

Template Akses Data MySql Menggunakan PHP

Eko Heri Susanto

ekoheri@gmail.com

<http://www.jawadwipa.com>

Lisensi Dokumen:

Copyright © 2007 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Dalam tulisan kali ini penulis mencoba membahas bagaimana membuat template untuk mengakses database MySql menggunakan PHP. Adapun tujuan dari penulisan ini adalah untuk membuat sebuah class yang berfungsi sebagai template agar supaya aplikasi tersebut lebih mudah untuk dikembangkan dan di-maintenance.

Pada dasarnya operasi manipulasi data atau yang lebih dikenal dengan istilah transact SQL, sebenarnya yang paling sering digunakan hanya ada empat jenis yaitu : SELECT, INSERT, UPDATE dan DELETE. Dimana SELECT berfungsi untuk mengambil data dari database, INSERT untuk menambah data baru, UPDATE untuk merubah atau mengedit data, sedangkan DELETE berfungsi untuk menghapus data. Jika kita amati, sebenarnya keempat perintah itulah yang terus menerus digunakan ketika kita sedang mengembangkan sebuah aplikasi. Sebagai ilustrasi, misalkan kita mempunyai sebuah tabel biodata mahasiswa dengan field-field sebagai berikut :

```
CREATE TABLE BIODATA_MHS
(
  NO_INDUK VARCHAR(12) NOT NULL,
  NAMA VARCHAR(50) NOT NULL,
  FAKULTAS VARCHAR(30) NOT NULL,
  JURUSAN VARCHAR(30) NOT NULL,
  TGL_MASUK DATETIME NOT NULL
);
```

Jika kita akan menambah data baru sebanyak tiga buah ke tabel biodata mahasiswa tersebut, tentunya kita akan menggunakan perintah INSERT sebanyak tiga kali pula, contoh :

```
INSERT INTO BIODATA_MHS(NO_INDUK, NAMA, FAKULTAS, JURUSAN, TGL_MASUK) VALUES ('001',
'EKO HERI', 'TI', 'TEHNIK INFORMATIKA','2007-07-01');
INSERT INTO BIODATA_MHS(NO_INDUK, NAMA, FAKULTAS, JURUSAN, TGL_MASUK) VALUES ('002',
'FERY AGUNG', 'TI', 'MANAJEMEN INFORMATIKA','2007-07-01');
INSERT INTO BIODATA_MHS(NO_INDUK, NAMA, FAKULTAS, JURUSAN, TGL_MASUK) VALUES ('003',
'BUDI HARYANTO, 'TI', 'ROBOTIKA','2007-07-01');
```

Dari contoh diatas sebenarnya transact SQL tersebut dipanggil secara berulang-ulang dan tentu saja jika jumlah field-nya semakin banyak maka transact SQL tersebut akan semakin panjang dan akan susah untuk dikembangkan lebih lanjut. Hal yang sama juga berlaku untuk transact SQL UPDATE dan DELETE. Berawal dari kasus inilah penulis mencoba mengembangkan sebuah class template yang berfungsi untuk memenuhi semua kebutuhan manipulasi data seperti SELECT, INSERT, UPDATE dan DELETE tersebut.

Sebagai sarana eksperimen, langkah pertama yang perlu kita lakukan adalah menyiapkan database beserta tabel-tabelnya. Pada contoh tulisan kali ini, penulis membuat sebuah database dengan nama mydb, sedangkan tabel yang disediakan adalah biodata_mhs seperti yang telah disebutkan diatas.

Template Select

Setelah database dan tabelnya selesai kita buat, maka langkah selanjutnya adalah kita akan membuat script yang berfungsi untuk mengambil data (SELECT) dari database MySQL. Adapun function untuk mengambil data tersebut dinamakan function getData. Script selengkapnya bisa dilihat dibawah ini:

```
01: <?php
02: class dataAccess
03: {
04:     var $query;
05:     var $servername;
06:     var $databasename;
07:     var $username;
08:     var $password;
09:     function set_query($value){
10:         $this->query=$value;
11:     }
12:     function get_query(){
13:         return $this->query;
14:     }
15:     function initConnection(){
16:         //setting connection on localhost
17:         $this->servername="localhost";
18:         $this->databasename="mydb";
19:         $this->username="root";
20:         $this->password="";
21:     }
22:     function getData(){
23:         $this->initConnection();
24:         $connection = mysql_connect($this->servername, $this->username, $this->password)
                or die("Tidak dapat konek ke $this->servername ....");
25:         mysql_select_db($this->databasename, $connection) or
                die("Tidak dapat membuka database $this->databasename ....");
26:         $datatable = mysql_query($this->query) or die(mysql_error());
27:         mysql_close($connection);
28:         return $datatable;
29:     }
30: }//end class
31: ?>
```

