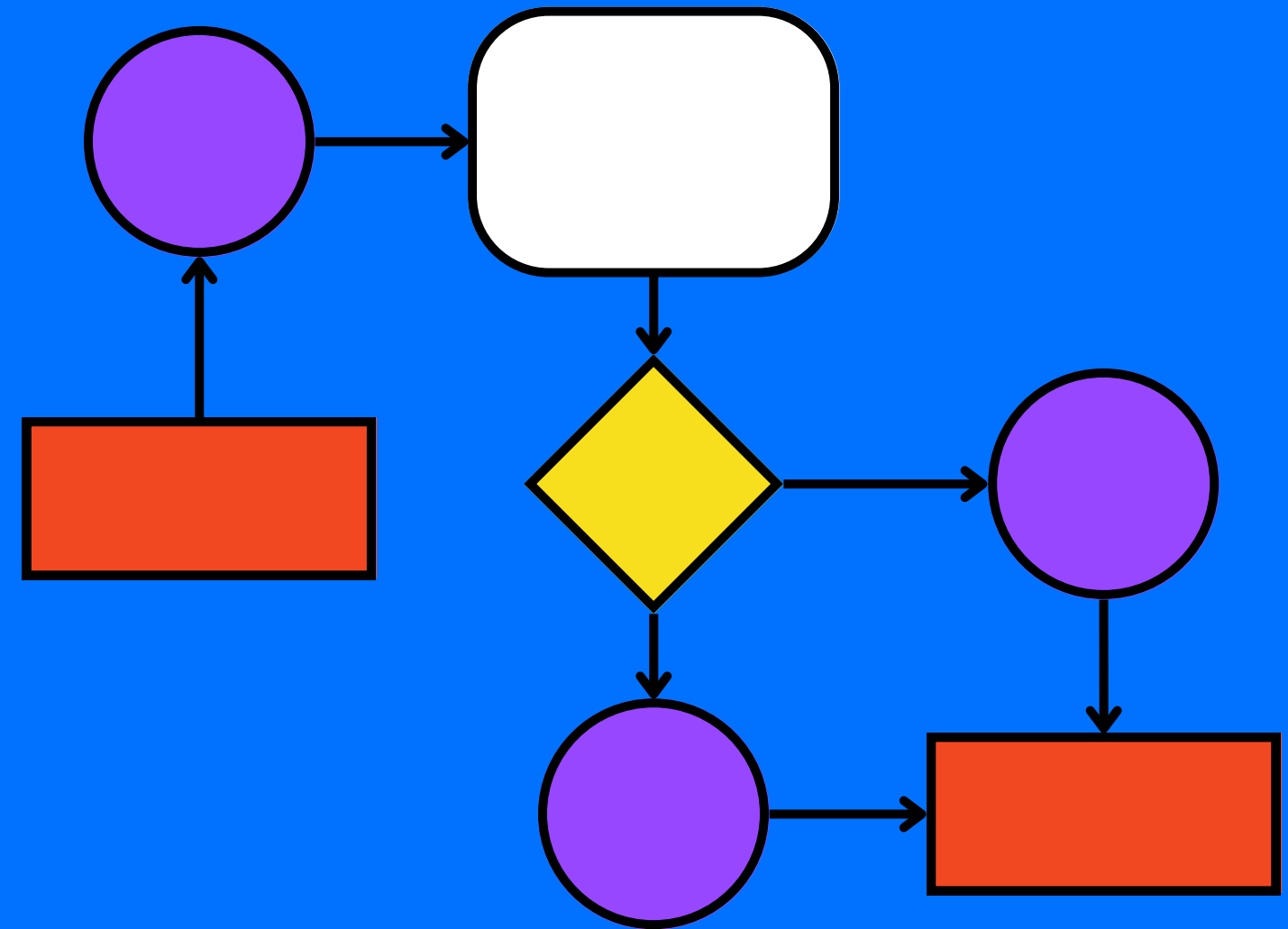


**Jefri Maruli**

# Back-End Development

## Materi 2 (Part 