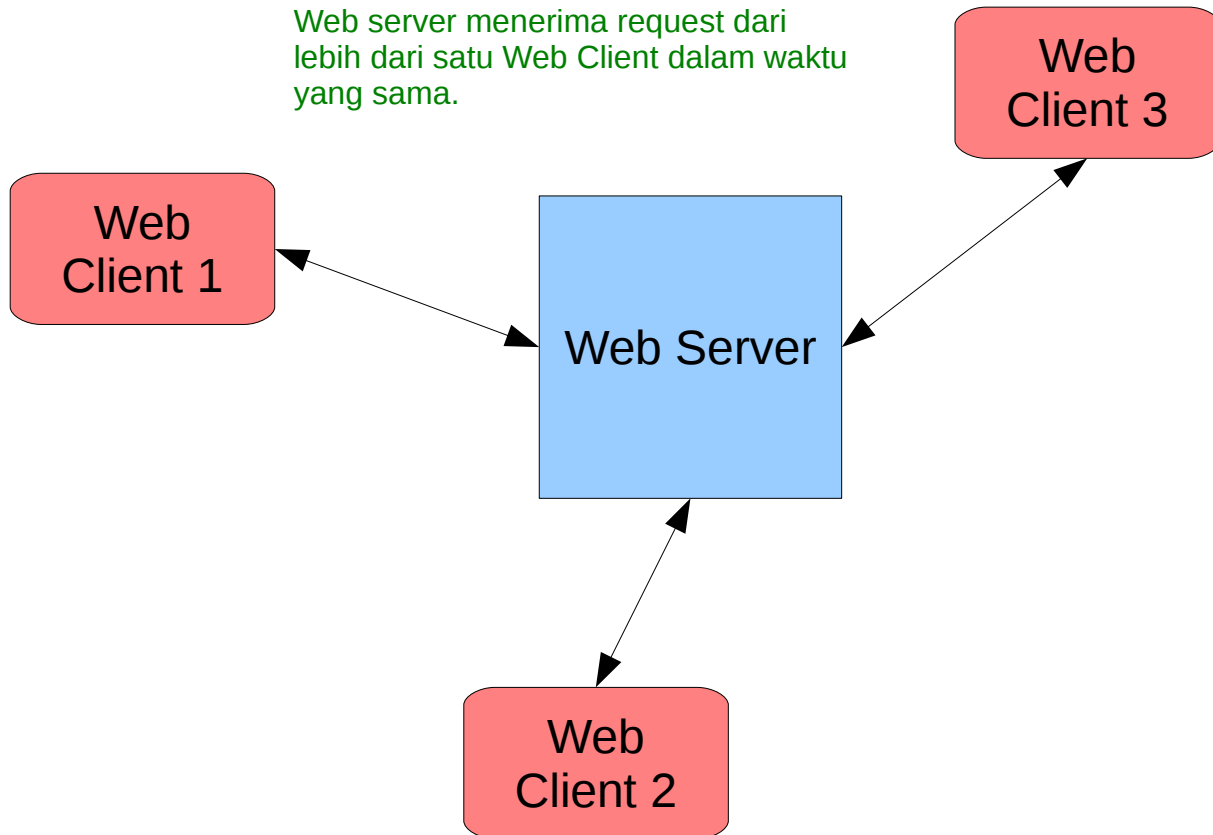
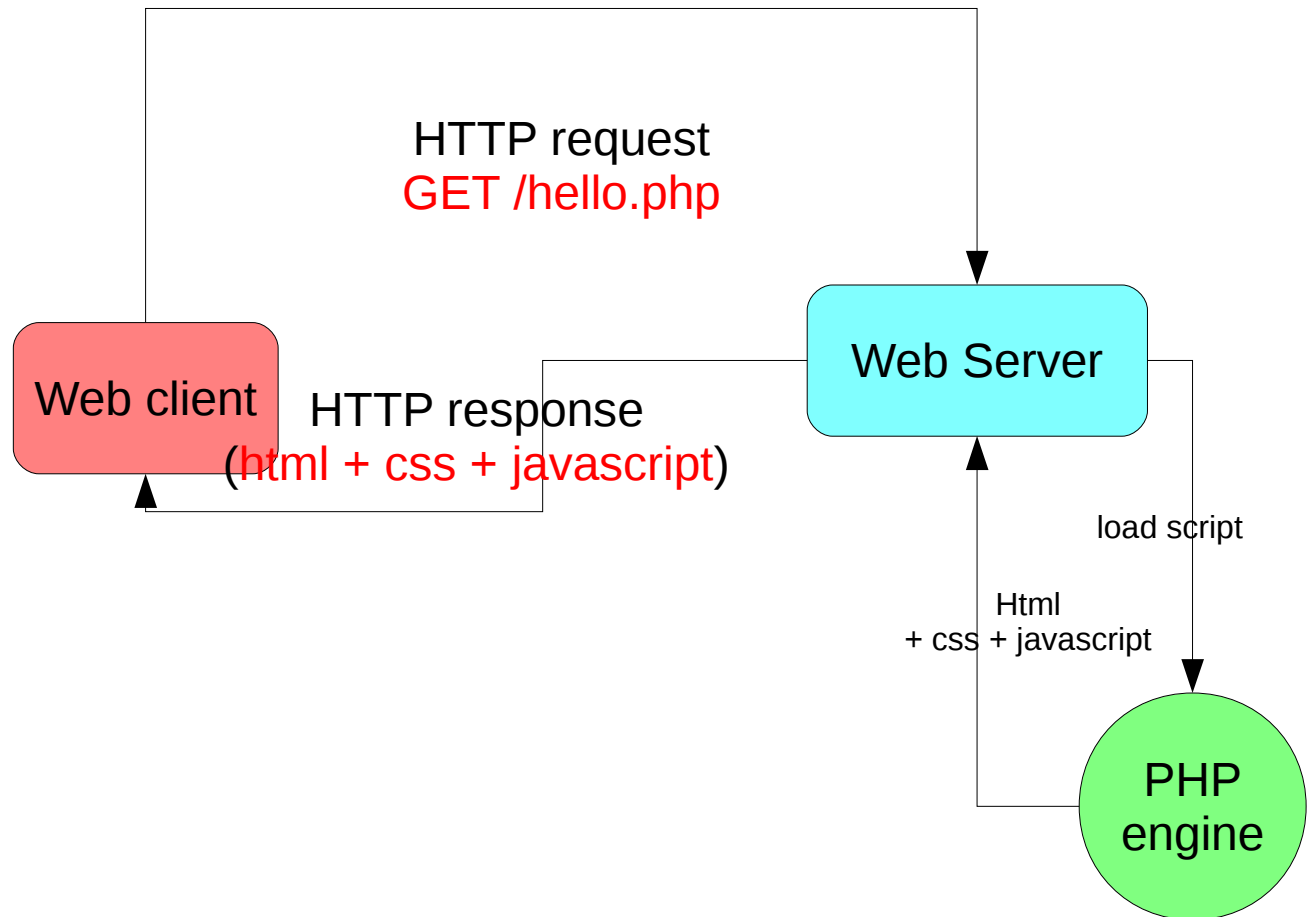