Berikut ini contoh penggunaan function getData tersebut diatas

```
01: $db=new dataAccess();
02: $db->set_query("SELECT * FROM `biodata_mhs` LIMIT 0 , 30");
03: $hasil=$db->getData();
04: echo "<table>
05:     <tr>
06:         <td>No.Induk</td>
07:         <td>Nama</td>
08:         <td>Fakultas</td>
09:         <td>Jurusan</td>
10:     </tr>";
11: while($hasilrow = mysql_fetch_array($hasil))
12: {
13:     echo "<tr>
14:         <td>$hasilrow[0]</td>
15:         <td>$hasilrow[1]</td>
16:         <td>$hasilrow[2]</td>
17:         <td>$hasilrow[3]</td>
18:     </tr>";
19: }
20: echo "</table>";
21: mysql_free_result($hasil);
```

Template Insert, Update dan Delete

Setelah template select tersebut selesai kita buat, maka selanjutnya kita akan membuat script template insert, update dan delete data. Sebagai contoh kasus, diasumsikan kita mempunyai sebuah tabel dimana dalam tabel tersebut terdapat data-data seperti terlihat dibawah ini :

No	Kode	Nama	Fakultas	Jurusan
1	001	Eko Heri Susanto	TI	Teknik Informatika
2	002	Fery Agung	TI	Manajemen Informatika
3	003	Budi Haryanto	TI	Robotika
...

Dari data-data yang terdapat dalam tabel seperti tersebut diatas, kemudian apa yang harus kita lakukan? Logika program kita akan dibawa kemana? Pertama yang harus kita bayangkan adalah bagaimana membuat script untuk menyimpan data-data tersebut kedalam table, dimana nantinya tabel tersebut akan tersimpan di memory komputer kita. Setelah data-data tersebut tersimpan di memory komputer, hal yang perlu kita bayangkan lagi adalah bagaimana menyusun data-data tersebut menjadi transact SQL yang dikenali oleh MySQL.

Ketika melihat gambaran model data tersebut diatas, hampir bisa dipastikan logika anda akan mengarah pada penggunaan array multi dimensi. Jika logika anda mengarah ke penggunaan array dapat dipastikan alur logika anda sama dengan alur logika penulis. Oleh karena itu dalam contoh kali ini penulis juga menggunakan array untuk menyimpan data-data tersebut. Setelah array tersebut selesai kita deklarasikan maka selanjutnya kita tinggal membuat fungsi untuk mengisi data kedalam array tersebut. Adapun deklarasi array dan function tersebut, bisa anda lihat seperti pada contoh dibawah ini :

```
1: var $datafields=array();
2: function addItem($index, $fields, $values, $keys, $isdatetime){
3:   $this->datafields[$index][0]=$fields;
4:   $this->datafields[$index][1]=$values;
5:   $this->datafields[$index][2]=$keys;
6:   $this->datafields[$index][3]=$isdatetime;
7: }
```

Pada contoh script diatas, function addItem mempunyai empat buah parameter yaitu, \$index, \$fields, \$value, \$keys dan \$isdatetime. Berikut ini penjelasan dari masing-masing parameter tersebut.

Parameter	Keterangan
\$index	Digunakan untuk menyimpan urutan data yang mana parameter ini nantinya akan berisi angka 0, 1, 2 dst.
\$fields	Digunakan untuk menyimpan nama-nama field seperti kode, nama, fakultas, jurusan dsb.
\$value	Digunakan untuk menyimpan data itu sendiri, misal '001', 'Eko Heri' dsb.
\$keys	Digunakan untuk menyimpan data filter yang biasanya digabungkan dengan transact WHERE atau HAVING. Parameter ini nantinya banyak digunakan untuk update dan delete.
\$isdatetime	Digunakan sebagai tanda (flag) bahwa data yang tersimpan tersebut adalah berupa tanggal dan jam (datetime).