5)



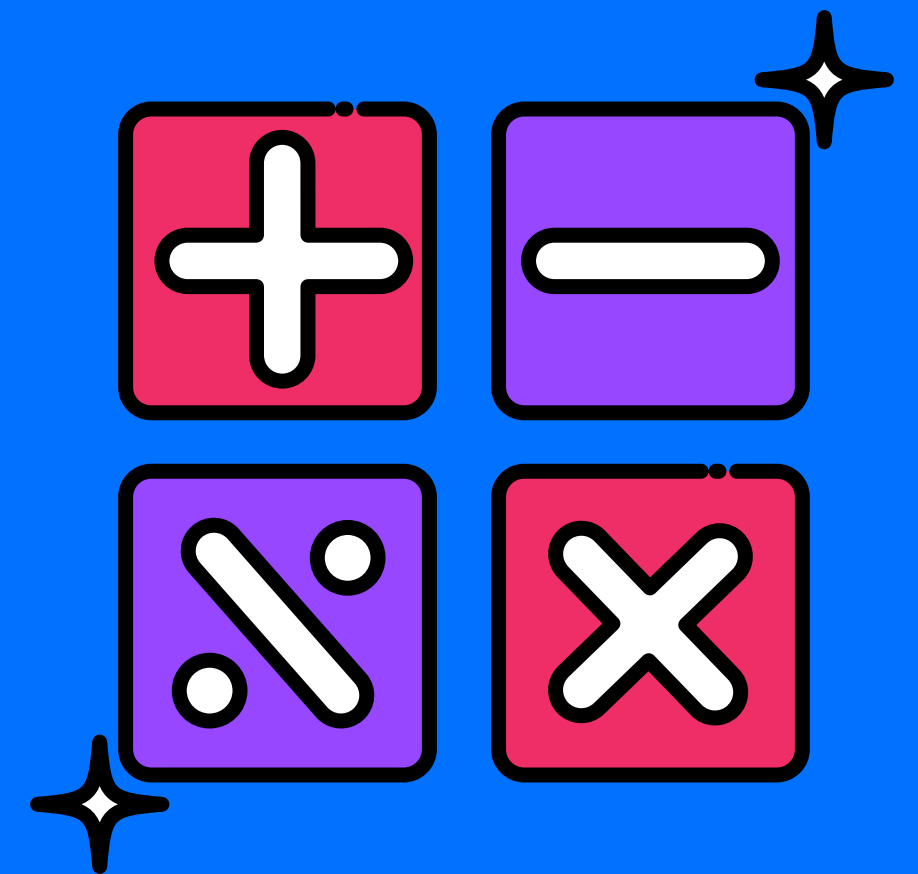
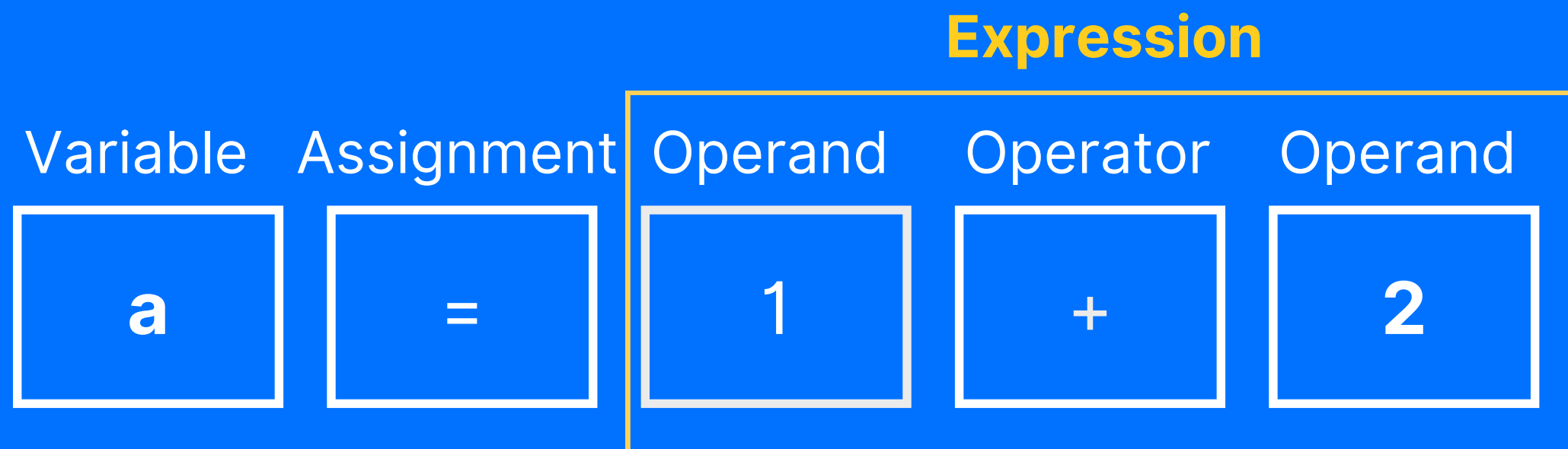
# Agenda Belajar

