

## BAB 1

# **Trik Dasar Memahami Pemrograman PHP & MySQL**

Pada bab ini, kami akan menunjukkan kepada Anda bagaimana PHP dapat mengolah data dari suatu database dengan mudah dan cepat. Secara garis besar, ada dua hal yang harus betul-betul Anda pahami untuk menguasai cara pengolahan data dengan PHP ini, yaitu:

1. Kuasai bahasa SQL (Structured Query Language) dengan baik, karena bahasa ini merupakan bahasa standar yang dipakai dalam mengolah semua database. Beberapa perintah yang sering digunakan adalah: SELECT, INSERT, UPDATE dan DELETE.
2. Pelajari dan ketahui fungsi-fungsi yang telah disediakan PHP dalam mengakses database. Beberapa fungsi PHP dalam mengakses database MySQL yang sering digunakan adalah: `mysql_connect`, `mysql_select_db`, `mysql_query`, `mysql_fetch_array` dan `mysql_num_rows`.

Gampang kan?? Oke, sekarang agar Anda benar-benar menjadi ahli dalam mengolah database di PHP, maka banyaklah berlatih, pasti jadi jago PHP.

Pada bab ini, kami telah memilihkan kasus yang paling sederhana agar Anda mudah dalam memahami pengolahan database di PHP, yaitu Guest Book atau Buku Tamu. Selamat Mengikuti.

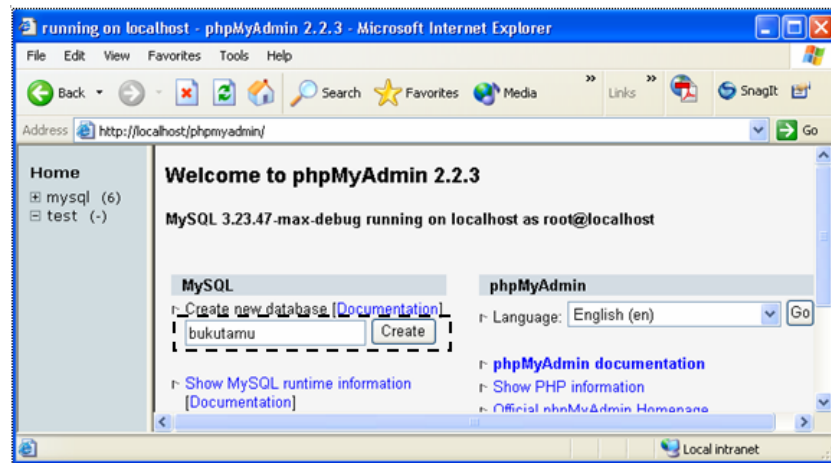
### **Trik. Cara Cepat Membuat Database dan Tabel**

Saat ini membuat database MySQL tidak perlu menggunakan cara manual lagi dengan mengetikkan perintah-perintah SQL, karena sekarang sudah banyak tool-tool visual yang membantu kita untuk membuat database dengan mudah dengan cepat, salah satunya adalah tool phpMyAdmin, tool ini merupakan salah satu

paket dari PHPTriad, jadi apabila Anda menginstall PHPTriad, maka secara otomatis di komputer Anda sudah terinstall juga phpMyAdminnya.

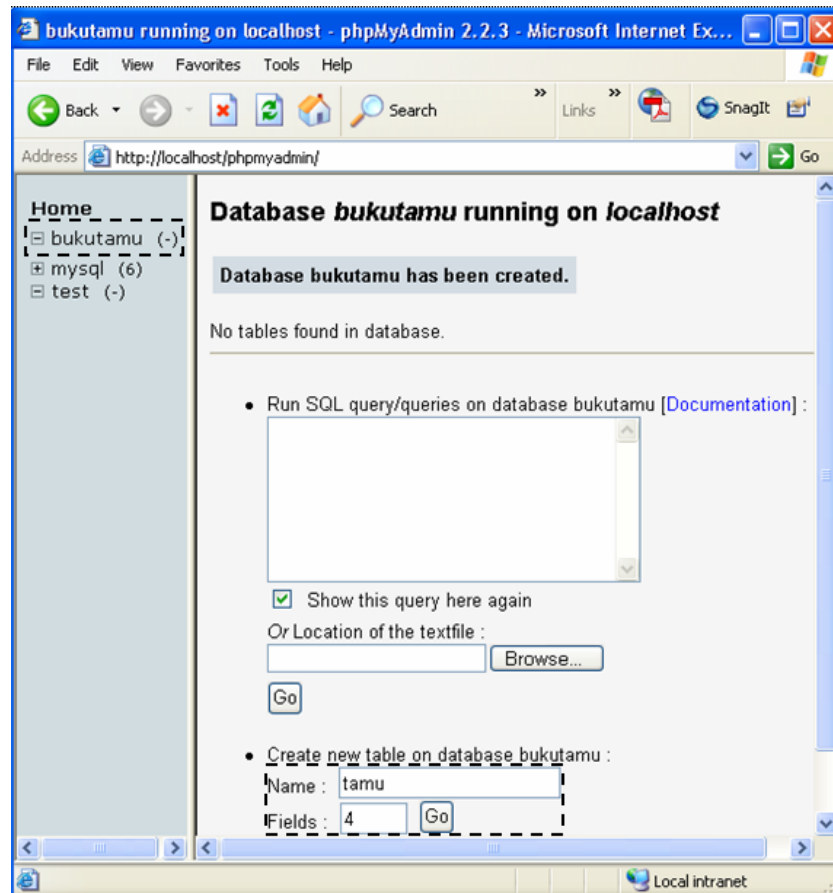
Langsung saja, mari kita praktekkan membuat database dan tabel di phpMyAdmin, ikuti langkah-langkah berikut:

1. Buka browser Anda (Internet Explorer), lalu ketikkan pada bagian **Address**: <http://localhost/phpmyadmin/>, lalu tekan tombol **Enter**, maka akan tampil halaman selamat datang dari phpMyAdmin. Lihat gambar.