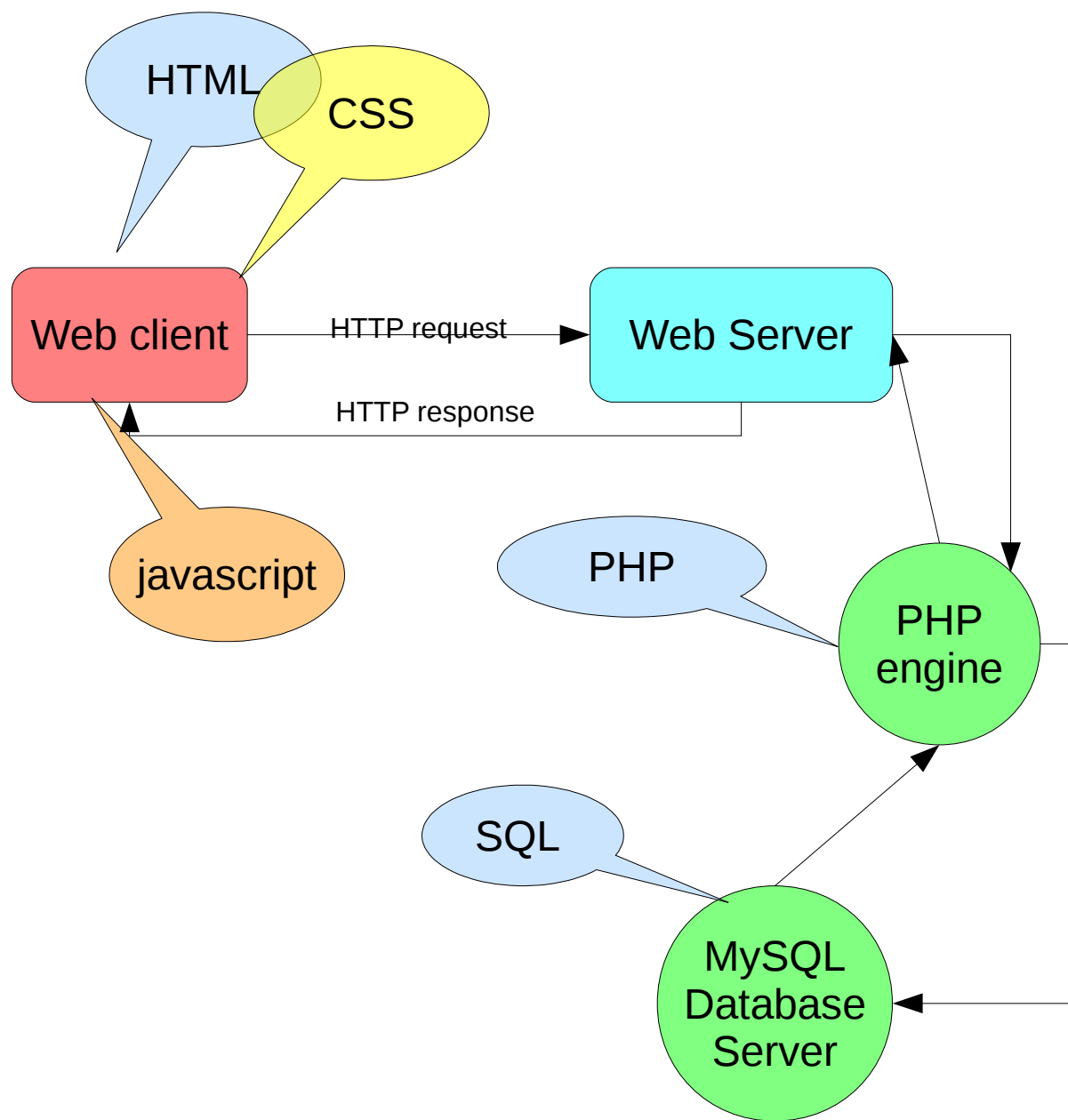