- Expression
- Statement
- Function

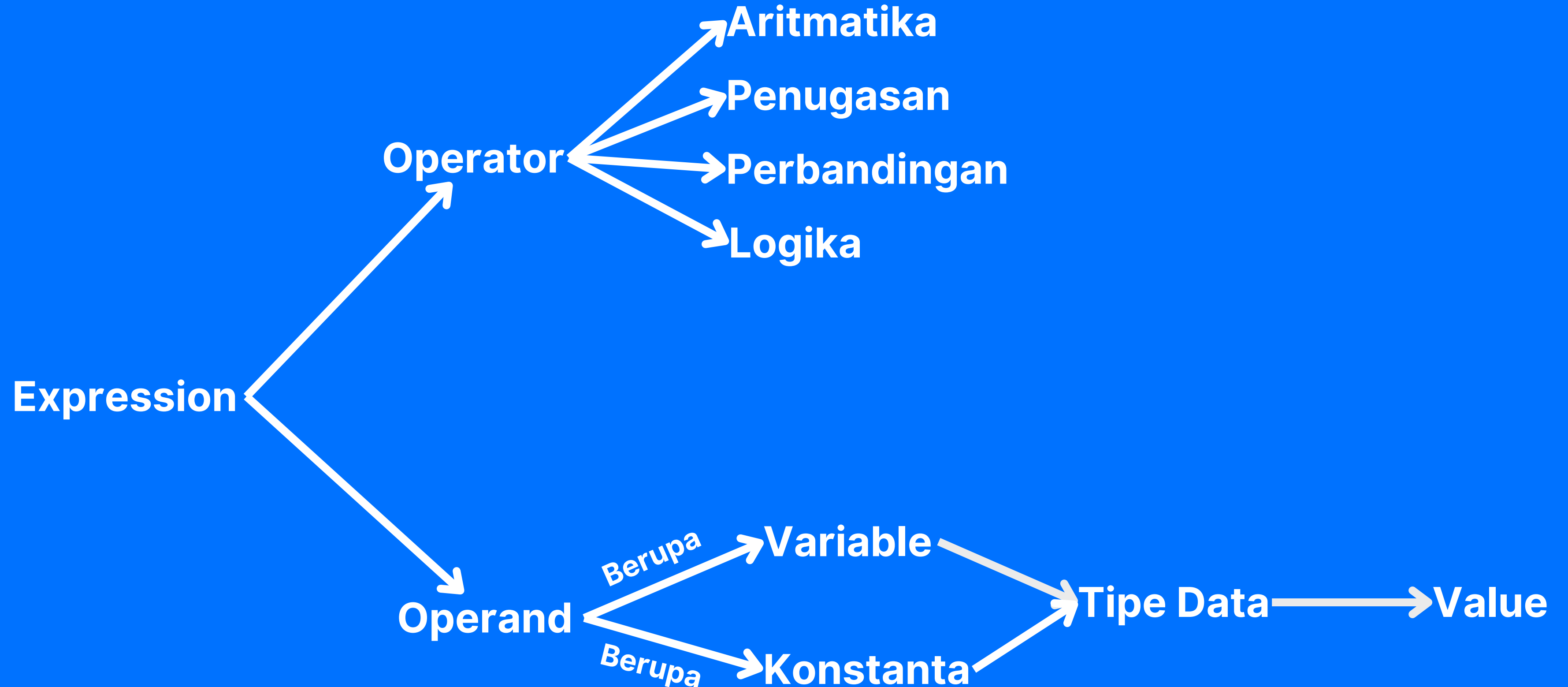


# Expression

Expression merupakan perpaduan dari dua komponen yaitu operator dan operand yang berfungsi untuk mengacu pada sesuatu yang dapat dihitung atau menghasilkan suatu nilai.



