LAPORAN PRAKTIKUM
PEMROGRAMAN WEB & MOBILE I

NAMA : PUTRI SAPTIA PUSPITA
NIM : 11191060
KELAS : C
MODUL II

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2021

BAB I

TUJUAN DAN LANDASAN TEORI

1. Tujuan Praktikum

1.1. Mahasiswa mampu membuat handling yang mampu mengolah data dari form HTML.

1.2. Mahasiswa mampu membuat batasan-batasan untuk menangani inputan dari form HTML.

2. Landasan Teori

Variabel superglobal PHP $_GET dan $_POST digunakan untuk mengumpulkan data-form. Contoh berikut menunjukkan form HTML sederhana dengan dua field input dan tombol submit:

```
<html>
    <body>
        <form action="welcome.php" method="post">
            Name: <input type="text" name="name"><br>
            E-mail: <input type="text" name="email"><br>
            <input type="submit">
        </form>

    </body>
</html>
```

Gambar 1.1 PHP

Ketika user mengisi form, dan menekan tombol click, data form dikirim untuk memproses file PHP dengan nama “welcome.php”. Data form dikirimkan dengan method HTTP POST. Untuk menampilkan data yang sudah disubmit bisa dilakukan dengan mencetak data tersebut menggunakan perintah echo. File “welcome.php” adalah sebagai berikut:

```
<html>
    <body>
        Welcome <?php echo $_POST["name"]; ?><br>
        Your email address is: <?php echo $_POST["email"];
    ?> </body>
</html>
```

Gambar 1.2 PHP

Jika field nama diinputkan dengan Tono dan email diinputkan dengan tono@mail.com maka output yang akan tampil adalah sebagai berikut:
Welcome Budi
Your email address is tono@mail.com

Hasil yang sama juga akan tampil dengan menggunakan method get sebagai berikut: Untuk itu, gunakan keyword static dalam pendeklarasian variabel yang nilainya ingin dipertahankan.

```
<html>
    <body>

        <form action="welcome_get.php" method="get">
            Name: <input type="text" name="name"><br>
            E-mail: <input type="text" name="email"><br>
            <input type="submit">
        </form>
    </body>
</html>
```

Gambar 1.3 PHP

dengan file "welcome_get.php" sebagai berikut:

```
<html>
    <body>
        Welcome <?php echo $_GET["name"]; ?><br>
        Your email address is: <?php echo $_GET["email"];
    ?> </body>
</html>
```

Gambar 1.4 PHP

GET vs. POST

GET dan POST membuat sebuah array (contoh array(kunci => nilai, kunci2 => nilai2, kunci3 => nilai3, …)). Array ini menyimpan pasangan kunci/nilai, dimana kunci-kunci adalah nama-nama dari form control dan nilai-nilai adalah data input dari user. Method GET diakses menggunakan $_GET dan method POST diakses menggunakan $_POST. Kedua variabel ini adalah variabel superglobal, yang selalu bisa diakses, tanpa memperhatikan lingkup dan bisa diakses dari fungsi, class atau file yang berbeda tanpa harus melakukan teknik khusus. $_GET adalah sebuah array dari variabel yang dikirimkan ke skrip melalui parameter URL. $_POST adalah sebuah array dari variabel yang dikirimkan ke skrip melalui method HTTP POST.

Kapan sebaiknya menggunakan GET?

Informasi dikirim dari sebuah form dengan method GET bisa dilihat oleh semua orang (semua nama dan nilai variabel ditampilkan di URL). GET juga memiliki batas pada jumlah informasi yang dikirim. Batasannya adalah sekitar 2000 karakter. Namun, karena variabel ditunjukkan di URL, ia memungkinkan untuk dilakukan bookmark halaman. Dalam beberapa kasus, hal ini sangat bermanfaat. GET bisa digunakan untuk mengirimkan data yang tidak sensitif.