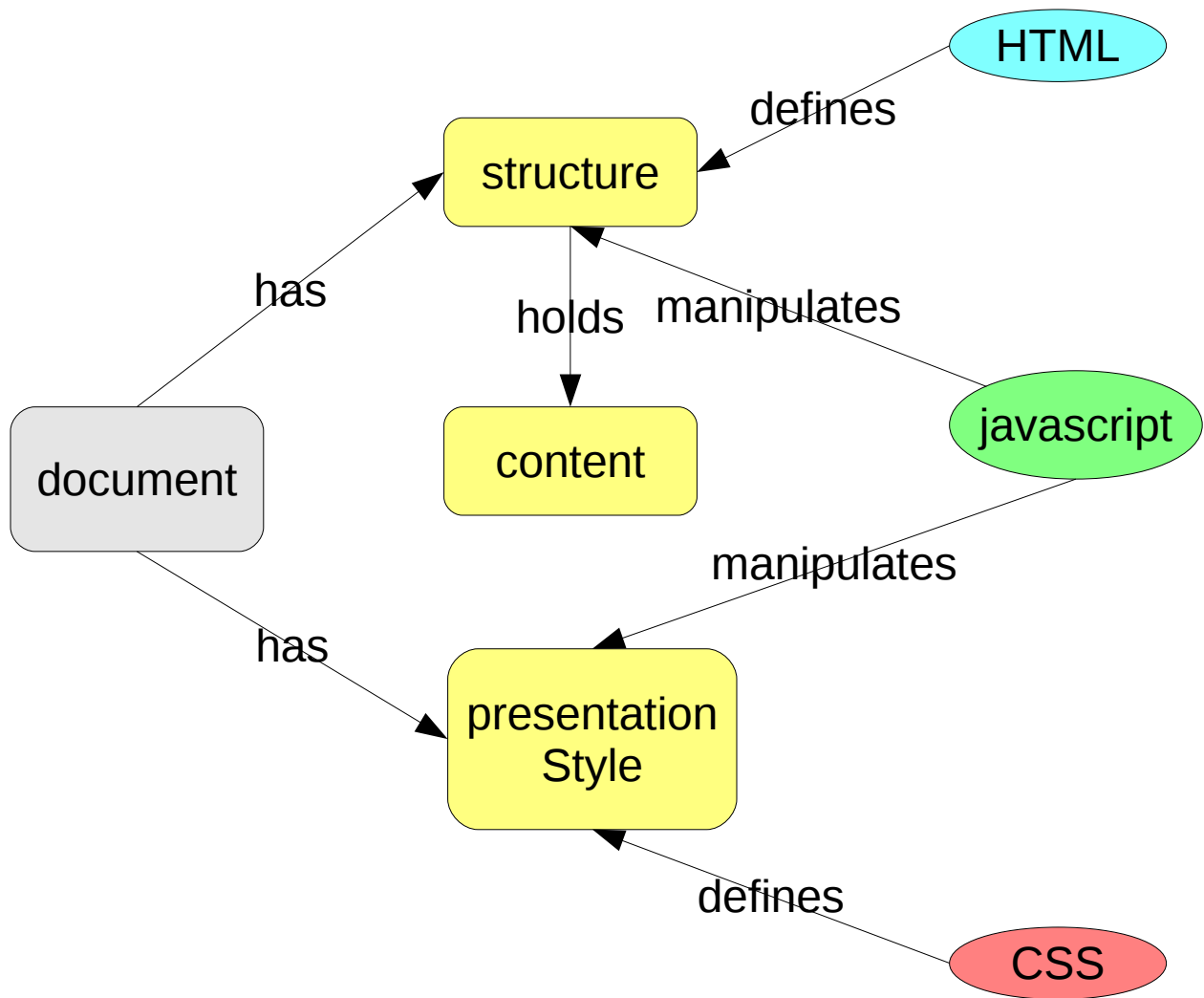
Web server menerima request dari lebih dari satu Web Client dalam waktu yang sama.