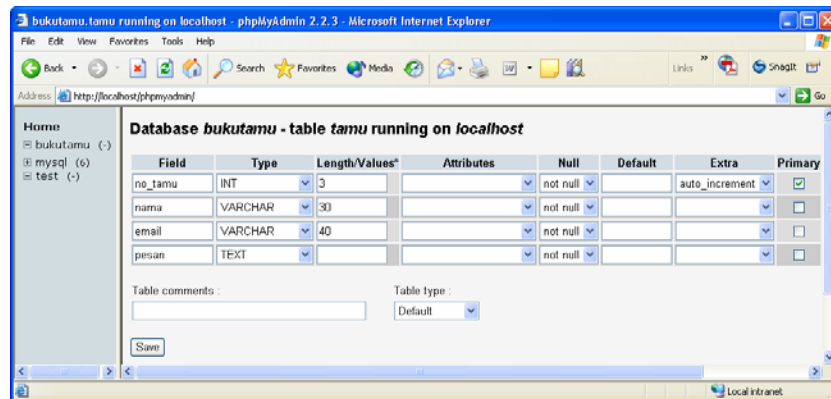
*Gambar. Halaman selamat datang phpMyAdmin*

2. Untuk membuat database, isikan nama database yang akan dibuat pada bagian **Create new database**, misalnya: **bukutamu** (tidak boleh ada spasi), lalu klik tombol **Create**. Lihat lagi gambar.
3. Maka akan terbentuk sebuah database dengan nama: **bukutamu**, Anda bisa melihatnya pada bagian jendela kiri di bawah teks **Home**. Setelah membuat database, maka langkah selanjutnya adalah membuat tabel di dalam database tersebut dengan cara mengisi nama tabel dan jumlah field di bagian **Create new table on database bukutamu**, misalnya **Name**: **tamu** dan **Fields**: **4**, lalu klik tombol **Go**. Lihat gambar.



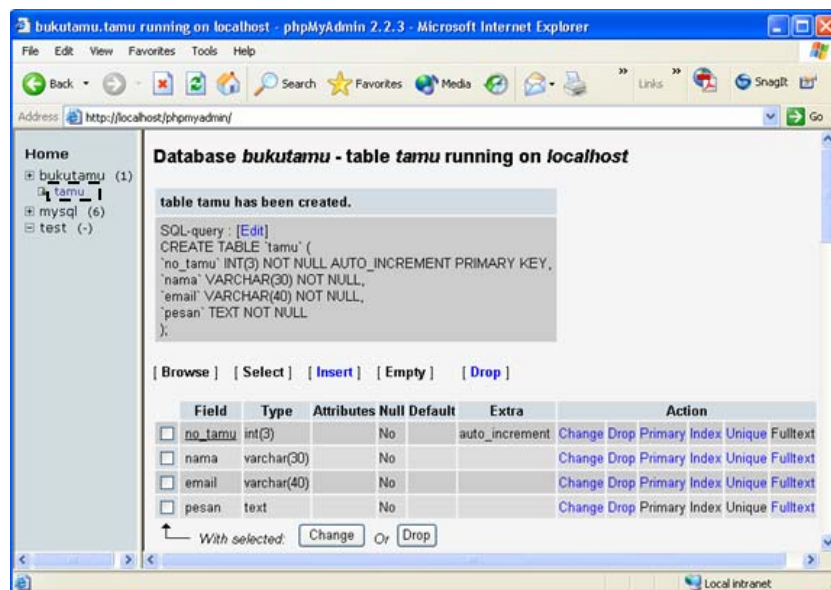
*Gambar. Membuat tabel*

4. Kemudian isikan field-field beserta atribut lainnya seperti pada gambar.



*Gambar. Mengisikan field-field dan atribut lainnya*

- Setelah selesai, klik tombol **Save**, maka akan terbentuk sebuah tabel dengan nama: tamu beserta field-field didalamnya. Hasilnya dapat dilihat pada gambar.



*Gambar. Tampilan tabel tamu beserta field-fieldnya*

Tipe-tipe data yang sering digunakan adalah:

- **INT**, untuk menyatakan data berupa angka.
- **VARCHAR**, untuk menyatakan data berupa karakter atau string (batasnya sampai 255 karakter).
- **DATE**, untuk menyatakan data berupa tanggal.
- **TIME**, untuk menyatakan data berupa waktu.
- **TEXT**, untuk menyatakan data berupa karakter dalam jumlah yang banyak (lebih dari 255 karakter), misalnya untuk isi berita dan pesan.

*Tips:*

- Sebaiknya **gunakan huruf kecil semua** dalam memberikan nama database, tabel beserta field-fieldnya. Hal ini akan memudahkan kita nantinya dalam pengolahan data di PHP.
- Sebaiknya sertakan selalu **nomor unik atau kunci primer** setiap membuat sebuah tabel. Hal ini berguna untuk membedakan antara data yang satu dengan data lainnya.
- Pada tabel tamu yang telah kita buat diatas, kita telah menentukan no\_tamu sebagai Primary key (kunci primer) dengan tambahan atribut Extra: **auto\_increment**, artinya no\_tamu akan terisi dengan angka yang selalu bertambah secara otomatis (tanpa diisi secara manual) seiring dengan bertambahnya data.

## **Trik. Pentingnya Memisahkan File Koneksi**

Untuk bisa mengakses database dan tabel di MySQL, kita harus melakukan koneksi terlebih dahulu agar bisa terhubung ke database tersebut.

Sebaiknya skrip untuk melakukan koneksi dibuat tersendiri dalam sebuah file, karena setiap kali kita akan mengolah data di database, maka kita harus selalu melakukan koneksi terlebih dahulu. Disamping itu yang terlebih penting lagi, hal ini biasanya terjadi ketika kita melakukan proses upload di Internet untuk mengonlinekan situs kita, maka biasanya Username dan Password akan ditentukan oleh web hosting (tempat kita meletakkan file-file situs kita). Sekarang coba bayangkan, apabila Anda mempunyai sepuluh file, maka Anda harus mengganti username dan password sebanyak sepuluh kali, gimana kalau seratus file? Alangkah sangat merepotkan. Nah, dengan memisah file koneksi

tersendiri dalam sebuah file, maka kita hanya perlu melakukan perubahan username dan passwordnya cukup sekali.