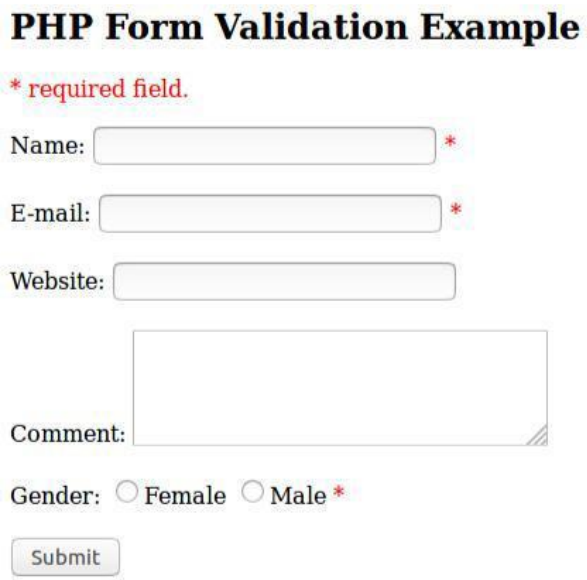
Ingat! GET tidak boleh digunakan untuk mengirimkan password atau informasi sensitif lainnya!

Kapan menggunakan POST?

Informasi yang dikirim dari sebuah form dengan method POST tidak bisa dilihat oleh siapapun (semua nama-nama atau nilai-nilai tertanam didalam body request HTTP) dan tidak memiliki batasan jumlah informasi yang akan dikirim. POST juga mendukung fungsionalitas lanjutan seperti dukungan untuk input biner multi-part ketika sedang melakukan upload file ke server. Namun, karena variabel tidak ditampilkan di URL, tidak mungkin untuk dilakukan bookmark halaman (data tidak ter-bookmark). Developer lebih baik menggunakan POST untuk mengirimkan data form.

Validasi Form PHP

Pertimbangkan keamanan ketika memproses form PHP!



Gambar 1.5 PHP Form

Form HTML yang akan kita gunakan pada modul ini, mengandung bermacam-macam field input, misalnya text field yang harus diisi dan text field yang opsional, tombol pilihan (radio button), dan tombol submit. Rule atau aturan validasi untuk form diatas adalah sebagai berikut:

Field	Rules Validasi
Name	Dibutuhkan. + Harus hanya mengandung huruf dan spasi
Email	Dibutuhkan. + Harus mengandung sebuah alamat email yang valid dengan @ dan .
Website	Opsional. Jika ada, harus mengandung URL yang valid.
Comment	Opsional. Field input multi-line (text area).
Gender	Dibutuhkan. Harus memilih salah satu

Kode HTML untuk membentuk Form tersebut adalah sebagai berikut:

Text Field

Field nama, email dan website adalah elemen-elemen text input, dan field komentar adalah textarea yaitu sebagai berikut:

```
Name: <input type="text" name="name">
E-mail: <input type="text" name="email">
Website: <input type="text" name="website">
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
```

Radio Button

Field jenis kelamin adalah radio button yaitu sebagai berikut:

```
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
```

Form Element

Kode HTML untuk membentuk form pada gambar diatas adalah sebagai berikut:

```
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);? >">
```

Ketika form disubmit, data pada form dikirim dengan method "post". **$_SERVER["PHP_SELF"]** adalah variabel super global yang mengembalikan nama file dari skrip yang sedang dieksekusi. Sehingga kode form diatas mengirim data pada form ke halaman itu sendiri. Sedangkan fungsi **htmlspecialchars()** adalah fungsi yang mengkonversikan karakter-karakter spesial ke entitas HTML. Sebagai contoh, fungsi tersebut akan mengkonversikan karakter < dan > menjadi < dan >. Fungsi ini mencegah injeksi yang bisa dilakukan dengan HTML atau javascript (Cross-site Scripting Attack) pada form tersebut.

Catatan Penting pada Keamanan Form PHP

Variabel $_SERVER["PHP_SELF"] bisa digunakan oleh hacker! Jika PHP_SELF digunakan pada halaman web, user bisa memasukkan skrip dengan terlebih dahulu memasukkan garis miring (/) kemudian beberapa perintah Cross Site Scripting (XSS) untuk dieksekusi. XSS adalah tipe kelemahan keamanan komputer yang secara tipikal ditemukan dalam aplikasi web.