Setelah deklarasi array dan function addItem tersebut selesai kita buat, maka langkah selanjutnya adalah menyusun data-data array tersebut menjadi transact SQL. Contoh script penyusunan transact SQL tersebut bisa dilihat dibawah ini

```
00:var $dataquery=array();
01:function buildQuery($tables, $method){
02:  $sql="";
03:  switch($method){
04:    case "INS":
05:      $datasql="";$valuesql="";
06:      foreach($this->datafields as $values){
07:        if($datasql=="")
08:          $datasql="INSERT INTO $tables ($values[0]";
09:        else
```

```
10:     $datasql.=", $values[0]";
11: } //end for
12: foreach($this->datafields as $values){
13:     if($valuesql==""){
14:         $x=$this->getDataTypes($values[1], $values[3]);
15:         $valuesql=") VALUES ($x";
16:     }
17:     else{
18:         $x=$this->getDataTypes($values[1], $values[3]);
19:         $valuesql.=", $x";
20:     }
21: } //end for
22: $sql="$datasql$valuesql) ; ";
23: break;
24: case "UPD":
25:     $datasql=""; $valuesql="";
26:     foreach($this->datafields as $values){
27:         if(gettype($values[1])!="NULL")
28:         {
29:             if($datasql==""){
30:                 $x=$this->getDataTypes($values[1], $values[3]);
31:                 $datasql="UPDATE $tables SET $values[0]=$x";
32:             }
33:             else {
34:                 $x=$this->getDataTypes($values[1], $values[3]);
35:                 $datasql.=", $values[0]=$x";
36:             }
37:         } //end if
38:     } //end for
39:     foreach($this->datafields as $values){
40:         if(gettype($values[2])!="NULL")
41:         {
42:             if($valuesql==""){
43:                 $x=$this->getDataTypes($values[2], $values[3]);
44:                 $valuesql=" WHERE ($values[0]=$x)";
45:             }
46:             else {
47:                 $x=$this->getDataTypes($values[2], $values[3]);
48:                 $valuesql.=" AND ($values[0]=$x)";
49:             }
50:         } //end if
51:     } //end for
52:     $sql="$datasql$valuesql ; ";
53: break;
54: case "DEL":
55:     $valuesql="";
56:     foreach($this->datafields as $values){
57:         if(gettype($values[2])!="NULL")
58:         {
59:             if($valuesql==""){
60:                 $x=$this->getDataTypes($values[2], $values[3]);
61:                 $valuesql="DELETE FROM $tables WHERE ($values[0]=$x)";
62:             }
63:             else {
64:                 $x=$this->getDataTypes($values[2], $values[3]);
65:                 $valuesql.=" AND ($values[0]=$x)";
66:             }
67:         } //end if
68:     } //end for
69:     $sql="$datasql$valuesql ; ";
70: break;
71: } //end switch
72: $index=count($this->dataquery);
73: foreach($this->datafields as $item) unset($item);
74: $this->dataquery[$index]=$sql;
75: return;
76: } //end function
```

Pada contoh script diatas, dalam function buildQuery tersebut terdapat dua buah parameter yaitu, \$tables, \$method. Parameter \$tables berfungsi untuk menyimpan nama tabel dalam database MySql, contoh tabel biodata_mhs. Sedangkan \$method adalah tanda (flag) operasi apa yang akan kita gunakan. Parameter

\$method tersebut nantinya akan berisi string 'INS' yang berarti script akan men-generate transact SQL insert, 'UPD' untuk update data dan 'DEL' untuk menghapus (delete) data.

Pada baris ke 00 seperti yang terlihat pada contoh diatas, terdapat deklarasi array var \$dataquery=array(). Deklarasi variable array \$dataquery ini berfungsi untuk menyimpan string transact SQL. Deklarasi array ini sengaja digunakan karena sangat dimungkinkan dalam sekali proses eksekusi data, terdapat lebih dari satu transact SQL yang akan diproses oleh MySql secara bersamaan. Untuk selanjutnya proses penyimpanan transact SQL ke array \$dataquery tersebut bisa dilihat di baris ke 74 pada contoh diatas.