Oke, sekarang mari kita buat file koneksinya sebagai berikut:

#### **Skrip koneksi\_bukutamu.php**

```
<?
mysql_connect("localhost","","");
mysql_select_db("bukutamu");
?>
```

#### ***Keterangan skrip:***

- `mysql_connect("hostname","username","password");`

Untuk melakukan koneksi ke server database MySQL.

Biasanya pada komputer lokal kita belum ada username dan password databasenya, jadi dikosongkan saja menjadi:

```
mysql_connect("localhost","","");
```

- `mysql_select_db("nama_database");`

Untuk memilih sebuah database, karena tentunya dalam server database MySQL terdapat banyak database, sehingga kita harus memilih salah satu database yang akan kita aktifkan/gunakan, dalam hal ini adalah database bukutamu.

Untuk menyisipkan/memanggil skrip koneksi.php dalam sebuah skrip lainnya, cukup tuliskan:

```
include "nama_file";
```

Nanti akan kita praktekan pada trik-trik berikutnya.

### **Trik. Proses Input Data**

Proses input data adalah suatu proses untuk memasukkan data-data ke dalam database/tabel. Perintah SQL yang akan digunakan adalah:

```
INSERT INTO nama_tabel (field1,field2,...) VALUES (data1,data2,...);
```

Untuk mempraktekannya, cobalah tuliskan dua file skrip berikut:

#### **Skrip form\_tamu.php**

```
<HTML>
<BODY>
<H3>INPUT BUKU TAMU</H3>
```

```

<FORM METHOD=POST ACTION=input_tamu.php>
Nama: <INPUT TYPE=TEXT NAME="nama"><BR>
e-mail: <INPUT TYPE=TEXT NAME="email"><BR>
Pesan: <BR><TEXTAREA NAME=pesan ROWS=5 COLS=25></TEXTAREA><BR>
<INPUT TYPE=SUBMIT VALUE=Kirim>
</FORM>
</BODY>
</HTML>

```

### Skrip input\_tamu.php

```

<?
include "koneksi_bukutamu.php";

$input="INSERT INTO tamu(nama,email,pesan)
VALUES('$nama','$email','$pesan')";
mysql_query($input);

//Apabila perintah SQL untuk menginput data benar
if ($input)
{
    echo "Proses Input Data Berhasil";
}
else
{
    echo "Proses Input Data Gagal";
}
?>

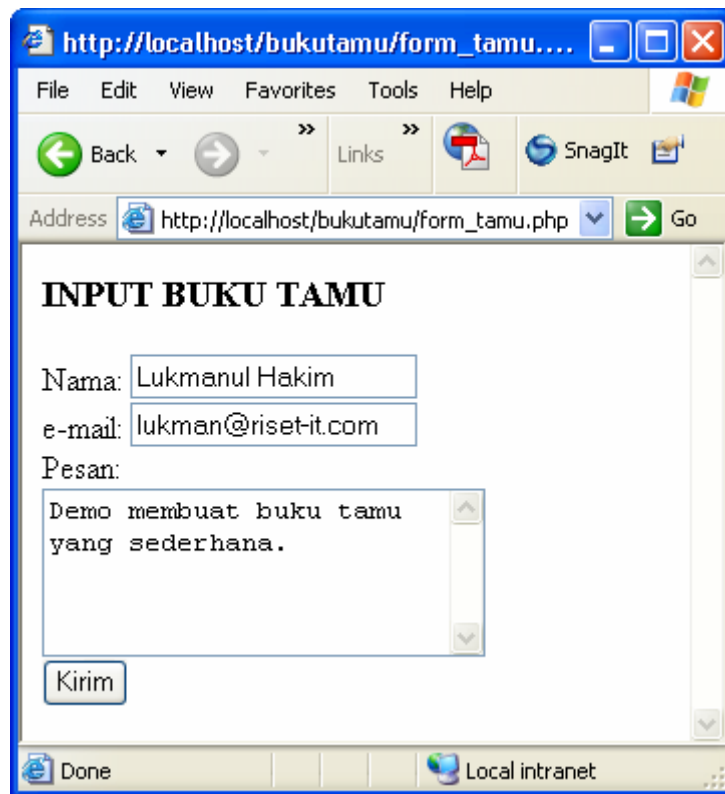
```

### ***Keterangan skrip:***

mysql\_query(“perintah\_SQL”);

Untuk mengirimkan perintah SQL ke server database MySQL untuk dieksekusi.

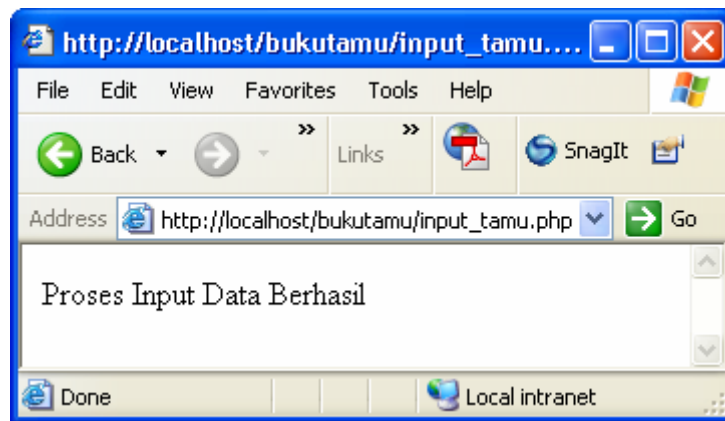
Sekarang coba jalankan skrip form\_tamu.php, lalu isikan data pada bagian Nama, E-mail dan Pesan. Lihat gambar.



*Gambar. Hasil skrip form\_tamu.php*

Kemudian klik tombol **Kirim**, maka data-data yang dimasukkan di form akan diproses oleh skrip input\_tamu.php dan apabila tidak ada kesalahan dalam skrip, maka proses input data berhasil. Lihat gambar.