Asumsikan kita memiliki halaman web dengan nama "test_form.php", dan form hanya kita deklarasikan sebagai berikut:

```
<form method="post" action="<?php echo $_SERVER["PHP_SELF"];?>">
```

Kemudian user memasukkan URL pada address bar dengan alamat sebagai berikut:

http://localhost/<nama_folder>/test_form.php/%22%3E%3Cscript%3Ealert('hacked')%3C/script %3E

yang jika ditranslasikan akan menjadi:

<form method="post" action="test_form.php/"><script>alert('hacked')</script>

Kode ini menambah tag script dan perintah alert atau peringatan, ketika halaman dibuka, kode javascript tersebut akan dieksekusi, maka user akan melihat kotak peringatan dengan tulisan “hacked”.

Bagaimana menghindari penyalahgunaan $_SERVER[“PHP_SELF”]?

Caranya adalah dengan menggunakan fungsi htmlspecialchars(). Fungsi tersebut akan mengkonversikan karakter khusus ke entitas HTML. Ketika user memasukkan URL dengan tag script seperti contoh sebelumnya, maka akan ditranslasikan sebagai berikut:

<form method="post"
action="test_form.php/"><script>alert('hacked')</script>">

dengan cara ini, percobaan penyalahgunaan akan gagal.

Memvalidasi data Form dengan PHP

Hal pertama yang akan kita lakukan adalah memasukkan semua variabel melalui fungsi htmlspecialchars(). Kemudian ada juga dua hal ketika user melakukan submit form:

1. Membuang karakter-karakter yang tidak dibutuhkan (seperti spasi extra, tab extra, dan baris baru yang ektra) dari data input user (dengan fungsi trim()).

2. Membuang backslash (\) tatu garis miring dari data input user (dengan fungsi stripslashed()).

Langkah berikutnya adalah membuat fungsi yang akan melakukan pemeriksaan kebenaran data yang diinputkan oleh user. Contohnya adalah sebagai berikut:

```php
<?php
	// define variables and set to empty values
	$name = $email = $gender = $comment = $website = "";

	if ($_SERVER["REQUEST_METHOD"] == "POST") {
		$name = test_input($_POST["name"]);
		$email = test_input($_POST["email"]);
		$website = test_input($_POST["website"]);
		$comment = test_input($_POST["comment"]);
		$gender = test_input($_POST["gender"]);
	}

	function test_input($data) {
		$data = trim($data);
		$data = stripslashes($data);
		$data = htmlspecialchars($data);
		return $data;

	}
?>
```

Gambar 1.6 PHP

Ingat bahwa pada permulaan skrip, adalah pemeriksaan apakah form sudah disubmit menggunakan $_SERVER[“REQUEST_METHOD”]. Jika REQUEST_METHOD adalah POST, maka form telah disubmit dan seharusnya tervalidasi. Jika belum tersubmit, lewati langkah validasi dan tampilkan form kosong. Namun pada contoh diatas semua field input adalah opsional. Skrip bekerja baik bahkan jika user tidak melakukan entri data.

Field yang Dibutuhkan

Kode program berikut terdapat tambahan variabel baru yaitu: $nameErr, $emailErr, $genderErr. Variabel-variabel error ini akan menangani pesan error untuk field yang dibutuhkan. Percabangan dengan if else juga akan ditambahkan untuk setiap variabel $_POST. Fungsinya untuk memeriksa apakah variabel $_POST kosong, hal ini dilakukan dengan menggunakan fungsi empty(). Jika kosong, maka pesan error disimpan dalam variabel error yang berbeda, dan jika tidak kosong, ia akan mengirim data input user melalui fungsi test_input():

```php
<?php
	// define variables and set to empty values
	$nameErr = $emailErr = $genderErr = $websiteErr = "";
	$name = $email = $gender = $comment = $website = "";

	if ($_SERVER["REQUEST_METHOD"] == "POST")
		{ if (empty($_POST["name"])) {
			$nameErr = "Name is required";
		} else {
			$name = test_input($_POST["name"]);
		}

		if (empty($_POST["email"])) {
			$emailErr = "Email is required";
		} else {
			$email = test_input($_POST["email"]);
		}

		if (empty($_POST["website"])) {
			$website = "";
```

```
        } else {
            $website = test_input($_POST["website"]);
        }

        if (empty($_POST["comment"])) {
            $comment = "";
        } else {
            $comment = test_input($_POST["comment"]);
        }

        if (empty($_POST["gender"])) {
            $genderErr = "Gender is required";
        } else {
            $gender = test_input($_POST["gender"]);
        }
    }
?>
```