# Expression Tree Point



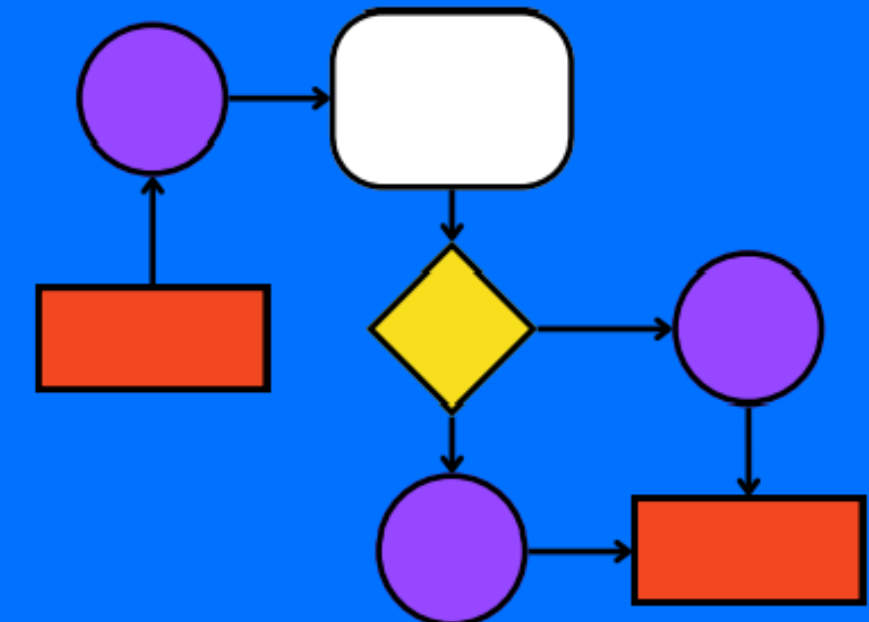
# Contoh Expression



```
1  <?php
2
3  $a = 1 + 1;
4  $b = 3 - 2;
5  $a *= 2;
6  $c = $a + $b;
7  $d = $a / $b;
8  $e = $a % $b;
9  $f = $a ** $b;
10 $g = $a < $b;
11 $h = $a > $b;
12 $i = $a != $b;
13 $j = $a == $b;
14 $k .= $a;
15 $b = TRUE && FALSE;
```

Jefri Maruli

## Back-End Development Materi 2 (Part 4)



Untuk mempelajari tentang expression lebih lengkap, silahkan mempelajari materi sebelumnya tentang operator, karena operator adalah bagian dari expression itu sendiri.

# Statement

statement merupakan bagian program yang berisi perintah yang akan dieksekusi atau dijalankan, karena itu statement menentukan bagaimana suatu variable dimanipulasi atau berubah kondisional statement

