
Bab 12

ARRAY

Selama ini bentuk variabel yang kita buat untuk menyimpan suatu value atau nilai adalah berbentuk tunggal. Maksudnya, satu variabel hanya bisa untuk menyimpan sebuah value saja. Misalkan sebuah value telah disimpan atau diassign ke dalam sebuah variabel, selanjutnya variabel tersebut akan diassign kembali dengan sebuah value yang lain, maka value yang sebelumnya akan ditimpa oleh value yang baru.

Nah... kira-kira ada tidak ya, variabel yang dapat diassign oleh beberapa value yang lebih dari satu? Oya.. ada, namanya adalah variabel bertipe data array.

Misalkan suatu variabel bernama X bertipe data array, maka X ini dapat Anda bayangkan seperti gambar berikut

X

0	1	2	3	4	...	N-1

Wah... kok 'X' nya seperti memiliki beberapa ruang/kamar/bilik atau apalah terserah namanya ☺ yang jelas jumlah ruangnya adalah berhingga yaitu sejumlah N. Tiap-tiap kamar terdapat semacam nomor ruangan: 0, 1, 2, 3, ..., N-1. Nah.. di dalam ruangan-ruangan inilah kita bisa assign beberapa value sekaligus.

Ingat ya... untuk PHP, nomor ruang array dimulai dari 0, bukan 1.

Dalam PHP, untuk mendeklarasikan suatu variabel (misalkan \$x) bertipe data array adalah sebagai berikut:

```
$x = array();
```

Selanjutnya bagaimana perintah untuk mengassign suatu value ke dalam ruangan array?

Oya... sebelumnya, saya perlu sampaikan bahwa untuk PHP, variabel bertipe data array ini bisa diassign dengan berbagai tipe data. Maksudnya, untuk setiap ruang yang ada, boleh diassign dengan nilai berbagai tipe data yang berlainan, misalnya untuk ruang nomor 1 akan diisi dengan nilai berupa bilangan bulat (integer), ruang no. 2 diisi dengan real, no. 3 diisi dengan string dsb. Itu bisa dilakukan.

Nah.. berikut ini contoh mengassign ruang variabel yang bertipe data array. Sebagai contoh misalkan kita akan mengassign ruang nomor 3 dengan value bilangan integer 10, dan ruang no. 5 dengan string 'I love PHP' perintahnya:

```
<?php

$x = array();
$x[3] = 10;
$x[5] = "I love PHP";

?>
```

Kita juga bisa mengisi value untuk setiap ruang array sekaligus secara berturut-turut seperti contoh berikut ini:

```
<?php

$x = array(10, 12, 3, 44, 50, "hallo");

?>
```

atau cara kedua bisa pula menggunakan seperti berikut ini:

```
<?php

$x = array(0 => 10, 1 => 12, 2 => 3, 3 => 44, 4 => 50, 5 => "hallo");

?>
```

Selanjutnya, bagaimana cara kita menampilkan value-value yang tersimpan dalam kamar array tersebut? Ya.. caranya mudah, tinggal panggil saja nomor kamarnya lantas di echo kan.

```
<?php

$x = array(10, 12, 3, 44, 50, "hallo");
echo $x[0]; // akan menampilkan value pada ruang ke-0, yaitu 10
echo $x[5]; // akan menampilkan value pada ruang ke-5, yaitu 'hallo'

$x[2] = -3; // mengganti value pada ruang ke-2 dengan value yang baru (-3)
echo $x[2]; // akan menghasilkan -3 (yang tampil bukan 3 karena sudah ditimpa -3)

?>
```

Kita juga bisa menampilkan semua value array sekaligus secara berturut-turut menggunakan looping

```
<?php
```

```
$x = array(10, 12, 3, 44, 50, "hallo");
```

```
for ($i = 0; $i <= 5; $i++)
```

```
{  
    echo "Nilai kamar ke ".$i. " dari array = ".$x[$i]. "<br />";  
}
```

```
?>
```

atau cara kedua, kita bisa menggunakan perintah `count()` untuk menentukan jumlah ruang array yang telah diisi value.

```
<?php
```

```
$x = array(10, 12, 3, 44, 50, "hallo");
```

```
for ($i = 0; $i <= count($x)-1; $i++)
```

```
{  
    echo "Nilai kamar ke ".$i. " dari array = ".$x[$i]. "<br />";  
}
```

```
?>
```