Gambar 1.7 PHP

Setelah kode diatas ditambahkan, beberapa skrip ditambahkan pada setiap field yang dibutuhkan pada form, fungsinya untuk menampilkan pesan error jika field yang dibutuhkan tidak diisi. Form HTMLnya adalah sebagai berikut:

```
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">

    Name: <input type="text" name="name">
    <span class="error">* <?php echo
    $nameErr;?></span> <br><br>
    E-mail:
    <input type="text" name="email">
    <span class="error">* <?php echo $emailErr;?></span>
    <br><br>
    Website:
    <input type="text" name="website">
    <span class="error"><?php echo $websiteErr;?></span>
    <br><br>
    Comment: <textarea name="comment" rows="5" cols="40"></textarea>
    <br><br>
    Gender:
    <input type="radio" name="gender" value="female">Female
    <input type="radio" name="gender" value="male">Male
    <span class="error">* <?php echo $genderErr;?></span>
    <br><br>
    <input type="submit" name="submit" value="Submit">

</form>
```

Gambar 1.8 PHP

Validasi Nama

Kode berikut menunjukkan cara sederhana untuk memeriksa apakah field nama hanya mengandung huruf dan spasi. Jika nilai dari nama tidak valid, maka pesan error akan disimpan didalam variabel $nameErr:

```
$name = test_input($_POST["name"]);
if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
    $nameErr = "Only letters and white space allowed";
}
```

Fungsi preg_match() mencari string berdasarkan pola, mengembalikan nilai true jika polanya ada, false jika polanya tidak ada.

Validasi Email

Cara paling mudah dan paling aman untuk memeriksa apakah sebuah alamat email memiliki pola yang sesuai adalah dengan menggunakan fungsi filter_var(). Kode dibawah memeriksa apakah alamat email yang dimasukkan menggunakan pola yang sesuai atau tidak, jika tidak, maka pesan error akan disimpan kedalam variabel $emailErr:

```
$email = test_input($_POST["email"]);
if (!filter_var($email, FILTER_VALIDATE_EMAIL))
        { $emailErr = "Invalid email format";
}
```

Gambar 1.9 PHP

Validasi URL

Kode program berikut menunjukkan cara untuk memeriksa apakah sintaks alamat URL valid atau tidak. Ekspresi reguler ini mengizinkan keberadaan tanda pisah pada URL. Jika sintaks alamat URL tidak valid, maka pesan error akan disimpan kedalam variabel $websiteErr:

```
$website = test_input($_POST["website"]);
if (!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~_|!:,.;]*[-a-z0-
9+&@#\/ %=~_|]/i",$website)) {
$websiteErr = "Invalid URL";
                }
```

Biasanya, jika user salah menginputkan nilai, maka halaman yang tampil adalah halaman yang sama dengan field yang sudah terisi dengan nilai field yang sudah diinput sebelumnya. Untuk menunjukkan nilai dalam field input setelah user menekan tombol submit, ada beberapa skrip PHP yang perlu ditambahkan didalam atribut value pada field input name, email, dan website. Khusus untuk field textarea, akan skrip tersebut akan ditambahkan antara tag <textarea> dan tag </textarea>. Skrip yang singkat akan mengeluarkan nilai dari variabel $name, $email, $website dan $comment. Untuk radio button atau tombol radio, akan ditambahkan kode yang membuat salah satu pilihan terpilih.

```
Name: <input type="text" name="name" value="<?php echo $name;?>">

E-mail: <input type="text" name="email" value="<?php echo $email;?>">

Website: <input type="text" name="website" value="<?php echo $website;?>">

Comment: <textarea name="comment" rows="5" cols="40"><?php echo
$comment;? ></textarea>

Gender:
<input type="radio" name="gender"
<?php if (isset($gender) && $gender=="female") echo
"checked";?> value="female">Female
<input type="radio" name="gender"
<?php if (isset($gender) && $gender=="male") echo
"checked";?> value="male">Male
```

Gambar 1.10 PHP

BAB II

PEMBAHASAN

Buatlah program web untuk menginputkan username dan password menggunakan form dan penanganan input data dengan kriteria sebagai berikut:

1. Username yang dinputkan tidak boleh lebih dari tujuh karakter.
2. Password yang dinputkan harus terdiri dari huruf kapital, huruf kecil, angka dan karakter khusus.
3. Jumlah karakter password tidak boleh kurang dari sepuluh karakter.