## Statement

### Expression

Variable

Assignment

Operand

Operator

Operand

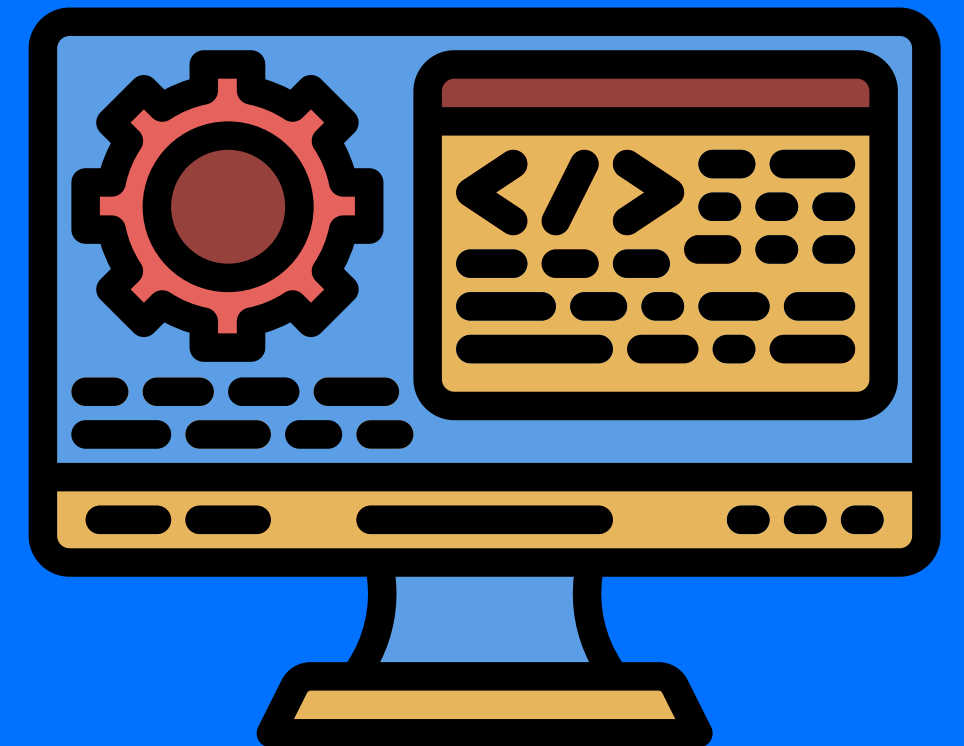
**a**

**=**

**1**

**+**

**2**



# Contoh Statement

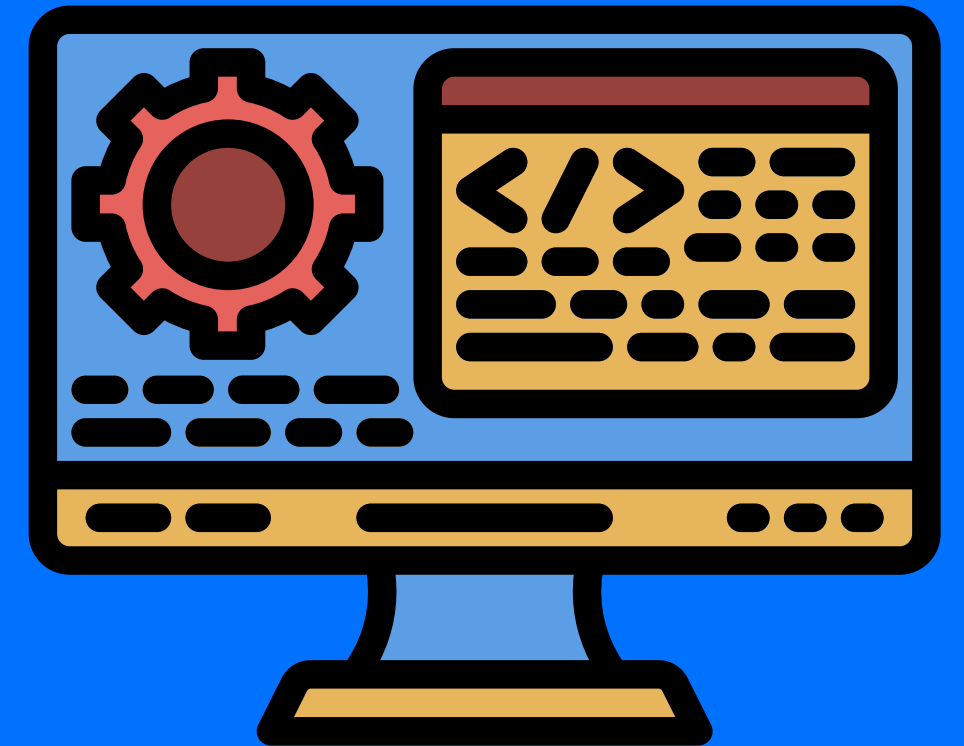


mardirabootcamp - Statement.php

```
1  <?php
2
3  $statement = 3; # statement untuk mendeklarasikan variable
4
5  echo $statement; # statement untuk memanggil variable
```

# Jenis Statement

- **Expression Statement / Inline Statement**
- **Compound Statement / Block Statement**
- **Control Statement**