*Gambar. Hasil skrip input\_tamu.php*

### **Trik. Menampilkan Data dari Database**

Untuk melihat hasil data-data yang telah dimasukkan ke database, maka kita perlu menampilkannya ke browser. Perintah SQL yang akan digunakan adalah:

```
SELECT * FROM nama_tabel;
```

Untuk mempraktekannya, cobalah tuliskan skrip berikut:

#### **Skrip tampil\_tamu\_user.php**

```
<?
echo "<H3>DATA TAMU</H3>";

include "koneksi_bukutamu.php";

$stampil="SELECT * FROM tamu ORDER BY no_tamu DESC";
$hasil=mysql_query($stampil);

while ($data=mysql_fetch_array($hasil))
{
    //Tampilkan nama, email dan pesan di browser
    echo "Nama: $data[nama]<BR>";
    echo "E-mail: $data[email]<BR>";
    echo "Pesan: $data[pesan]<BR><HR>";
}
?>
```

#### **Keterangan skrip:**

- SELECT \* FROM tamu ORDER BY no\_tamu DESC;

Tanda asterik (\*) menyatakan bahwa semua data di database akan ditampilkan dan **ORDER BY** berfungsi untuk mengurutkan data secara **DESC** (dari besar ke kecil, misal 100...1 atau Z...A). Jadi perintah SQL tersebut artinya: “Tampilkan semua data dari tabel tamu, kemudian urutkan berdasarkan no\_tamu secara descending berarti diurutkan berdasarkan pada data terakhir yang diinputkan.

- `mysql_fetch_array("hasil_perintah_SQL");`

Untuk menampilkan hasil dari perintah SQL dalam bentuk array.

Sekarang coba jalankan skrip `tampil_tamu_user.php`, maka hasilnya akan terlihat seperti pada gambar.



*Gambar. Hasil skrip tampil\_tamu\_user.php*

## **Trik. Merapikan Tampilan Data dengan Tabel**

Biasanya kita lebih suka melihat tampilan data dalam bentuk tabel, karena tampilannya lebih rapi dan enak dibaca. Sebenarnya pembuatan tabel menggunakan tag-tag HTML, namun tag-tag tersebut bisa disisipkan di PHP. Berikut beberapa tag-tag HTML yang digunakan dalam membuat tabel:

- **<TABLE> ... </TABLE>**, digunakan untuk membuat tabel, jadi setiap membuat tabel harus berada diantara tag tersebut.
- **<TR> ... </TR>**, digunakan untuk membuat baris (rows) pada suatu tabel.
- **<TD> ... </TD>**, digunakan untuk membuat kolom-kolom (columns) pada suatu tabel.
- **<TH> ... </TH>**, digunakan untuk membuat header atau judul kolom dan judul baris pada suatu tabel.

Lalu bagaimana menerapkan tag-tag tersebut ke dalam PHP, mari kita praktekkan untuk menampilkan data tamu dengan sedikit memodifikasi skrip tampil\_tamu\_user.php (trik) menjadi skrip berikut:

### **Skrip tampil\_tamu\_tabel.php**

```
<?
echo "<H3 ALIGN=CENTER>DATA TAMU</H3>";

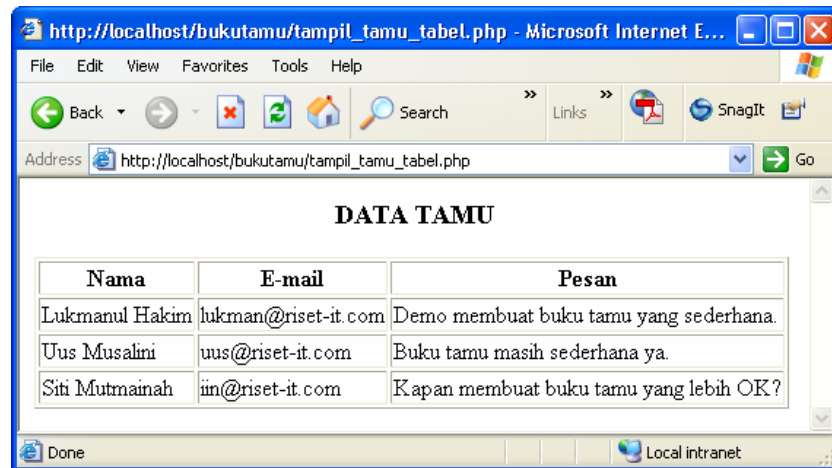
include "koneksi_bukutamu.php";

$stampil="SELECT * FROM tamu ORDER BY no_tamu DESC";
$hasil=mysql_query($stampil);

echo "<TABLE BORDER=1>
<TR>
    <TH>Nama</TH>
    <TH>E-mail</TH>
    <TH>Pesan</TH>
</TR>";

while ($data=mysql_fetch_array($hasil))
{
    echo "<TR><TD>$data[nama]</TD>";
    echo "<TD>$data[email]</TD>";
    echo "<TD>$data[pesan]</TD></TR>";
}
echo "</TABLE>";
?>
```

Sekarang coba jalankan skrip `tampil_tamu_tabel.php`, maka hasilnya akan terlihat seperti pada gambar.



*Gambar. Hasil skrip `tampil_tamu_tabel.php`*

## Trik. Menampilkan Data untuk Admin

Kalau pada trik, kita hanya menampilkan data untuk umum/user, sehingga dapat dilihat semua orang, sedangkan tampilan data untuk keperluan admin harus berbeda, karena ada tambahan tools untuk melakukan pengeditan dan penghapusan data. Disamping itu kita akan menampilkan total keseluruhan data tamu yang sudah terdaftar di database. Untuk lebih jelasnya, cobalah skrip berikut:

### Skrip `tampil_tamu_admin.php`

```
<?
echo "<H3 ALIGN=CENTER>DATA TAMU ADMIN</H3>";

include "koneksi_bukutamu.php";

$stampil="SELECT * FROM tamu ORDER BY no_tamu DESC";
$hasil=mysql_query($stampil);
$total=mysql_num_rows($hasil);

echo "<TABLE ALIGN=CENTER BORDER=1>
<TR>
<TH>Nama</TH>
```

```

        <TH>E-mail</TH>
        <TH>Pesanan</TH>
        <TH>Tools</TH>
    </TR>";

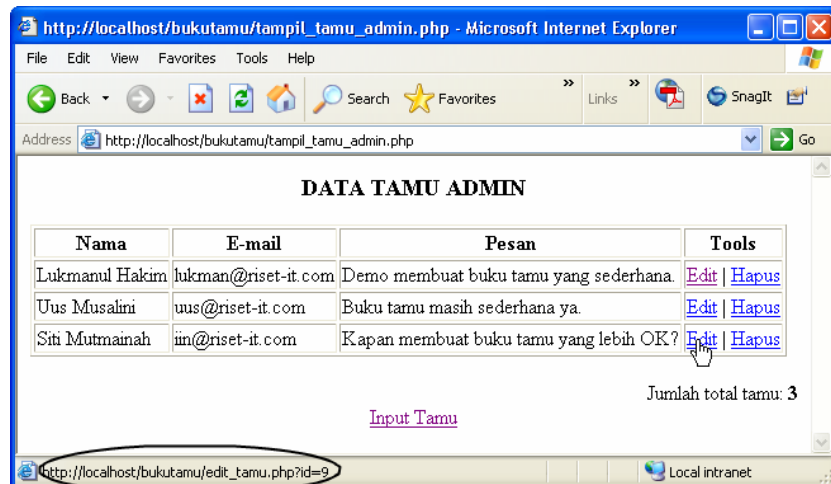
    while ($data=mysql_fetch_array($hasil))
    {
        echo "<TR><TD>$data[nama]</TD>";
        echo "<TD>$data[email]</TD>";
        echo "<TD>$data[pesan]</TD>";
        echo "<TD><A HREF=edit_tamu.php?id=$data[no_tamu]>Edit</A> |";
        echo "<A HREF=hapus_tamu.php?id=$data[no_tamu]>Hapus</A></TD></TR>";
    }
    echo "</TABLE>";
    echo "<P ALIGN=RIGHT>Jumlah total tamu: <B>$total</B>";
    echo "<BR><CENTER><A HREF=form_tamu.php>Input Tamu</A>";
    ?>

```

***Keterangan skrip:***

- `mysql_num_rows("perintah_SQL");`  
Untuk menampilkan/menghitung jumlah total baris data yang ada di database. Hasilnya dalam bentuk angka.
- `<A HREF=edit_tamu.php?id=$data[no_tamu]>Edit</A>`  
Untuk membuat link pada teks Edit dengan tujuan file edit\_tamu.php dan pada link tersebut diberi variabel id yang berisi nilai yang diambilkan dari no\_tamu. Hal tersebut juga berlaku untuk link Hapus.

Sekarang jalankanlah skrip tampil\_tamu\_admin.php, maka hasilnya akan terlihat seperti pada gambar.



*Gambar. Hasil skrip tampil\_tamu\_admin.php*

Cobalah arahkan kursor mouse Anda ke link [Edit](#), lalu lihat pada status bar browser, maka disana akan terlihat tujuan file yaitu edit\_tamu.php dengan id sesuai dengan no\_tamu yaitu 9, dan no\_tamu atau id tersebut tentunya berbeda dengan data-data lainnya, hal ini bisa Anda coba dengan mengarahkan kursor mouse pada link [Edit](#) atau [Hapus](#) di data-data lainnya dan perhatikan perubahan id-nya pada status bar browser.

## **Trik. Melakukan Redirect pada Halaman Web**

Redirect adalah suatu teknik untuk mengalihkan alamat atau halaman web secara otomatis. Adapun perintah untuk melakukan redirect adalah:

```
header("location:nama_file_yg_dituju");
```

Masih bingung? Oke, untuk lebih jelasnya, mari langsung kita praktekkan saja, misalnya pada trik kita telah membuat skrip untuk input data (input\_tamu.php), dimana apabila input tersebut berhasil, maka akan tampil teks "Proses Input Data Berhasil" (lihat lagi gambar), nah sekarang halaman input\_tamu.php akan kita alihkan secara otomatis (redirect) ke halaman tampil\_tamu\_admin.php, sehingga kita bisa langsung melihat hasil data yang baru diinputkan. Untuk mewujudkan hal tersebut, maka kita akan sedikit melakukan modifikasi pada skrip input\_tamu.php menjadi seperti berikut:

### **Skrip input\_tamu.php**

```

<?
include "koneksi_bukutamu.php";

$input="INSERT INTO tamu(nama,email,pesan)
VALUES('$nama','$email','$pesan')";
$hasil=mysql_query($input);

if ($hasil)
{
    //Perintah untuk melakukan redirect
    header("location:tampil_tamu_admin.php");
}
else
{
    echo "Proses Input Data Gagal";
}
?>

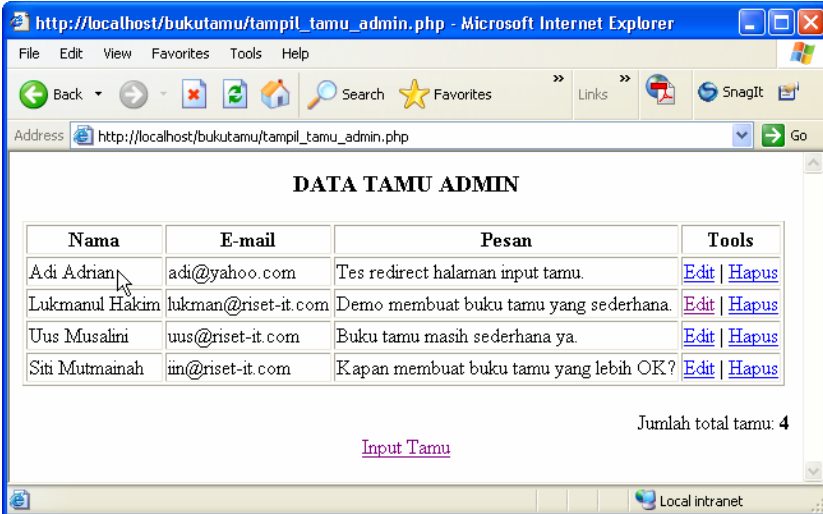
```

Sekarang coba jalankan skrip form\_tamu.php, lalu isikan datanya seperti pada gambar.