Berikut adalah kode program untuk tugas diatas:

```php
<DOCTYPE html>
<html>
    <?php
        $username = $password = "";

        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            $username = test_input($_POST["username"]);
            $password = test_input($_POST["password"]);
        }

        function test_input($data) {
            $data = trim($data);
            $data = stripslashes($data);
            $data = htmlspecialchars($data);
            return $data;
        }
    ?>

    <?php
        $usernameErr = $passwordErr = "";
        $username = $password = "";
```

```php
    if ($_SERVER["REQUEST_METHOD"] == "POST")
    $username = test_input($_POST["username"]);
        if (strlen($username) >= 7) {
            $usernameErr = "Username tidak boleh lebih dari 7 karakter";
        }
        {if (empty($_POST["username"])) {
            $usernameErr = "Name is required";
        } else {
            $username = test_input($_POST["username"]);
        }

    $password = test_input($_POST["password"]);
        if (strlen($password) <=10 ) {
            $passwordErr = "Password minimal 10 digit";
        }
        if (!preg_match('@[a-zA-Z^\w]@', $password)){
            $passwordErr = "Password minimal memiliki satu Huruf Kapital, satu Huruf Kecil, satu angka dan satu karakter khusus";
        }
        if (empty($_POST["password"])) {
            $passwordErr = "password is required";
        }
        else {
            $password = test_input($_POST["password"]);
        }
    }
?>
```

```php
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">

    username: <input type="text" name="username">
    <span class="error">* <?php echo $usernameErr;?></span> <br><br>
    password:
    <input type="text" name="password">
    <span class="error">* <?php echo $passwordErr;?></span> <br><br>
    <input type="submit" name="submit" value="Submit">
</form>
</html>
```

Pada tugas nomor 1, kita diminta untuk menerima input Username berupa huruf yang tidak boleh lebih dari 7 karakter.

```
  if ($_SERVER["REQUEST_METHOD"] == "POST")
        $username = test_input($_POST["username"]);
            if (strlen($username) >= 7) {
                $usernameErr = "Username tidak boleh lebih dari 7 karakter";
            }
            {if (empty($_POST["username"])) {
                $usernameErr = "Name is required";
            } else {
                $username = test_input($_POST["username"]);
            }
```

Saya menggunakan fungsi if dalam dengan method post, yaitu jika panjang input yang dimasukkan lebih dari 7, maka akan muncul pesan “Username tidak boleh lebih dari 7 karakter”. Kemudian jika inputan kosong, maka akan muncul pesan “Name is Required”.

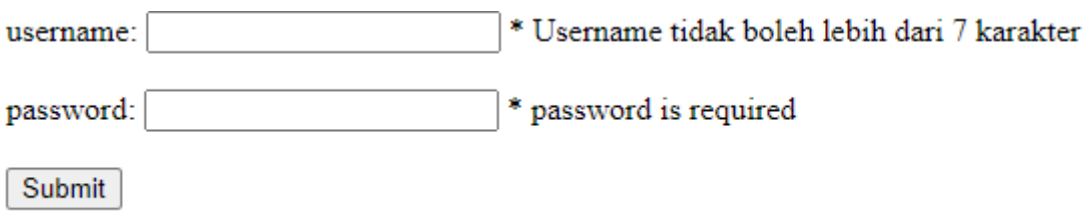
Pada tugas nomor 2 , kita diminta untuk menerima input Password berupa huruf kapital, huruf kecil, angka dan karakter khusus dan untuk tugas nomor 3 kita diminta untuk menerima input password minimal sepanjang 10 karakter.

```
      $password = test_input($_POST["password"]);
            if (strlen($password) <=10 ) {
                $passwordErr = "Password minimal 10 digit";
            }
            if (!preg_match('@[a-zA-Z1-9^\w]@', $password)){
                $passwordErr = "Password minimal memiliki satu Huruf Kapital, sa
tu Huruf Kecil, satu angka dan satu karakter khusus";
            }
            if (empty($_POST["password"])) {
                $passwordErr = "password is required";
            }
            else {
                $password = test_input($_POST["password"]);
            }
        }
```