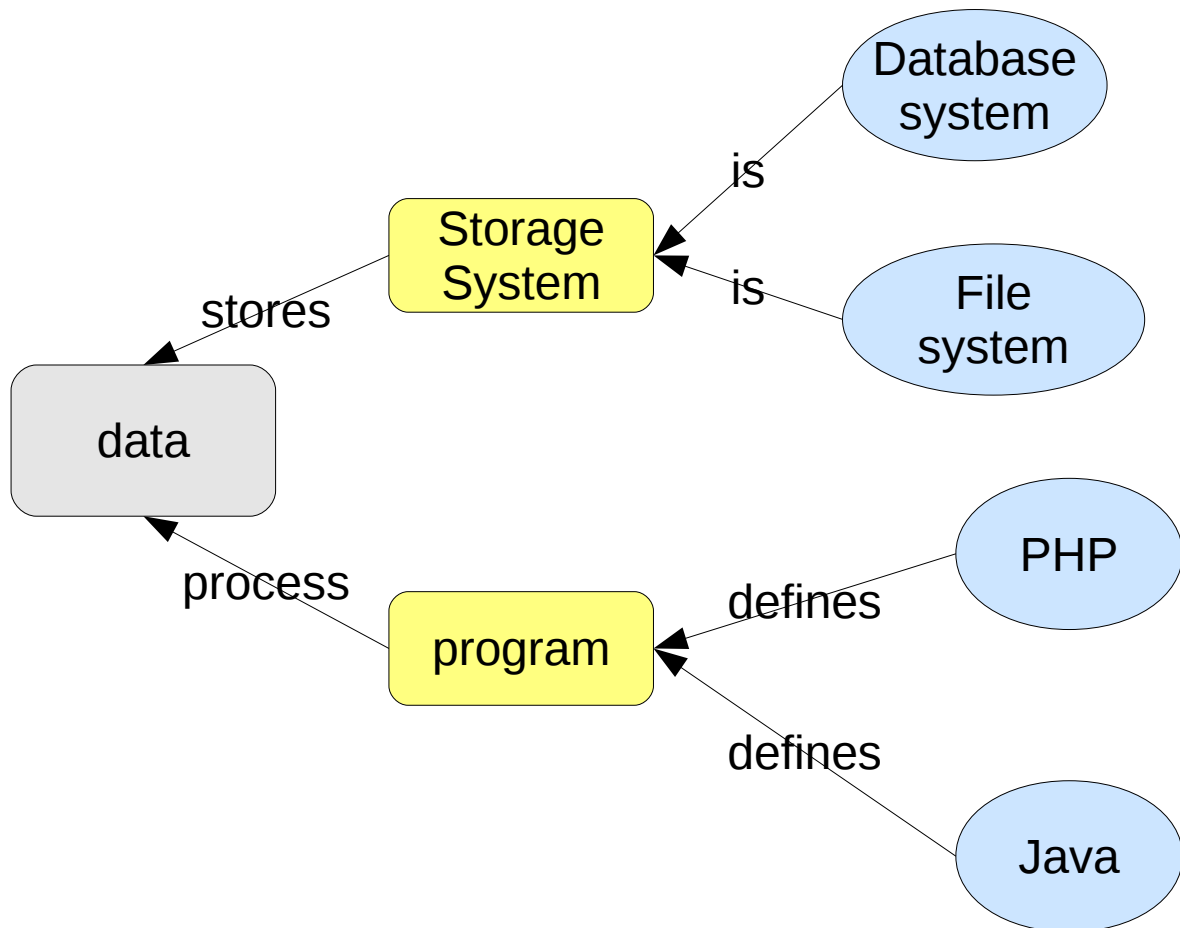




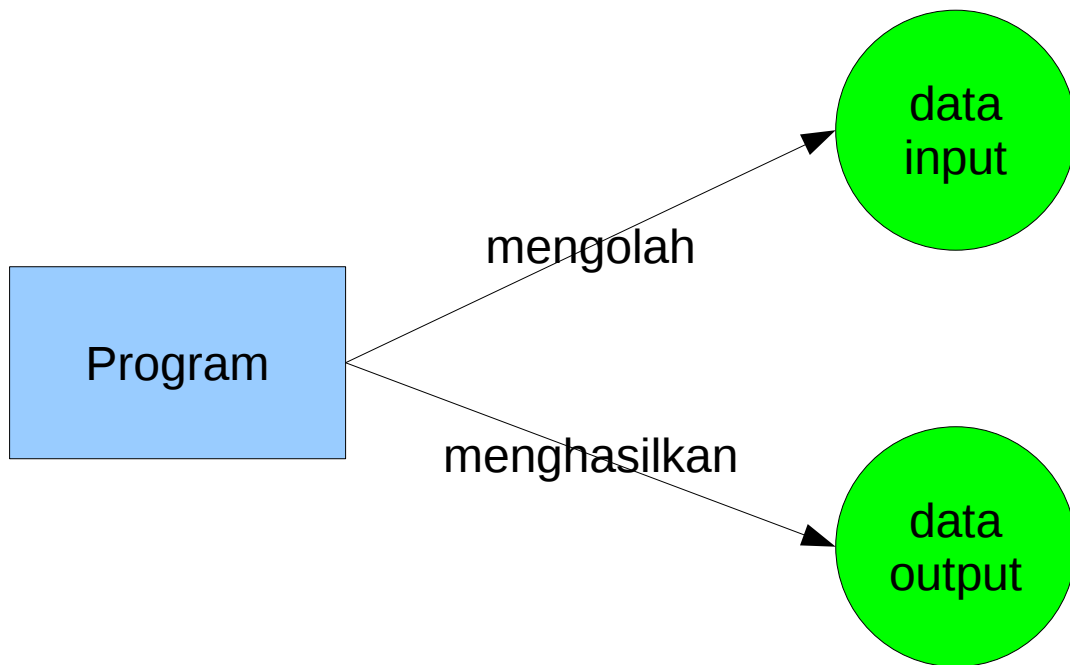
Web Client Side



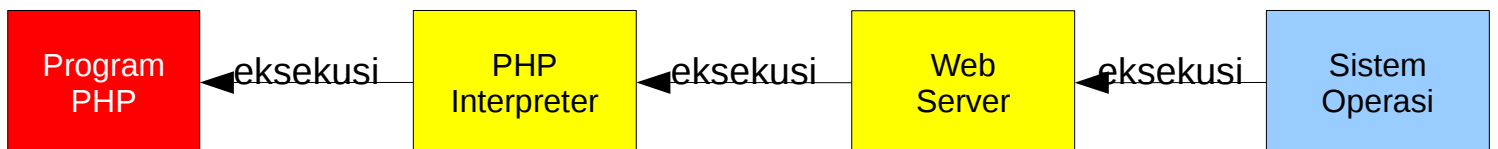
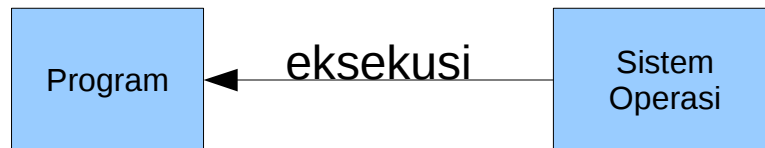
Web Server Side



Program Komputer



Model Eksekusi Program



Program PHP

```
<html>
  <head>
    <title>Contoh Program PHP</title>
  </head>
  <body>

<?php
  echo "Ini Program PHP";
?>

<p>Ini bagian dari HTML</p>
<!-- ini komentar dalam HTML -->

<?php
// ini komentar dalam PHP
  $n = 1;
// mencetak 9 baris
  while($n < 10) {
    echo "Ini baris ke $n", "<br/>";
    $n++; // sama dengan $n = $n+1
  }
?>

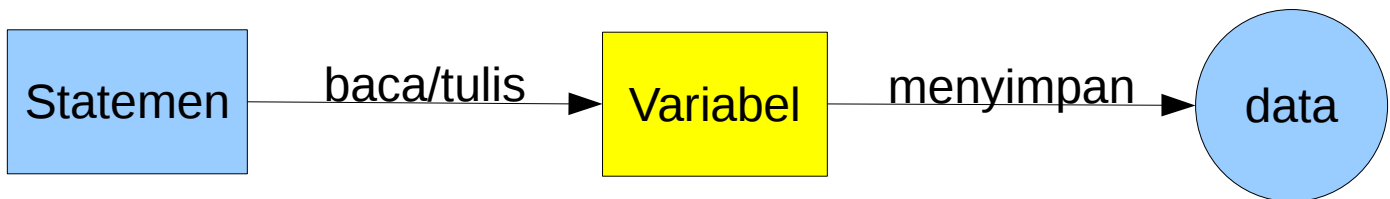
  </body>
</html>
```


Statemen & komentar

```
<?php
// ini statemen pertama
echo "Hai apa kabar ?";
// ini kedua
echo "Selamat belajar";
// ini yang ketiga
echo "PHP";
?>
```

```
<?php
/* statemen-statemen boleh
ditulis dalam satu baris,
asalkan tetap dipisah tanda ;
*/
echo "Hai apa kabar ?";echo
"Selamat belajar"; echo "PHP";
?>
```