The screenshot shows a web browser window with the title "http://localhost/bukutamu/form\_tamu....". The browser's address bar contains "http://localhost/bukutamu/form\_tamu.php". The main content area displays a form titled "INPUT BUKU TAMU". The form includes three input fields: "Nama" (filled with "Adi Adrian"), "e-mail" (filled with "adi@yahoo.com"), and "Pesan" (filled with "Tes redirect halaman input tamu."). Below the "Pesan" field is a "Kirim" button. The browser's status bar at the bottom shows "Done" and "Local intranet".

*Gambar. Hasil skrip form\_tamu.php*

Kemudian klik tombol **Kirim**, maka data-data yang dimasukkan di form akan diproses oleh skrip input\_tamu.php dan apabila proses input data berhasil, maka secara otomatis halaman akan dialihkan (proses pengalihan halaman tidak terlihat di layar, karena prosesnya berjalan sangat cepat) ke halaman Data Tamu Admin (tampil\_tamu\_admin.php) dan hasil input data tersebut langsung dapat dilihat hasilnya seperti pada gambar.



The screenshot shows a web browser window with the address bar displaying 'http://localhost/bukutamu/tampil\_tamu\_admin.php'. The page title is 'DATA TAMU ADMIN'. The table has the following data:

Nama	E-mail	Pesan	Tools
Adi Adrian	adi@yahoo.com	Tes redirect halaman input tamu.	<a href="#">Edit</a>   <a href="#">Hapus</a>
Lukmanul Hakim	lukmanul@riset-it.com	Demo membuat buku tamu yang sederhana.	<a href="#">Edit</a>   <a href="#">Hapus</a>
Uus Musalini	uus@riset-it.com	Buku tamu masih sederhana ya.	<a href="#">Edit</a>   <a href="#">Hapus</a>
Siti Mutmainah	in@riset-it.com	Kapan membuat buku tamu yang lebih OK?	<a href="#">Edit</a>   <a href="#">Hapus</a>

Below the table, it says 'Jumlah total tamu: 4' and there is a link 'Input Tamu'.

*Gambar. Hasil redirect ke halaman Data Tamu Admin.php*

### **Trik. Proses Edit Data**

Proses edit data adalah suatu proses untuk mengubah data-data yang ada dalam database/tabel. Ada dua tahap yang perlu dilakukan dalam melakukan proses edit data ini, pertama dapatkan terlebih dahulu id (kunci primer) dari data yang akan diedit, dalam hal ini adalah no\_tamu, jadi id adalah sebuah variabel yang berisi nilai dari no\_tamu. Adapun perintah SQL yang digunakan adalah:

```
SELECT * FROM nama_tabel WHERE $field_kunci='$id';
```

Setelah mendapatkan no\_tamu yang akan diedit, tampilkan dalam bentuk form data-data yang akan diedit. Kemudian proseslah data-data tersebut untuk diubah (update) menggunakan perintah SQL berikut:



```
UPDATE nama_tabel SET nama_field1='$data1', '$data2','...' WHERE
field_kunci='$id';
```

Untuk mempraktekkannya, cobalah tuliskan dua file skrip berikut:

#### **Skrip edit\_tamu.php**

```
<?
include "koneksi_bukutamu.php";

$edit="SELECT * FROM tamu WHERE no_tamu='$id'";
$hasil=mysql_query($edit);
$data=mysql_fetch_array($hasil);

echo "<H3>EDIT BUKU TAMU</H3>
<FORM METHOD=GET ACTION=update_tamu.php>
<INPUT TYPE=HIDDEN NAME=id VALUE='$id'>
Nama: <INPUT TYPE=TEXT NAME=nama VALUE='$data[nama]'><BR>
e-mail: <INPUT TYPE=TEXT NAME=email VALUE='$data[email]'><BR>
Pesan: <BR>
<TEXTAREA NAME=pesan ROWS=5 OLS=25>$data[pesan]</TEXTAREA><BR>
<INPUT TYPE=SUBMIT VALUE=Edit></FORM>";
?>
```