Saya menggunakan fungsi if yang mirip seperti pada nomor 1 dengan method post, yaitu jika panjang input yang dimasukkan kurang dari 10 karakter, maka akan muncul pesan “Password minimal 10 digit”. Kemudian untuk memvalidasi password adalah dengan memasukan preg_match yang memiliki parameter ($regex, $string), di mana $regex adalah pola yang akan dicari dan $string adalah variabel yang akan dicari apakah ada pola $regex di dalamnya. Dan disini saya memasukan `[a-zA-Z1-9^\w]` sebagai pola yang akan dicari dalam password, jika tidak ada salah satu pola tersebut dalam password, maka akan muncul pesan “Password minimal memiliki satu huruf kapital, satu huruf kecil, satu angka dan satu karakter khusus”. Jika inputan password kosong, maka akan muncul pesan “password is required”. Berikut adalah hasil dari kode program diatas.

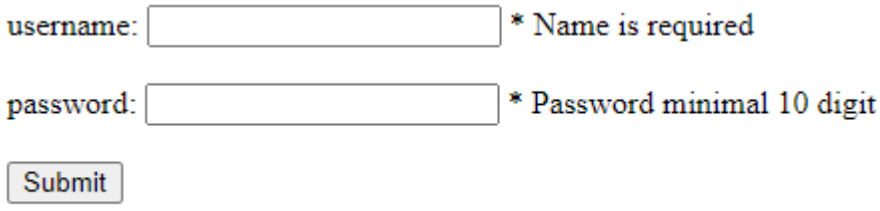
Gambar 1.1 Hasil Output nomor 1

Ini adalah hasil jika username yang dimasukan lebih dari 7 karakter.



Gambar 1.2 Hasil Output nomor 1

Ini adalah hasil jika password yang dimasukan kurang dari 10 karakter.



Gambar 1.3 Hasil Output nomor 1

KESIMPULAN

Form adalah sebuah fasilitas yang sering kita jumpai pada sebuah website. Fungsi form adalah sebagai media untuk mengisi data dan akan dikirim ke webserver, sehingga sebuah website akan tampak lebih interaktif. Komponen-komponen form terdiri dari : Atribut Teks, Atribut Radio, Atribut Checkbox, Atribut Select & Option, Atribut Teks Area, dan Atribut Submit.

Form HTML biasanya digunakan untuk menangani input dari user pada dokumen yang dinamis/interaktif. Analogi dari penggunaan form ini di dokumen html adalah bahwa form tersebut adalah sebuah formulir isian yang akan diisi oleh user, kemudian disimpan ke dalam server dengan mekanisme penyimpanan data.

DAFTAR PUSTAKA

Praktikum, K. (n.d.). *MODUL PRAKTIKUM PEMROGRAMAN WEB I Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya*.

LAMPIRAN

Praktikum PewMeb 1 Modul 1

Nama : Putri Saptia Puspita

NIM : 11191060

Nomor 1 : Nama Anggota Keluarga

Suyono
Nurhayati
Ade Kurniawan Wibowo
Putri Saptia Puspita
Rakha Syafiq Febrian

Nomor 2 : Jumlah Huruf

6
9
20
20
20

Nomor 2 : Jumlah Kata

1
1
3
3
3

Nomor 3 : Kebalikan Nama

onoyuS
itayahruN
owobiW nawainruK edA
atipsuP aitpaS irtuP
nairbeF qifayS ahkaR

Nomor 3 : Kebalikan Nama

onoyuS
itayahruN
owobiW nawainruK edA
atipsuP aitpaS irtuP
nairbeF qifayS ahkaR

Nomor 4 : Jumlah Huruf Vokal

3
4
8
8
7

Nomor 4 : Jumlah Huruf Konsonan

3
5
12
12
13