Dalam script di atas, perintah `count($x)` akan menghasilkan 6 karena dalam array `$x` tersebut terdapat 6 ruang yang terisi value. Penggunaan `count()` ini akan efektif bila kita malas menghitung jumlah ruang array yang telah diisi value secara manual, ibaratnya kita minta tolong si PHP untuk menghitungkannya. Bayangkan seandainya terdapat 100 lebih ruang dalam suatu array, masa' kita akan menghitungnya satu-persatu secara manual? Capee deh... ☺

Array Asosiatif

Dari contoh-contoh di atas, kita lihat bahwa ruang atau kamar dari array diberikan nomor ruang sebagai penanda, yaitu mulai dari 0, 1, 2, dst... Nah... yang menjadi pertanyaan, apakah penanda setiap ruang ini harus berupa angka? Jawabnya tidak, Anda juga bisa memberi penanda setiap ruang dengan suatu kata atau karakter. Array yang demikian ini disebut array asosiatif.

Berikut ini contoh membuat array asosiatif dan cara menampilkan value pada suatu ruangnya

```
<?php
```

```
$x = array("kamar1" => 10, "kamar2" => 12, "kamar3" => 3, "kamar4" => "hallo");  
echo $x["kamar2"]; // akan menampilkan 12  
echo $x["kamar4"]; // akan menampilkan "hallo"
```

```
?>
```

Sekarang bagaimana cara menampilkan semua value yang tersimpan dalam array asosiatif tersebut sekaligus? Sama caranya dengan sebelumnya, namun kita khusus akan menggunakan looping foreach. Looping ini khusus untuk array asosiatif.

```
<?php
```

```
$x = array("kamar1" => 10, "kamar2" => 12, "kamar3" => 3, "kamar4" => "hallo");
```

```
foreach($x as $ruang => $value)  
{  
    echo "Isi dari ruang : ".$ruang. " adalah : ".$value. "<br />";  
}
```

```
?>
```

Menambah Elemen Array

Misalkan kita sebelumnya telah memiliki array dengan value-value tertentu dalam setiap ruangnya, perhatikan script berikut ini

```
<?php
```

```
$x = array(10, 12, 3, 44, 50, "hallo");
```

```
?>
```

Nah... bagaimana seandainya jika kita ingin menambahkan sebuah ruang lagi untuk diisi dengan suatu value? Dalam hal ini pada array di atas nomor ruangnya terdiri dari 0, 1, 2, ..., 5. Atau dengan kata lain kita akan menambah sebuah ruang lain yaitu ruang nomor 6 untuk diisi dengan suatu value, misalkan 100. OK.. caranya adalah sebagai berikut

Cara pertama, Anda bisa langsung memberikan perintah

```
$x[6] = 100;
```

atau

```
<?php
```

```
$x = array(10, 12, 3, 44, 50, "hallo");
```

```
$x[6] = 100;
```

```
?>
```

Atau dengan cara kedua menggunakan perintah

```
$x[] = 6;
```

Atau

```
<?php
```

```
$x = array(10, 12, 3, 44, 50, "hallo");
```

```
$x[] = 100;
```

```
?>
```

Sekarang kita lihat apakah ruangnya benar-benar bertambah.

```
<?php
```

```
$x = array(10, 12, 3, 44, 50, "hallo");
```

```
$x[] = 100;
```

```
for ($i = 0; $i <= count($x)-1; $i++)
```

```
{
```

```
    echo "Nilai kamar ke ".$i. " dari array = ".$x[$i]. "<br />";
```

```
}
```

```
?>
```

Yes... benar-benar bertambah ruangnya ☺

Contoh Kasus

Sekarang kita akan mencoba membahas beberapa contoh kasus yang menggunakan array.

Kasus 1. Menghitung Rata-rata beberapa bilangan

Misalkan diberikan bilangan-bilangan berikut ini 9, 2, 10, 3, 7, 5, 2. Kemudian kita diminta membuat script untuk mencari rata-rata dari semua bilangan tersebut. Nah..

dalam hal ini, kita bisa menggunakan array untuk menghitungnya. Anda bisa saja sih menggunakan script berikut ini

```
<?php

$rataRata = (9 + 2 + 10 + 3 + 7 + 5 + 2)/7;
echo $rataRata;

?>
```

Namun... bagaimana bila jumlah data bilangannya sampai dengan 100 atau bahkan ribuan? He... 3x Anda bisa bayangkan sendiri alangkah repotnya dalam menulis formulanya. Nah.. untuk memudahkan perhitungan, kita akan menggunakan pendekatan array.