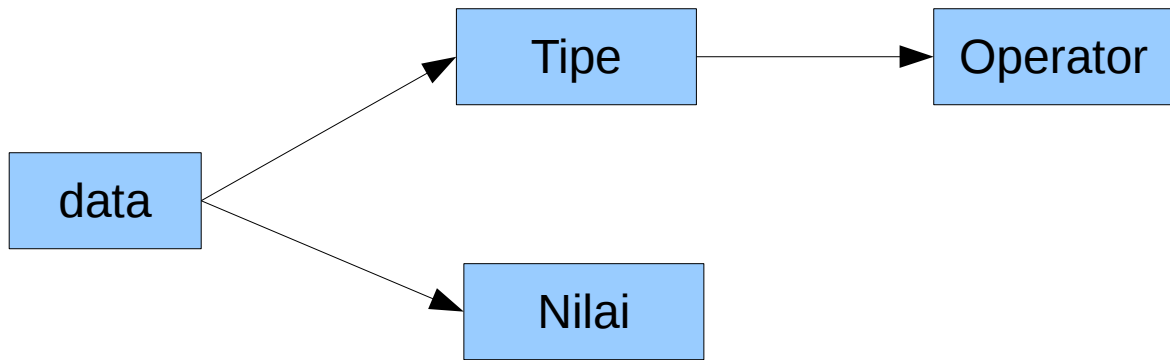
Variabel



```
// statemen yg menulis variabel  
$dataNama = "Fadhilah";  
  
// statemen yg membaca variabel  
echo $dataNama, "<br/>";
```

```
// statemen yg menulis variabel  
$kenalan = "Nama Saya...  
...Fadhilah";  
// statemen yg membaca variabel  
echo $kenalan, "<br/>";
```

Tipe Data



Tipe Data	Contoh
string	"Apa kabar"
integer	234
float	23.4
boolean	TRUE, FALSE
Object	
array	
Resource	
null	NULL

Operator assignment =

Salin nilai di sisi kanan ke variabel di sisi kiri

```
$a = 100;  
$b = $a;  
  
//mendeklarasikan 2 var, sekaligus  
//inisialisasi var  
$c = $d = 200;
```

mengetahui tipe variabel

```
$bil = 100;  
$t = gettype($bil);  
echo "tipenya: $t <br/>";  
  
$str = "salamu'alaikum";  
$t = gettype($str);  
echo "tipenya: $t <br/>";
```

String literal

String dapat didefinisikan menggunakan kutip tunggal (') atau kutip ganda (")

```
$str1 = 'waktu yang tepat';  
$str2 = "untuk belajar PHP";
```

Parsing string

Variabel \$str akan berisi nilai "ini hari jum'at"

```
$hari = 'jum\'at';  
  
// $hari disubstitusi dengan  
// nilai literal yang disimpan pada  
// variabel $hari  
$str = "ini hari $hari";  
      ini hari jum'at
```

Variabel \$str akan berisi nilai "ini hari \$hari"

```
$hari = "jum'at";  
  
// $hari adalah string biasa (literal)  
// $hari tidak dibaca sebagai variabel  
$str = 'ini hari $hari';  
      ini hari $hari
```

Statemen if

```
if ( <ekspresi bertipe boolean> )
{
    // statemen-statemen yang dieksekusi
    // hanya jika <ekspresi bertipe boolean>
    // adalah true
}
```

if - else

```
if ( <ekspresi bertipe boolean> )
{
    // blok statemen yang dieksekusi
    // hanya jika <ekspresi bertipe boolean>
    // adalah true
}
else
{
    // blok statemen yang dieksekusi
    // hanya jika <ekspresi bertipe boolean>
    // adalah false
}
```

if – else if - else

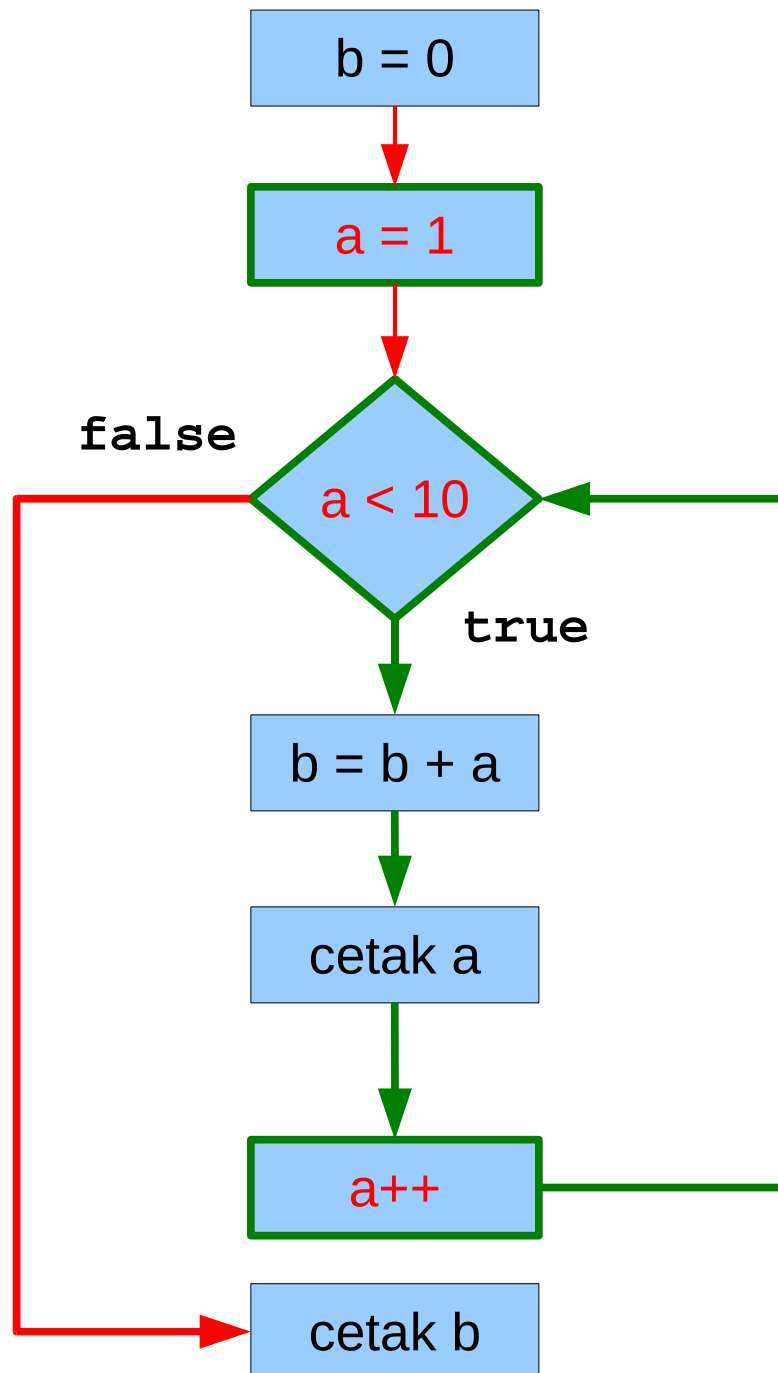
```
$komentar = '';  
if ($umur < 40 && $umur >= 17)  
{  
    $komentar = "dewasa masih muda";  
}  
else if ($umur < 9 && $umur > 5)  
{  
    $komentar = "anak-anak";  
}  
else if ($umur < 17 && $umur >= 9)  
{  
    $komentar = "remaja";  
}  
else if ($umur <= 5 && $umur > 0)  
{  
    $komentar = "balita";  
}  
else if ($umur > 40)  
{  
    $komentar = "sudah tidak muda";  
}  
else  
{  
    $komentar = "data tidak sah";  
}
```

