid, 2)."</td></tr>";
    }
    echo "</table>";
    oci_free_statement($stid);
    oci_close($con);
?>
```

MEMILIH DATA DENGAN KLAUSA WHERE

```
<form name="kirim" method="post" action="">
  Department ID
  <input type="text" name="dept_id">
  <input type="submit" name="Submit" value="Kirim">
</form>
```

```
<?php
  $dept_id=@$_POST['dept_id'];
  if($dept_id<>""){
    $con=oci_connect("hr","hr","10.252.44.102/orcl");
    $stid=oci_parse($con,"select last_name, salary,
    department_id from employees
    where department_id='`$dept_id'");
```

MEMILIH DATA DENGAN KLAUSA WHERE (LANJUTAN)

```
oci_execute($stid);
echo "<table border=\"1\" align=\"left\">";
echo "<tr><th>Nama</th><th>Gaji</th>"
echo "<th>Departement ID</th></tr>";
while (oci_fetch($stid)){
    echo "<tr><td>".oci_result($stid, 1)."</td>";
    echo "<td>".oci_result($stid, 2)."</td>";
    echo "<td>".oci_result($stid, 3)."</td></tr>";
}
echo "</table>";
oci_free_statement($stid);
oci_close($con);
}
?>
```

MENAMBAH DAN MENGUBAH DATA

- ✘ Untuk menambah dan mengubah data, tidak memerlukan fungsi OCI khusus.
- ✘ Sebelum mencoba latihan2 berikut, buatlah table **pegawai** pada user **hr** dengan perintah berikut:

```
create table pegawai  
(nama varchar2(34), gaji number);
```

CONTOH MENAMBAH DATA

```
<form name=" kirim" method="post" action="">
  Nama <input type="text" name=" nama">
  Gaji <input type="text" name=" gaji">
  <input type="submit" name="Submit" value="Insert">
</form>
<?php
  $nama=@$_POST[ 'nama' ];
  $gaji=@$_POST[ 'gaji' ];
  if($nama<>"") {
    $con=oci_connect("hr","hr","10.252.44.102/orcl");
    $stid=oci_parse($con,"insert into pegawai
values( '$nama', '$gaji' )");
    oci_execute($stid);
    oci_free_statement($stid);
    oci_close($con);
  }
?>
```

CONTOH MENGUBAH DATA

```
<form name="kirim" method="post" action="">
  Nama <input type="text" name="nama">
  Gaji <input type="text" name="gaji">
  <input type="submit" name="Submit" value="Update Gaji">
</form>
<?php
  $nama=@$_POST['nama'];
  $gaji=@$_POST['gaji'];
  if($nama<>""){
    $con=oci_connect("hr","hr","10.252.44.102/orcl");
    $stid=oci_parse($con,"update pegawai set gaji='$gaji'
where nama='$nama'");
    oci_execute($stid);
    oci_free_statement($stid);
    oci_close($con);
  }
?>
```


FUNGSI2 OCI UNTUK PHP VERSI LAMA

- ✘ Untuk PHP dengan versi sebelum 5.0.0, gunakan alias berikut sebagai pengganti fungsi2 PHP pada contoh2 didepan:

+ `oci_connect()` → `ocilogon()`

+ `oci_close()` → `ocilogoff()`

+ `oci_parse()` → `ociparse()`

+ `oci_free_statement()` → `ocifreecursor()`

+ `oci_execute()` → `ociexecute()`

+ `oci_fetch()` → `ocifetch()`

+ `oci_result()` → `ociresult()`

REFERENSI

- ✘ Oracle website: www.oracle.com
- ✘ PHP website: www.php.net
- ✘ Apache website: www.apache.org
- ✘ XAMPP website: www.apachefriends.org/en/xampp.html
- ✘ Daftar lengkap fungsi2 dari php_oci8.dll
www.php.net/manual/en/book.oci8.php