Langkah pertama: kita buat suatu array yang ruangnya diisi oleh bilangan-bilangan tersebut.

```
<?php

$bil = array(9, 2, 10, 3, 7, 5, 2);

?>
```

Selanjutnya.. langkah kedua: kita hitung dulu hasil penjumlahan setiap value dalam array tersebut. Untuk menghitungnya, kita manfaatkan looping. Konsepnya adalah seperti ini: mula-mula kita asumsikan hasil jumlahnya adalah 0 (karena kita belum mulai menjumlahkan), kemudian kita baca array ke-0, yaitu 9. Value ini kita jumlahkan dengan hasil penjumlahan sebelumnya yaitu 0 atau $0 + 9 = 9$. Dengan demikian hasil penjumlahan sekarang adalah 9. Selanjutnya kita baca array ke-1, yaitu 2. Value ini kembali kita jumlahkan dengan hasil penjumlahan sebelumnya yaitu 9 atau $9 + 2 = 11$. So.. hasil penjumlahan sekarang menjadi 11. Proses ini terus diulang sampai pada value terakhir dari array. Setelah selesai, maka akan kita peroleh hasil dari penjumlahan total semua value.

```
<?php

$bil = array(9, 2, 10, 3, 7, 5, 2);

$jumlah = 0;
for ($i = 0; $i <= count($bil)-1; $i++)
{
    $jumlah = $jumlah + $bil[$i]; // atau bisa dituliskan $jumlah += $bil[$i];
}
?>
```

Setelah kita peroleh hasil penjumlahan setiap value dari array, langkah ketiga kita bisa hitung deh rata-rata bilangannya.

```
<?php
```

```
$bil = array(9, 2, 10, 3, 7, 5, 2);
```

```
$jumlah = 0;
```

```
for ($i = 0; $i <= count($bil)-1; $i++)
```

```
{
```

```
    $jumlah = $jumlah + $bil[$i]; // atau bisa dituliskan $jumlah += $bil[$i];
```

```
}
```

```
$rataRata = $jumlah/count($bil);
```

```
echo "Rata-ratanya adalah : ".$rataRata;
```

```
?>
```

Kasus 2. Mencari Nilai Terbesar dari Kumpulan Bilangan

Misalkan diberikan bilangan-bilangan berikut ini 9, 2, 10, 3, 7, 5, 2. Kita diminta membuat script untuk menampilkan bilangan terbesar dari bilangan-bilangan tersebut. Tentu kita tahu, dalam hal ini bilangan terbesarnya adalah 10. Nah... yang jadi masalah, bagaimana cara mendapatkan 10 tersebut sebagai bilangan terbesar bila dinyatakan dalam program.

Secara sekilas, kasusnya tampak sederhana namun ketika dinyatakan dalam program bisa jadi membuat pusing orang ☺

Untuk mencari nilai terbesar, kita gunakan konsep perbandingan.

Anda yang sudah berumur 17 tahun ke atas, mungkin tidak merasakan logika atau konsep ini karena Anda sudah banyak makan garam dalam masalah perhitungan sederhana seperti ini. Secara tidak sadar, ketika Anda diminta mencari bilangan terbesar, sebenarnya dalam pikiran Anda terjadi proses perbandingan tersebut beberapa kali. Namun, karena proses yang terjadi begitu cepat (seiring bertambahnya pengalaman dalam berpikir), maka proses ini seakan tidak terasa. Mungkin hal inilah yang kadang membuat para pembelajar program sering sulit mengekspresikan proses yang terjadi dalam pikirannya ke dalam bentuk program.

Tapi... sebenarnya tidak demikian, membuat program pada dasarnya mirip ketika Anda sedang mengajarkan konsep atau ide penyelesaian suatu kasus pada anak kecil yang sebelumnya belum tahu apa-apa. Kita harus bisa menyatakan setiap proses langkah demi langkah secara urut.

Nah... stop ber-intermezzo... Now.. back to the case.. ☺

Trus... idenya bagaimana untuk mencari bilangan terbesar?

OK.. idenya adalah, kita harus membuat asumsi terlebih dahulu. Kita asumsikan bilangan terbesar mula-mula adalah bilangan pertamanya. Selanjutnya bilangan terbesar asumsi ini kita bandingkan dengan setiap bilangan yang ada secara urut satu persatu. Jika terdapat bilangan yang lebih besar dari bilangan terbesar asumsi, maka bilangan tersebut akan dianggap sebagai bilangan terbesar asumsi berikutnya. Dengan demikian setelah proses perbandingan selesai semuanya, maka akan kita peroleh bilangan terbesar dari semua bilangan.