Pada baris ke 14, 18, 30 dan seterusnya seperti yang tertera pada contoh script diatas, terdapat pemanggilan function `getDataType($values[1], $values[3])`. Function ini berfungsi untuk mengenali tipe data apa yang akan diinputkan kedalam database. Adapun script function `getDataType` tersebut dapat anda lihat seperti dibawah ini :

```
01: function getDataType($Data, $IsDateTime)
02: {
03:     $datareturn="";
04:     $Data=trim($Data);
05:     switch(gettype($Data)) {
06:         case "NULL": $datareturn="NULL"; break;
07:         case "object": $datareturn="NULL"; break;
08:         case "array": $datareturn="NULL"; break;
09:         case "string":
10:             if(!$IsDateTime) {
11:                 $x=str_replace("'", "", $Data);
12:                 $datareturn="'$x'";
13:             }
14:             else {
15:                 switch($Data)
16:                 {
17:                     case "NOW()" :
18:                         $datareturn="NOW() ";
19:                         break;
20:                     case "CURDATE()" :
21:                         $datareturn="CURDATE() ";
22:                         break;
23:                     default :
24:                         $x=substr($Data, 6, 4)."-".substr($Data, 3, 2)."-".
25:                             substr($Data, 0, 2). " 00:00:00";
26:                         $datareturn="'$x'"; // '2007-05-28 00:00:00'
27:                         break;
28:                 } //end switch
29:             } //end else
30:             break;
31:         case "boolean":
32:             if($Data==true) $datareturn="TRUE";
33:             else
34:                 if($Data==false) $datareturn="FALSE";
35:             break;
36:         case "integer": $datareturn=$Data; break;
37:         case "double": $datareturn=$Data; break;
38:         case "float": $datareturn=$Data; break;
39:         default : $datareturn="NULL"; break;
40:     } //end switch
41:     return $datareturn;
42: } //end function
```

Setelah langkah-langkah tersebut diatas selesai dilakukan, maka langkah terakhir adalah mengeksekusi transact SQL. Berikut ini contoh function yang berfungsi untuk mengeksekusi transact SQL tersebut.

```
01: function execQuery() {
02:     $sql="";
03:     foreach($this->dataquery as $values) {
04:         $sql.=$values;
05:     }
06:     foreach($this->dataquery as $item) unset($item);
07:     $this->query=$sql;
08:     $this->initConnection();
09:     $connection = mysql_connect($this->servername, $this->username,
```

```
$this->password) or die("Tidak dapat konek ke $this->servername");
10: mysql_select_db($this->databasename, $connection) or
    die("Tidak dapat membuka database $this->databasename");
11: mysql_query("BEGIN;");
12: $result=mysql_query($sql);// or die(mysql_error());
13: if($result)
14:     mysql_query("COMMIT;");
15: else
16:     mysql_query("ROLLBACK;");
17: mysql_close($connection);
18: return $result;
19: }//end function
```

Adapun cara penggunaan function template insert, update dan delete tersebut diatas, dapat anda lihat seperti contoh dibawah ini :

```
//Contoh Insert Data
01: $db=new dataAccess();
02: $db->addItem(0, 'KODE', '001', NULL, false);
03: $db->addItem(1, 'NAMA', 'Eko Heri S', NULL, false);
04: $db->addItem(2, 'FAKULTAS', 'IT', NULL, false);
05: $db->addItem(3, 'JURUSAN', 'TEHNIK INFORMATIKA', NULL, false);
06: $db->addItem(5, 'TGL MASUK', 'NOW()', NULL, true);
07: $db->buildQuery("biodata_mhs","INS");
08: if( !$db->execQuery() )
09:     echo "Querynya Error, boss !!!";
10: else
11:     echo "Sukses Melakukan Insert Data Baru";

//Contoh Update Data
01: $db=new dataAccess();
02: $db->addItem(0, 'KODE', NULL, '001', false);
03: $db->addItem(1, 'NAMA', 'Eko Heri Susanto', NULL, false);
04: $db->addItem(2, 'FAKULTAS', 'TEHNIK INFORMATIKA', NULL, false);
05: $db->buildQuery("biodata_mhs","UPD");
06: if( !$db->execQuery() )
07:     echo "Querynya Error, boss !!!";
08: else
09:     echo "Sukses Melakukan Update Data";

//Contoh Delete Data
01: $db=new dataAccess();
02: $db->addItem(0, 'KODE', NULL, '001', false);
03: $db->buildQuery("biodata_mhs","DEL");
04: if( !$db->execQuery() )
05:     echo "Querynya Error, boss !!!";
06: else
07:     echo "Sukses Melakukan Delete Data";
```