#### **Skrip update\_tamu.php**

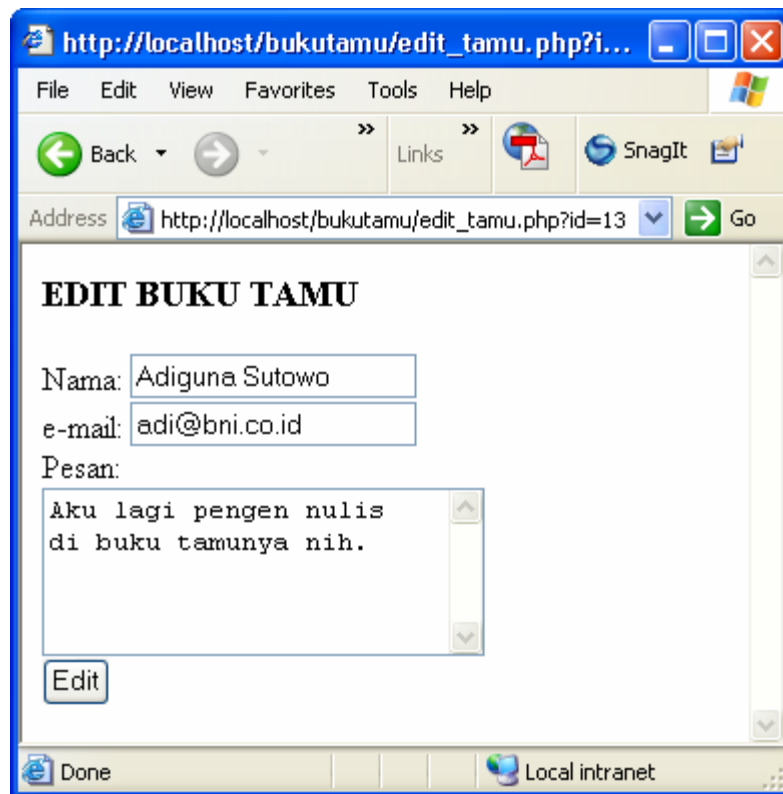
```
<?
include "koneksi_bukutamu.php";

$update="UPDATE tamu SET nama='$nama',
                        email='$email',
                        pesan='$pesan'
                        WHERE no_tamu='$id'";

$hasil=mysql_query($update);

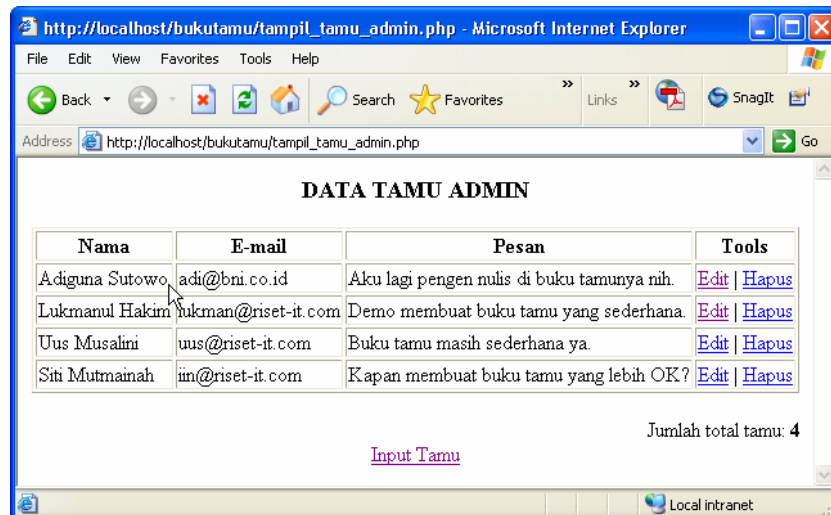
//Apabila perintah SQL untuk mengupdate data benar
if ($hasil)
{
    header("location:tampil_tamu_admin.php");
}
else
{
    echo "Proses Update Data Gagal";
}
?>
```

Sekarang coba jalankan skrip tampil\_tamu\_admin.php, maka akan terlihat tampilannya seperti pada gambar, coba klik link Edit untuk data paling atas (Adi Adrian), maka akan tampil halaman form edit (skrip edit\_tamu.php), kemudian editlah/ubahlah datanya seperti pada gambar.



*Gambar. Hasil skrip edit\_tamu.php*

Kemudian klik tombol **Edit**, maka data-data yang diubah di form akan diproses oleh skrip update\_tamu.php dan apabila proses edit data berhasil, maka secara otomatis halaman akan dialihkan ke halaman Data Tamu Admin dan hasil pengeditan data tersebut langsung dapat dilihat hasilnya seperti pada gambar.



*Gambar. Hasil pengeditan data*

### Trik. Proses Hapus Data

Proses hapus data adalah proses untuk melakukan penghapusan data-data yang ada di database/tabel. Cara kerja proses penghapusan data sama dengan proses edit data yang telah diterangkan sebelumnya (ada dua tahap). Adapun perintah SQL yang akan digunakan adalah:

```
DELETE FROM nama_tabel WHERE field_kunci='$id';
```

Untuk mempraktekkannya, cobalah tuliskan skrip berikut:

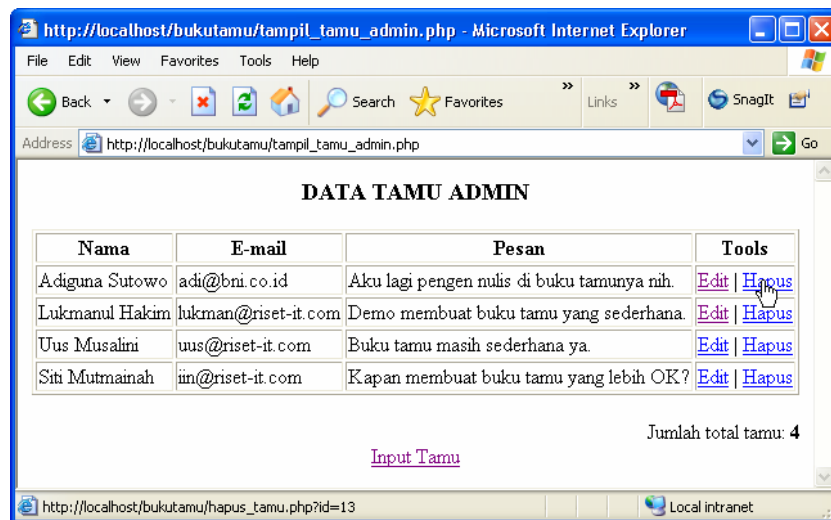
#### Skrip hapus\_tamu.php

```
<?
include "koneksi_bukutamu.php";

$hapus="DELETE FROM tamu WHERE no_tamu='$id'";
$hasil=mysql_query($hapus);

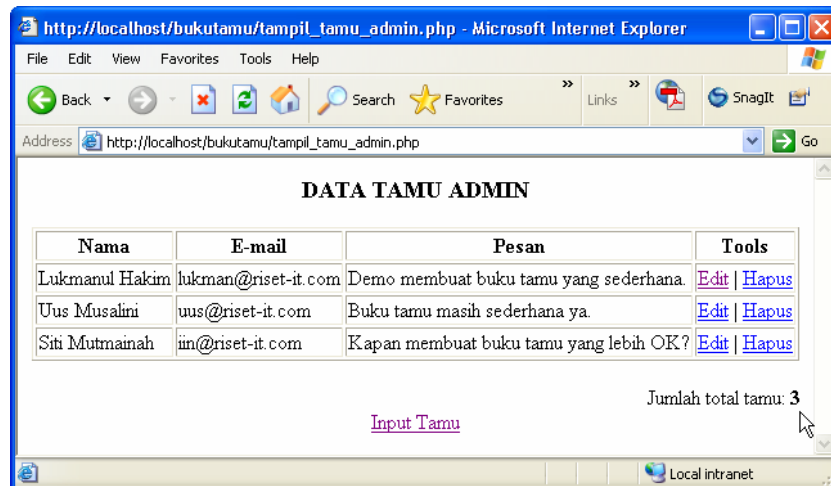
if ($hasil)
{
    header("location:tampil_tamu_admin.php");
}
else
{
    echo "Proses Penghapusan Data Gagal";
}
?>
```

Sekarang coba jalankan skrip tampil\_tamu\_admin.php, maka akan terlihat tampilannya seperti pada gambar, coba klik link Hapus untuk data paling atas (Adiguna Sutowo). Lihat gambar.