Contoh: dari kasus di atas, pertama kita asumsikan bilangan terbesarnya adalah bilangan pertama yaitu 9. Selanjutnya bilangan terbesar ini kita bandingkan dengan bilangan ke-1, yaitu 9. Karena 9 tidak lebih besar dari 9, maka bilangan terbesarnya tetap 9. Selanjutnya dibandingkan dengan bilangan ke-2, yaitu 2. Dalam hal ini 2 tidak lebih besar dari 9, maka bilangan terbesarnya tetap 9. Trus.. kita bandingkan dengan bilangan ke-3 yaitu 10. Nah... 10 lebih besar dari 9, maka bilangan terbesarnya sekarang menjadi 10. Proses ini dilanjutkan sampai dengan bilangan terakhir.

```
<?php
```

```
$bil = array(9, 2, 10, 3, 7, 5, 2);
```

```
$max = $bil[0];
```

```
for ($i = 0; $i <= count($bil)-1; $i++)
```

```
{
```

```
    if ($bil[$i] > $max) $max = $bil[$i];
```

```
}
```

```
echo "Bilangan terbesarnya adalah : ".$max;
```

```
?>
```

Kasus 3. Identik dengan kasus 1, namun bilangan-bilangan yang akan dicari rata-ratanya diinputkan melalui form.

Adapun skenario untuk input bilangannya adalah: pertama kali akan muncul form untuk memasukkan jumlah bilangan yang akan dimasukkan. Setelah dimasukkan jumlah bilangannya, kemudian disubmit berikutnya akan muncul form baru yang berisi komponen textbox untuk memasukkan masing-masing bilangan. Jumlah komponen textbox ini sejumlah bilangan yang sebelumnya diinputkan. Misalkan pada form pertama tadi kita masukkan jumlah bilangannya 10, maka setelah disubmit akan muncul form baru berisi 10 buah textbox untuk memasukkan ke-10 bilangan. Setelah dimasukkan masing-masing bilangan, selanjutnya disubmit dan akhirnya akan muncul jumlah rata-ratanya.

Untuk itu pertama kita buat form untuk memasukkan jumlah bilangannya terlebih dahulu.

form1.htm

```
<h1>Cari Rata-rata Bilangan</h1>
```

```
<form method="post" action="form2.php">
Masukkan Banyaknya Bilangan <input type="text" name="n" />
<input type="submit" name="submit" value="Submit" />
</form>
```

Selanjutnya kita buat form kedua untuk menampilkan form dinamis guna memasukkan masing-masing bilangan. Karena sesuai skenario bahwa jumlah textboxnya sejumlah banyaknya bilangan yang dimasukkan pada form pertama, maka untuk mudahnya kita buat melalui perulangan FOR dengan jumlah perulangannya n kali.

form2.php

```
<h1>Cari Rata-rata Bilangan</h1>
```

```
<?php
$n = $_POST['n'];

echo "<form method='post' action='proses.php'>";
echo "<table>";
for ($i = 0; $i <= $n-1; $i++)
{
    echo "<tr><td>Bil ke-".($i+1)."</td>
        <td><input type='text' name='data'.'. $i.' /></td></tr>";
}
echo "<tr><td></td>
    <td><input type='submit' name='submit' value='Submit' /></td></tr>";
echo "</table>";
echo "</form>";

?>
```

Hasil script di atas akan menghasilkan tag HTML seperti di bawah ini (jika n = 4)

```
<h1>Cari Rata-rata Bilangan</h1>

<form method='post' action='proses.php'>
<table>
<tr>
    <td>Bil ke-1</td>
    <td><input type='text' name='data0' /></td>
</tr>
<tr>
```

```

        <td>Bil ke-2</td>
        <td><input type='text' name='data1' /></td>
    </tr>
    <tr>
        <td>Bil ke-3</td>
        <td><input type='text' name='data2' /></td>
    </tr>
    <tr>
        <td>Bil ke-4</td>
        <td><input type='text' name='data3' /></td>
    </tr>
    <tr>
        <td></td>
        <td><input type='submit' name='submit' value='Submit' /></td>
    </tr>
</table>
</form>

```

Perhatikan tag HTML di atas! Untuk setiap komponen textbox yang dihasilkan tampak pada bagian atribut name="" secara otomatis akan membentuk nama: 'data0', 'data1', dst. Mengapa dimulai dari 'data0'? kok tidak mulai dari 'data1'? ya... ini masalah kesukaan saja. Kalau saya biasa mulai dari ke-0. Kalau Anda mulai dari 'data1' ya gpp ☺

Selanjutnya kita buat script untuk memprosesnya. Nah... pada bagian ini, mungkin yang sulit adalah bagaimana cara membaca value yang diisikan pada setiap komponen textbox mengingat komponen ini dihasilkan secara dinamis (dengan jumlah textbox yang berubah-ubah).