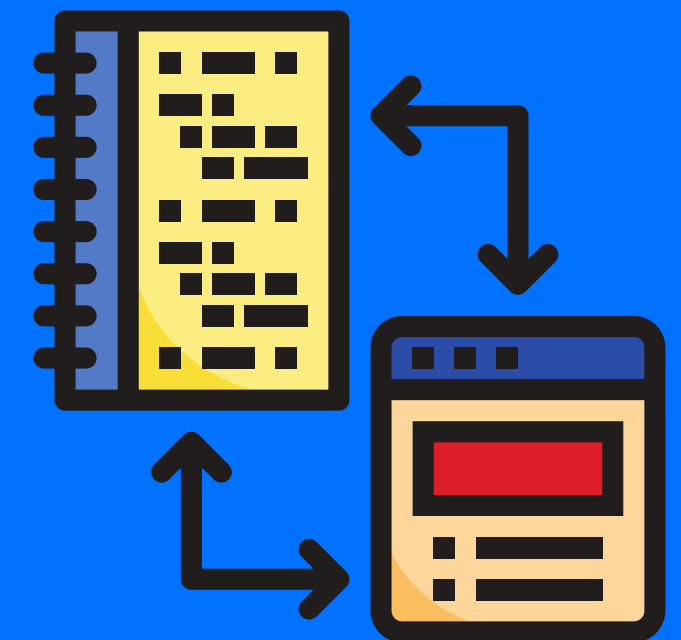


# Expression Statement/Inline Statement

Expression statement adalah statement yang di dalam perintahnya terdapat expression dan diikuti dengan tanda titik koma (;) di akhir perintah.



```
1  <?php
2
3  // expression statement
4  $a = 1 + 1 * 2 - 3 / 4 % 5 ** 6;
5
6  // statement
7  echo $a;
```



# Implementasi Expression Statement

```
mardirabootcamp - ExpressionStatement.php

1  <?php
2
3  echo "<pre>";
4  // expression statement
5  $a = 1 + 1 * 10 - 2 / 1;
6  var_dump($a);
7
8  // expression statement addition
9  $b = $a + 1;
10 var_dump($b);
11
12 // expression statement subtraction
13 $c = $a - $b;
14 var_dump($c);
15
16 // expression statement multiplication & division
17 $d = $a / $b * $c;
18 var_dump($d);
19
20 // expression statement comparison
21 $e = $b > $c;
22 var_dump($e);
23 echo "</pre>";
```

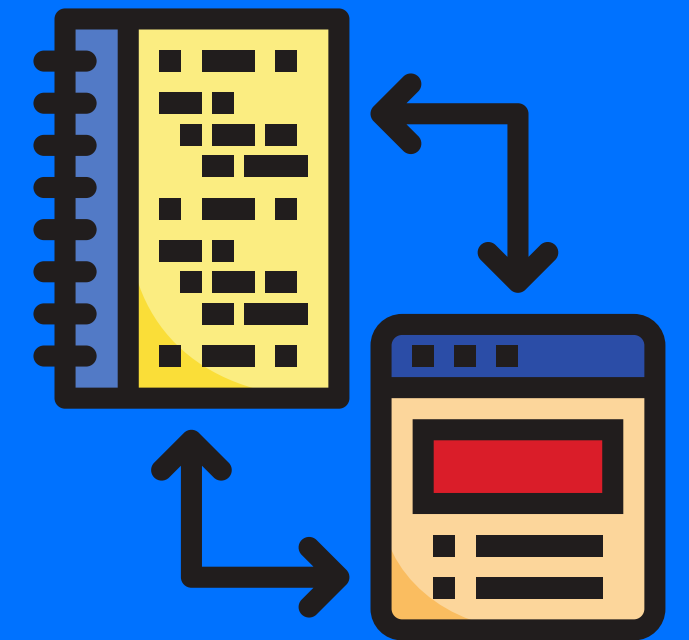
```
int(9)
int(10)
int(-1)
float(-0.9)
bool(true)
```

# Compound Statement/Block Statement

Compound Statement adalah kumpulan dari beberapa statement yang dikelompokkan menjadi satu dengan cara memberi batas tanda kurung kurawal di awal dan tanda kurung kurawal akhir.



```
1  <?php
2
3  {
4      $b = 2;
5      $c = 3;
6      $d = $b + $c;
7  }
```