Dari contoh penggunaan template diatas, sebenarnya transact SQL yang diproses oleh MySQL adalah seperti berikut ini:

```
Transact SQL Insert Data :
INSERT INTO biodata_mhs(KODE, NAMA, FAKULTAS, JURUSAN, TGL_MASUK) VALUES ('001','Eko
Heri S','TI','TEHNIK INFORMATIKA', NOW());

Transact SQL Update
UPDATE biodata_mhs SET NAMA='Eko Heri Susanto', FAKULTAS='TEHNIK INFORMATIKA' WHERE
(KODE='001');

Transact SQL Delete
DELETE FROM biodata_mhs WHERE (KODE='001');
```

Contoh lengkap dari class template manipulasi data ini, bisa anda lihat seperti dibawah ini :

```
<?php
class dataAccess
{
    var $query;
    var $servername;
    var $databasename;
    var $username;
    var $password;
    var $datafields=array();
    var $dataquery=array();

    function set_query($value){
        $this->query=$value;
    }
    function initConnection(){
        //setting connection on my local computer
        $this->servername="localhost";
        $this->databasename="mydb";
        $this->username="root";
        $this->password="";
    }

    function getData(){
        $this->initConnection();
        $connection = mysql_connect($this->servername, $this->username, $this->password) or
        die("Tidak dapat konek ke $this->servername ....");
        mysql_select_db($this->databasename, $connection) or die("Tidak dapat membuka
        database $this->databasename ....");
        $datatable = mysql_query($this->query) or die(mysql_error());
        mysql_close($connection);
        return $datatable;
    }

    function addItem($index, $fields, $values, $keys, $isdatetime){
        $this->datafields[$index][0]=$fields;
        $this->datafields[$index][1]=$values;
        $this->datafields[$index][2]=$keys;
        $this->datafields[$index][3]=$isdatetime;
    }

    function buildQuery($tables, $method){
        $sql="";
        switch($method){
            case "INS":
                $datasql=""; $valuesql="";
                foreach($this->datafields as $values){
                    if($datasql=="")
                        $datasql="INSERT INTO $tables ($values[0]";
                    else
                        $datasql.=", $values[0]";
                } //end for
                foreach($this->datafields as $values){
                    if($valuesql==""){
                        $x=$this->getDataType($values[1], $values[3]);
                        $valuesql=") VALUES ($x";
                    }
                    else{
                        $x=$this->getDataType($values[1], $values[3]);
                        $valuesql.=", $x";
                    }
                } //end for
                $sql="$datasql$valuesql) ; ";
                break;
            case "UPD":
                $datasql=""; $valuesql="";
                foreach($this->datafields as $values){
                    if(gettype($values[1])!="NULL")
                    {
                        if($datasql==""){
                            $x=$this->getDataType($values[1], $values[3]);
                            $datasql="UPDATE $tables SET $values[0]=$x";
                        }
                    }
                }
            }
        }
    }
}
```

```
        else {
            $x=$this->getDataType($values[1], $values[3]);
            $datasql.=" , $values[0]=$x";
        }
    } //end if
} //end for
foreach($this->datafields as $values){
    if(gettype($values[2])!="NULL")
    {
        if($valuesql==""){
            $x=$this->getDataType($values[2], $values[3]);
            $valuesql=" WHERE ($values[0]=$x)";
        }
        else {
            $x=$this->getDataType($values[2], $values[3]);
            $valuesql.=" AND ($values[0]=$x)";
        }
    } //end if
} //end for
$sql="$datasql$valuesql ; ";
break;
case "DEL":
    $valuesql="";
    foreach($this->datafields as $values){
        if(gettype($values[2])!="NULL")
        {
            if($valuesql==""){
                $x=$this->getDataType($values[2], $values[3]);
                $valuesql="DELETE FROM $tables WHERE ($values[0]=$x)";
            }
            else {
                $x=$this->getDataType($values[2], $values[3]);
                $valuesql.=" AND ($values[0]=$x)";
            }
        }
    } //end if
} //end for
$sql="$datasql$valuesql ; ";
break;
} //end switch

$index=count($this->dataquery);
foreach($this->datafields as $item) unset($item);
$this->dataquery[$index]=$sql;
return;
} //end function

function getDataType($Data, $IsDateTime)
{
    $datareturn="";
    $Data=trim($Data);
    switch(gettype($Data)){
        case "NULL":$datareturn="NULL";break;
        case "object":$datareturn="NULL";break;
        case "array":$datareturn="NULL";break;
        case "string":
            if(!$IsDateTime){
                $x=str_replace("'", "", $Data);
                $datareturn="'$x'";
            }
            else {
                switch($Data)
                {
                    {
                        case "NOW()" :
                            $datareturn="NOW()";
                            break;
                        case "CURDATE()" :
                            $datareturn="CURDATE()";
                            break;
                        default :
                            $x=substr($Data, 6, 4)."-".substr($Data,3, 2)."-".
                                substr($Data,0, 2). " 00:00:00";
                            $datareturn="'$x'"; // '2007-05-28 00:00:00'
                            break;
                    }
                }
            }
        }
    } //end switch
}
```



```
        }//end else
        break;
    case "boolean":
        if($Data==true)$datareturn="TRUE";
    else
        if($Data==false)$datareturn="FALSE";
        break;
    case "integer":$datareturn=$Data;break;
    case "double":$datareturn=$Data;break;
    case "float":$datareturn=$Data;break;
    default :$datareturn="NULL";break;
    }//end switch
    return $datareturn;
}//end function

function execQuery(){
    $sql="";
    foreach($this->dataquery as $values){
        $sql.=$values;
    }
    foreach($this->dataquery as $item) unset($item);
    $this->query=$sql;

    $this->initConnection();
    $connection = mysql_connect($this->servername, $this->username, $this->password) or
    die("Tidak dapat konek ke $this->servername");
    mysql_select_db($this->databasename, $connection) or die("Tidak dapat membuka
    database $this->databasename");
    mysql_query("BEGIN;");
    $result=mysql_query($sql);// or die(mysql_error());

    if($result)
        mysql_query("COMMIT;");
    else
        mysql_query("ROLLBACK;");

    mysql_close($connection);
    return $result;
}//end function
}//end class
?>
```