OK... trik untuk membaca value yang berasal dari komponen yang dinamis seperti ini adalah: **kita memanfaatkan perulangan saja**

```

for ($i=0; $i <= jumlahkomponen - 1; $i++)
{
    $bilangan[$i] = $_POST['data'.$i];
}

```

Perhatikan! Mengapa pada bagian \$_POST berbentuk \$_POST['data'.\$i] ?

Ya... karena nama komponen dari form kan bentuknya name='data0', name='data1', name='data2', dst. Maka dalam membaca valuenya harusnya juga menggunakan \$_POST['data0'], \$_POST['data1'], dst. Nah... supaya mudah, maka kita gunakan \$_POST['data'.\$i] dengan \$i nya diperoleh dari FOR. Simpel bukan? Dengan alasan inilah mengapa saya suka memberi nama komponen dalam formnya mulai dari 0 terlebih dahulu, seperti name='data0', name='data1', dst...

Kemudian... muncul masalah lagi yang lain. Untuk FOR nya harusnya \$i bergerak dari 0, 1, 2, sampai dengan 'jumlah komponen' dikurangi 1. Nah.. yang menjadi masalah, bagaimana mendapatkan 'jumlah komponen' pada script ini? Padahal jumlah komponen ini berada pada form2.php. Atau dengan kata lain, bagaimana caranya jumlah komponen yang berada pada form2.php ini juga bisa dikenali di script untuk memproses inputnya

ini? Ya... solusinya mudah, kita tinggal buat saja komponen 'hidden textbox' yang berisi jumlah komponennya.

form2.php

```
<h1>Cari Rata-rata Bilangan</h1>
```

```
<?php
$n = $_POST['n'];

echo "<form method='post' action='proses.php'>";
echo "<table>";
for ($i = 0; $i <= $n-1; $i++)
{
    echo "<tr><td>Bil ke-".($i+1)."</td>
        <td><input type='text' name='data'.". $i. "' /></td></tr>";
}
echo "<tr><td></td>
    <td><input type='submit' name='submit' value='Submit' /></td></tr>";

echo "<tr><td></td>
    <td><input type='hidden' name='n' value='". $n. "' /></td></tr>";

echo "</table>";
echo "</form>";

?>
```

Ingat... komponen hidden text ini harus terletak dalam <form></form>. Sesuai namanya, maka komponen ini tersembunyi (tidak tampak dalam browser). Namun begitu user mengklik submit, maka value ini ikut tersubmit untuk diproses pada script pemrosesnya.

OK... masalah teratasi, dan sekarang kita bisa buat scriptnya

proses.php

```
<h1>Cari Rata-rata Bilangan</h1>
```

```
<?php

// membaca jumlah komponen dari hidden textbox

$n = $_POST['n'];
```

```
// membaca value dari masing-masing komponen textbox
```

```
for ($i=0; $i <= $n - 1; $i++)  
{  
    $bilangan[$i] = $_POST['data'][$i];  
}
```

```
// menjumlahkan nilai semua bilangan
```

```
$jumlah = 0;
```

```
for ($i=0; $i <= $n - 1; $i++)  
{  
    $jumlah += $bilangan[$i];  
}
```

```
// hitung rata-rata
```

```
$rataRata = $jumlah/$n;
```

```
echo "Rata-ratanya adalah : ".$rataRata;
```

```
?>
```

Atau cara lain Anda bisa menggabung proses menjumlahkan nilai semua bilangan bersama proses membaca bilangan dari form menjadi satu bagian

proses.php

```
<h1>Cari Rata-rata Bilangan</h1>
```

```
<?php
```

```
// membaca jumlah komponen dari hidden textbox
```

```
$n = $_POST['n'];
```

```
// membaca value dari masing-masing komponen textbox dan menjumlahkannya
```

```
$jumlah = 0;
```

```
for ($i=0; $i <= $n - 1; $i++)  
{  
    $bilangan[$i] = $_POST['data'][$i];  
    $jumlah += $bilangan[$i];  
}
```

```
// hitung rata-rata

$rataRata = $jumlah/$n;

echo "Rata-ratanya adalah : ".$rataRata;

?>
```