Statemen while

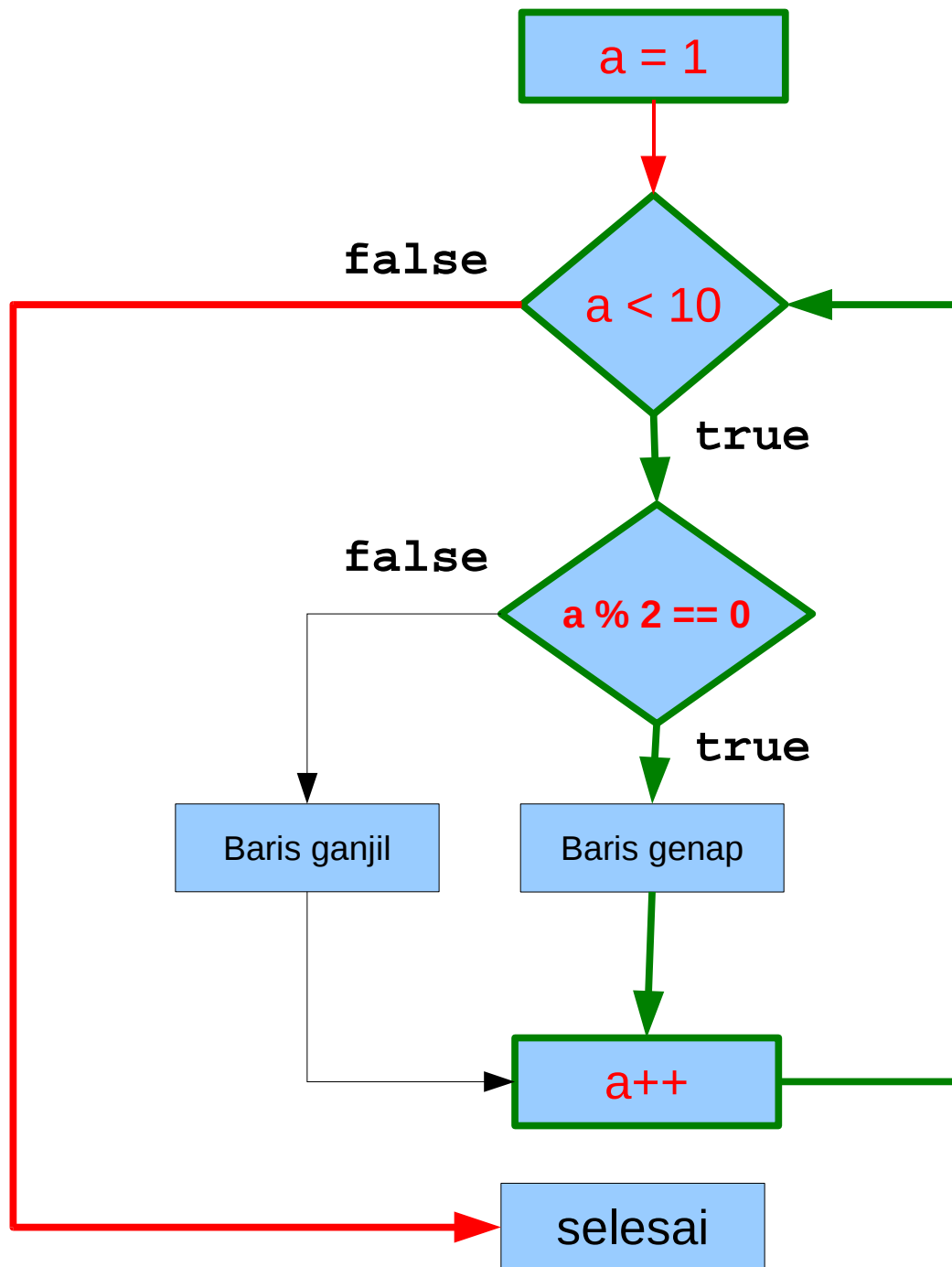
```
while(<ekspresi boolean>)  
{  
    // blok statemen ini  
    // di eksekusi secara ber-ulang  
    // selama <ekspresi boolean>  
    // bernilai TRUE  
}
```

```
//hitung: 1+2+3+4+5+6+7+8+9  
$b = 0; $a = 1;  
while($a<10)  
{  
    $b = $b + $a;  
    echo $a, "<br/>";  
    $a++; //sama: $a = $a + 1  
}  
echo "total: $b <br/>";
```