Demikian artikel sederhana ini dibuat, semoga bermanfaat bagi kita semua. Penulis sangat menyadari bahwa artikel ini masih sangat jauh dari kata sempurna oleh karena itu penulis sangat mengharapkan kritik dan saran dari pembaca yang budiman demi kesempurnaan artikel ini maupun artikel-artikel penulis selanjutnya.

REFERENCES

1. <http://www.w3schools.com/ajax/default.asp>
2. <http://www.php.net>
3. <http://www.bengkelprogram.com>

BIOGRAFI PENULIS



Eko Heri Susanto. Putra daerah Pacitan Jawa Timur ini lahir pada tanggal 5 Maret 1979. Menamatkan pendidikan sekolah dasar di SDN Tahunan II Tegalombo Pacitan Jatim pada tahun 1991. Menamatkan sekolah menengah tingkat pertama di SLTPN I Ponorogo Jatim pada tahun 1994. Menamatkan sekolah menengah tingkat atas di SMUN II Ponorogo Jatim pada tahun 1997. Menyelesaikan study S1 di Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) Malang Jatim, pada tahun 2002.

Semasa kuliah, penulis berkecimpung dalam sebuah organisasi Unit Kegiatan Mahasiswa (UKM) STIKI Computer Networking Club (SCeN) “Kampus ELANG” STIKI Malang.

Dalam kesehariannya penulis berprofesi sebagai programmer, adapun bahasa pemrograman maupun tools lain yang digeluti penulis saat ini adalah Visual Basic 6, Microsoft .Net, ASP, PHP, Java, SQL Server, MySql.

Saat ini penulis bekerja sebagai programmer di PT. Modulo Lima Belas, Cikarang Bekasi, Jawa Barat.

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:

URL: <http://www.jawadwipa.com>

Email: ekoheri@gmail.com