Tugas

1. Perhatikan kembali kasus no. 2 pada bab ini. Apakah asumsi bilangan terbesar untuk pertama kali harus diberikan pada bilangan yang pertama? Jelaskan jawaban Anda
2. Identik dengan kasus no. 2 pada bab ini, sekarang coba Anda buat script PHP untuk mencari nilai terkecil dari bilangan-bilangan 283, 182, 381, 119, 391, 591, 123, 124, 284, 215, 312.

Keterangan: tanpa menggunakan form input

3. Identik dengan kasus no. 3 pada bab ini, coba Anda buat form input untuk memasukkan n buah bilangan, yang selanjutnya dicari nilai terbesar dan nilai terkecil, serta nilai jangkauan yang diperoleh dari nilai terbesar dikurangi nilai terkecil.
4. Diketahui sebuah data bilangan berikut ini:

273
281
384
119
392
184
105
129
204
219
274
275
263

Buatlah script untuk fasilitas pencarian data bilangan. Mula-mula buatlah form untuk memasukkan bilangan yang ingin dicari. Selanjutnya apabila ditekan submit, maka berikutnya muncul status 'ada atau tidaknya' bilangan yang dicari

dalam keseluruhan data. Jika ada, maka tampilkan urutan data ke berapa dari keseluruhan data (urutan dimulai dari 1, 2, 3, ...)

Contoh:

Masukkan Bilangan yang dicari = 281 [submit]

Selanjutnya muncul keterangan:

“Bilangan yang Anda cari ada dalam data pada urutan ke-2”

5. Sebuah toko swalayan ingin membuat aplikasi sederhana untuk menghitung total harga yang harus dibayarkan si pembeli. Misalkan dalam toko tersebut hanya terdapat 10 jenis barang saja. Berikut daftar barang yang ada dalam toko beserta harganya

Kode Barang	Nama Barang	Harga Satuan
1001	Sabun Lifebuoy	Rp. 1.500,-
1002	Permen Blaster	Rp. 5.600,-
1003	Pasta Gigi Pepsodent	Rp. 4.560,-
1004	Madu Arbain	Rp. 30.000,-
1005	Kecap ABC	Rp. 7.250,-
1006	Saus Tomat ABC	Rp. 6.700,-
1007	Gula Gulaku	Rp. 8.900,-
1008	Rinso	Rp. 7.100,-
1009	Super Pel	Rp. 6.450,-
1010	Permen Tango	Rp. 5.600,-

Aplikasi ini terletak pada kasir. Berikut ini tampilan aplikasinya

Form pertama

Masukkan banyaknya jenis barang yang dibeli [BANYAK JENIS BARANG]

[SUBMIT]

Form kedua

Masukkan kode barang yang dibeli dan banyaknya

Barang ke -1 [MASUKKAN KODE BARANG] [JUMLAH BRG]
Barang ke -2 [MASUKKAN KODE BARANG] [JUMLAH BRG]
Barang ke -3 [MASUKKAN KODE BARANG] [JUMLAH BRG]

.
.

[SUBMIT]

Tampilan Output

Daftar Barang Yang Dibeli

KODE BARANG	NAMA BARANG	JUM BRG	HARGA
XXX	XXX	XXX	Rp. XXX
XXX	XXX	XXX	Rp. XXX
XXX	XXX	XXX	Rp. XXX

.

.

TOTAL HARGA : Rp. XXX

Contoh:

Misalkan ada konsumen yang membeli sabun Lifebuoy (10 buah), dan permen Blaster (20 pack), maka formnya:

Form Pertama:

Masukkan banyaknya jenis barang yang dibeli [2]

[SUBMIT]

Form Kedua

Masukkan kode barang yang dibeli dan banyaknya

Barang ke -1 [1001] [10]

Barang ke -2 [1002] [20]

[SUBMIT]

Tampilan Output

Daftar Barang Yang Dibeli

KODE BARANG	NAMA BARANG	JUM BRG	HARGA
1001	Sabun Lifebuoy	10	Rp. 15000,-
1002	Permen Blaster	20	Rp. 112000,-

TOTAL HARGA : Rp. 127000,-