Flow chart untuk statemen while



Flow chart untuk statemen while



Statemen for

```
$b = 0;  
for($a=1; $a<10; $a++)  
{  
    $b = $b + $a;  
    echo "$a <br>";  
}  
echo "total: $b <br/>";
```

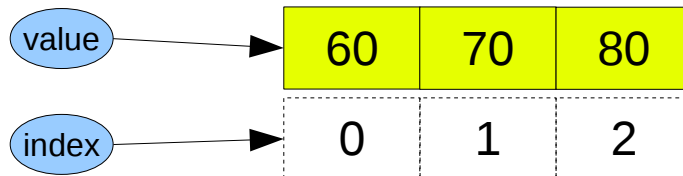
\$a++ sama dengan
\$a = \$a + 1

Array 1 dimensi

```
$a = array(60,70,80); // cara 1
```

```
$a[] = 60; $a[]=70; $a[] = 80; // cara 2
```

```
$i=0; $a[$i++]=60; $a[$i++]=70; $a[$i++]=80; // cara 3
```



```
echo $a[0]; // 60  
echo $a[1]; // 70  
echo $a[2]; // 80
```

```
$k = 0;  
echo $a[$k++]; // 60  
echo $a[$k++]; // 70  
echo $a[$k]; // 80
```

Array 2 dimensi

```
$x[] = array(60,70,80);  
$x[] = array(40,50);  
$x[] = array(90,10,20,30);
```

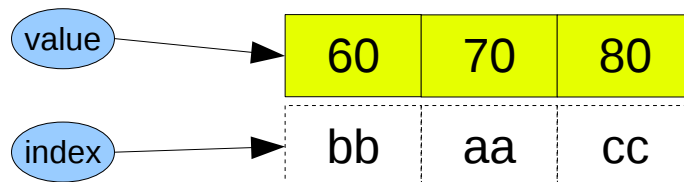
60	70	80	40	50	90	10	20	30
0			1		2			

```
echo $x[0][0]; // 60  
echo $x[1][0]; // 40  
echo $x[2][0]; // 90
```

Array 1 dimensi

```
$a = array('bb'=>60,'aa'=> 70,'cc' => 80); // cara 1
```

```
$a['bb']=60; $a['aa']=70; $a['cc'] = 80; // cara 2
```



```
echo $a['bb']; // 60  
echo $a['aa']; // 70  
echo $a['cc']; // 80
```

Array 2 dimensi

```
$x['yy'] = array('bb'=>60,'aa'=>70,'cc'=>80);  
$x['zz'] = array('bb'=>40,'aa'=>50);  
$x['xx'] = array('dd'=>90,'ee'=>10,'bb'=>20,'cc'=>30);
```

60	70	80	40	50	90	10	20	30
bb	aa	cc	bb	aa	dd	ee	bb	cc
yy			zz		xx			

```
echo $x['yy']['bb']; // 60  
echo $x['zz']['bb']; // 40  
echo $x['xx']['bb']; // 20
```

Array

```
// inisialisasi
$nilai = array(80,70,60);

// membaca menggunakan index array
echo $nilai[0]; // mencetak 80

$idx = 2;
echo $nilai[$idx]; // mencetak 60

// mengubah nilai sebelumnya
$nilai[1] = 86;
$nilai[$idx] = 88;

// menambah elemen diposisi terakhir
$nilai[] = 77;
$nilai[] = 99;

// mencetak nilai variabel utk debugging
print_r($nilai);
```

Indeks Array

array dapat menggunakan kunci indeks yang unik, yang dapat bertipe data apapun. Misal: integer, string atau objek

```
// inisialisasi
$days = array('senin' => 'monday', 'selasa'
=> 'tuesday', 'rabu' => 'wednesday' , 'kamis'
=> 'thursday', "jum'at" => 'friday');

// membaca menggunakan index array
echo $days['rabu']; // mencetak 'wednesday'

$idx = 'rabu';
echo $days[$idx]; // mencetak 'wednesday'

// menimpa (overwrite) nilai sebelumnya
$days['senin'] = 'isnaini';
$days[$idx] = 'doornertag';

// menambah elemen baru
$days['sabt'u'] = 'saturday';
```


Iterasi elemen array dengan foreach

```
// inisialisasi
$daftar = array(80,70,60);

// menggunakan foreach mencetak:
// 80<br>70<br>60<br>
foreach($daftar as $v)
{
    echo $v;
    echo "<br>";
}
```

foreach untuk array dengan kunci indeks

```
// inisialisasi
$maks = array(5 =>'E', 30=>'D', 70=>'C',
80=>'B', 100=>'A');

// menggunakan foreach
foreach($maks as $kunci => $nilai)
{
    echo $kunci;
    echo " adalah ";
    echo $nilai;
}
```

Membaca data dari <form>

Jika form menggunakan method POST maka data dibaca dari array \$_POST

Jika form menggunakan method GET maka data dibaca dari array \$_GET

File form.php

```
<form method="post" action="hitung.php">
Harga Satuan <input type="text" name="hrg" />
<br/>
Kuantitas <input type="text" name="qty" /> <br/>
<input type="submit" name="hit" value="Hitung"/>
</form>
```

File hitung.php

```
<?php
echo 'harga adalah ', $_POST['hrg'], '<br/>';
echo 'kuantitas adalah ', $_POST['qty'], '<br/>';
$subtotal = $_POST['hrg'] * $_POST['qty'];
echo 'subtotal adalah ', $subtotal, '<br/>';
?>
```

Tipe data <form> adalah string

Data dari FORM bertipe data **string** atau **array** yang setiap elemennya bertipe **string**.

Jika hendak melakukan operasi yang memerlukan tipe data numerik (integer atau float), maka data dari FORM harus dikonversi ke numerik menggunakan **intval()** atau **floatval()**.