*Gambar. Hasil skrip tampil\_tamu\_admin.php*

Setelah link Hapus diklik, maka skrip hapus\_tamu.php akan memproses data yang dihapus dan apabila proses penghapusan data berhasil, maka secara otomatis halaman akan dialihkan ke halaman Data Tamu Admin dan hasil penghapusan data tersebut langsung dapat dilihat hasilnya seperti pada gambar.



*Gambar. Hasil penghapusan data*

### Trik. Proses Pencarian Data

Proses pengolahan data terakhir adalah proses pencarian data. Perintah SQL yang akan digunakan adalah:

```
SELECT * FROM nama_tabel WHERE $field_yg_dicari LIKE '%$id%';
```

Untuk mempraktekkannya, cobalah tuliskan dua buah file skrip berikut:

#### Skrip form\_cari.php

```
<HTML>
<BODY>
<H3>PENCARIAN DATA TAMU</H3>
<FORM METHOD=POST ACTION=cari_tamu.php>
Masukkan Nama Tamu: <INPUT TYPE=TEXT NAME="nama">
<INPUT TYPE=SUBMIT VALUE=Cari>
</FORM>
</BODY>
</HTML>
```

#### Skrip cari\_tamu.php

```
<?
echo "<H3 ALIGN=CENTER>HASIL PENCARIAN</H3>";

include "koneksi_bukutamu.php";

$cari="SELECT * FROM tamu WHERE nama LIKE '%$nama%'";
```

```

$hasil=mysql_query($cari);

//Hitung jumlah data yang ditemukan
$jumlah=mysql_num_rows($hasil);

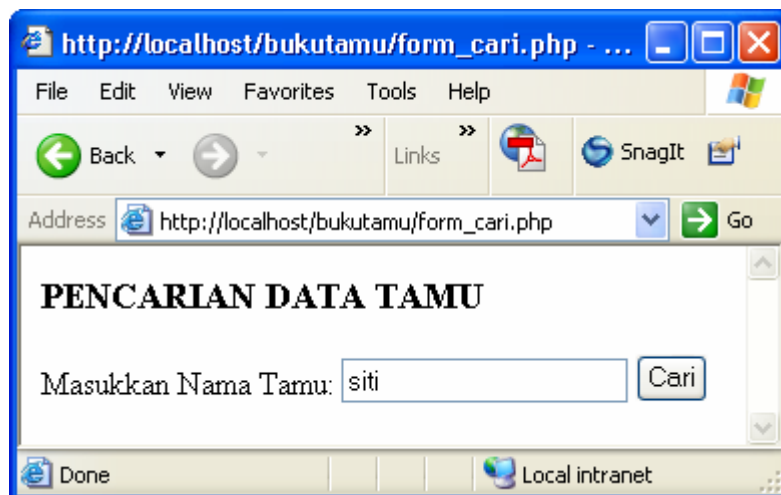
//Apabila jumlah data yang ditemukan lebih dari nol, berarti
//data ditemukan.
if ($jumlah > 0)
{
    echo "Tamu yang bernama <B>$nama</B> ada: $jumlah
    orang<BR><BR>";

    echo "<TABLE BORDER=1>
    <TR>
        <TH>Nama</TH>
        <TH>E-mail</TH>
        <TH>Pesan</TH>
    </TR>";

    while ($data=mysql_fetch_array($hasil))
    {
        echo "<TR><TD>$data[nama] </TD>";
        echo "<TD>$data[email] </TD>";
        echo "<TD>$data[pesan] </TD></TR>";
    }
    echo "</TABLE>";
}
else
{
    echo "Tamu dengan nama <B>$nama</B> tidak ditemukan";
}
?>

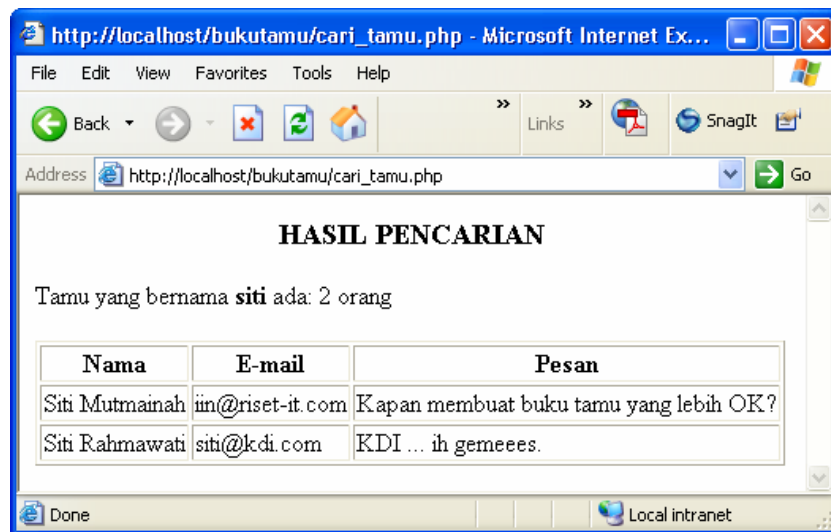
```

Sekarang coba jalankan skrip form\_cari.php, lalu isikan nama tamu yang akan dicari, misalnya siti. Lihat gambar.



*Gambar. Hasil skrip tampil\_tamu\_admin.php*

Kemudian klik tombol **Cari**, maka skrip cari\_tamu.php akan memproses data yang dicari dan apabila data yang dicari ada, maka datanya akan ditampilkan seperti pada gambar.



*Gambar. Hasil skrip cari\_tamu.php*