File FormBitOp.php

```
<form method="post" action="bitwise.php">
Harga Satuan <input type="text" name="op1" />
<br/>
Kuantitas <input type="text" name="op2" /> <br/>
<input type="submit" name="hit" value="BITWISE
AND" />
</form>
```

File bitwise.php

```
<?php
$hasil = $_POST['op1'] & $_POST['op2'];
echo "$op1 & $op2 = ", $hasil, '<br/>';
?>
```

Pada program di atas jika nilai op1 adalah '**27**' dan op2 adalah '**3**' maka nilai hasil adalah '**2**'. Padahal, nilai hasil yang benar adalah **3**.

include & require

awal.php

```
echo "<h1>$judul</h1>";  
echo "<h2>baris pembuka</h2>";
```

utama.php

include

```
$judul = 'baris atas';  
include 'awal.php';  
echo '<p>baris 1</p>';  
echo '<p>baris 2</p>';
```

HTML output

```
<h1>baris atas</h1>  
<h2>baris pembuka</h2>  
<p>baris 1</p>  
<p>baris 2</p>
```

include

Jika terjadi Fatal Error di *included file*, maka *tidak* menghentikan eksekusi script utama

```
$judul = 'baris atas';  
// pada contoh ini tidak ada file awal_xyz.php  
include 'awal_xyz.php';  
echo '<p>baris 1</p>';  
echo '<p>baris 2</p>';
```

require

Jika terjadi Fatal Error di *required file*, maka *menghentikan* eksekusi script utama

```
$judul = 'baris atas';  
// pada contoh ini tidak ada file awal_xyz.php  
require 'awal_xyz.php';  
echo '<p>baris 1</p>';  
echo '<p>baris 2</p>';
```

fungsi

Mendefinisikan Fungsi di file
fungsi.php

```
function cetak_list($judul,$arr)
{
    echo "<h2>$judul</h2><ol>";
    foreach($arr as $a)
    {
        echo "<li>$a</li>";
    }
    echo '</ol>';
}
```

Menggunakan fungsi yang ada di
file fungsi.php dari file lain

```
// supaya fungsi dapat dieksekusi
require 'fungsi.php';
$j = 'daftar nama';
$nama = array('amir','cici','ito');
cetak_list($j,$nama);
```

return

Mengembalikan nilai dari fungsi

Mendefinisikan Fungsi di file
fungsi.php

```
function hitung_total($arr)
{
    $ttl = 0;
    foreach($arr as $a)
    {
        $ttl = $ttl + $a;
    }
    return $ttl;
}
```

Menggunakan fungsi yang
mengembalikan (*returns*) nilai

```
require 'fungsi.php';
$nilai = array(20,80,70);
$hasil = hitung_total($nilai);
```

return (contoh lain)

Cetak list dengan fungsi yang mengembalikan nilai

```
//mengembalikan string berisi list
function buat_list($judul,$arr)
{
    $s = "<h2>$judul</h2><ol>";
    foreach($arr as $a)
    {
        $s = $s."<li>$a</li>";
    }
    $s = $s . '</ol>';
    return $s;
}
```

Menggunakan fungsi buat_list

```
// supaya fungsi dapat dieksekusi
require 'fungsi.php';
$j = 'daftar nama';
$nama = array('amir','cici','ito');
$hasil = buat_list($j,$nama);
echo $hasil;
```


Fungsi isset()

Variabel yang sah untuk dibaca adalah yang sudah di-inisialisasi atau di-set.

Fungsi **isset()** berguna untuk mengetahui jika sebuah variabel (termasuk yang bertipe array) *sudah* di-set atau *belum* di-set.

Variabel yang di-set **null** dianggap *belum* di-set.

```
$x = 123;
// variabel $x sudah di-inisialisasi
if ( isset($x) )
{
    echo "x sudah di inisialisasi";
}
else
{
    echo "x belum di inisialisasi";
}

// variabel $y belum di-inisialisasi
if ( isset($y) )
{
    echo "y sudah di inisialisasi";
}
else
{
    echo "y belum di inisialisasi";
}
```

Fungsi empty()

Variabel yang sudah di-set, tapi bernilai 0 atau string kosong (") atau array kosong dianggap kosong atau *empty*.

Fungsi *empty()* berguna untuk mengetahui jika sebuah variabel bernilai kosong.

Fungsi *empty()* sering dikombinasikan dengan operator NOT (!). untuk menguji kondisi bernilai TRUE = **NOT empty**.

Variabel yang tidak kosong (*not empty*) pasti *sudah di-set*.

```
// variabel $x tidak empty
$x = 123;
if ( !empty($x) ) {
    echo "x tidak kosong";
} else{
    echo "x kosong";
}

$y = '';
// variabel $y empty
if ( !empty($y) ) {
    echo "y tidak kosong";
} else {
    echo "y kosong";
}
```

bermacam2 empty

```
// semua variabel
// ini adalah
// empty

// array kosong
$a = array();

// string kosong
$str = '';

// null
$y = null;

// bilangan 0
$num = 0;

// boolean false
$salah = FALSE;
```

Operator ===

```
// membedakan empty
// dengan nilai 0 ?

if($x === 0){
// benar2 bernilai 0
}

// dengan null ?
if($x === null){
// benar2 null
}

// dengan false ?
if($x === false){
// benar2 false
}
```

Fungsi-fungsi array

Fungsi `count()` untuk menghitung banyaknya elemen array

```
$arr = array('a','b','c','d');  
$num = count($arr);  
echo $num; // 4
```

Fungsi `array_keys()` untuk menghitung banyaknya elemen array

```
$arr['nama'] = 'telur ayam';  
$arr['harga'] = 10000;  
$arr['satuan'] = 'kg';  
$keys = array_keys($arr);  
// nama harga satuan  
foreach($keys as $k) {  
    echo $k." ";  
}
```

Fungsi `sort()` untuk mengurutkan elemen array sesuai tipe datanya.

```
$arr = array('34','212','7');  
// sort sebagai string  
sort($arr,SORT_STRING);  
// 212 34 7  
foreach($arr as $a) {  
    echo $a.' ';  
}  
  
$arr = array('34','7','212');  
// sort sebagai bilangan  
sort($arr,SORT_NUMERIC);  
// 7 34 212  
foreach($arr as $a) {  
    echo $a.' ';  
}
```