# Implementasi Compound Statement

```
mardirabootcamp - CompoundStatement.php
1  <?php
2
3  $a = 3;
4  if ($a > 2) {
5      $b = 2;
6      $c = 3;
7      $d = $b + $c;
8  } else {
9      $b = 5;
10     $c = 2;
11     $d = $b - $c;
12 }
13
14 echo "Hasil dari {$b} dan {$c} adalah {$d}";
```

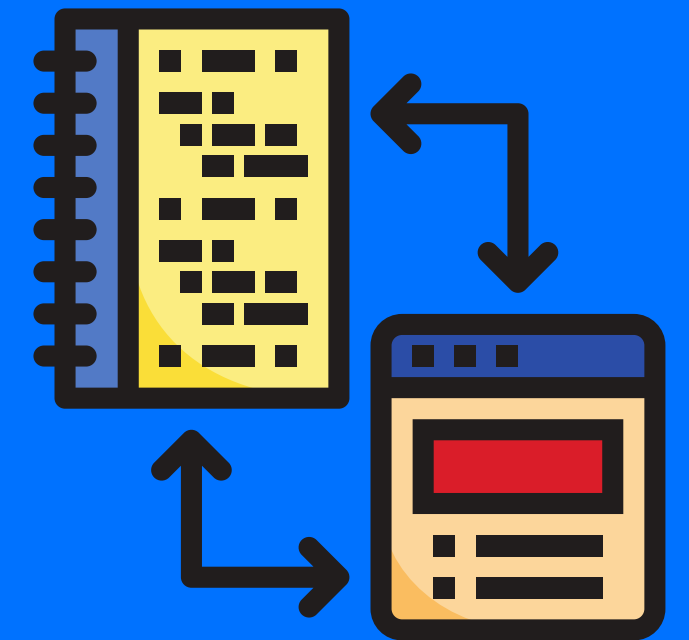
Hasil dari 2 dan 3 adalah 5

Jika a lebih besar dari 2 maka akan menjalankan block statement penjumlahan, dan sebaliknya, jika a lebih kecil dari 2 maka akan menjalankan block statement pengurangan.

# Control Statement

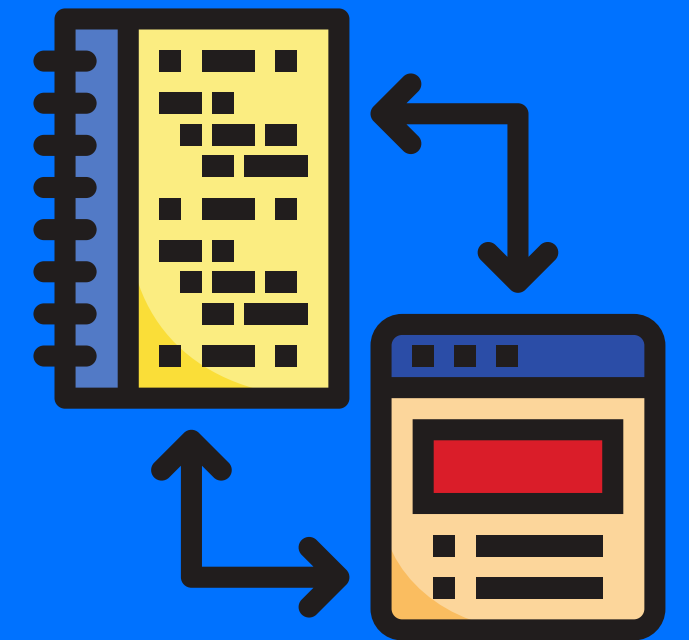
Control statement adalah statement yang digunakan untuk mengarahkan jalannya program, sehingga urutan pengerjaan program dapat melewati tahap dengan kondisi tertentu atau program dapat menjalankan eksekusi dengan secara berulang-ulang.

```
1  <?php
2
3  $a = 15;
4  if ($a > 10) {
5      for ($i = 0; $i < $a; $i++) {
6          echo $i++;
7      }
8  } else {
9      echo $a = 20;
10 }
```



# Jenis Control Statement

- Percabangan (Selection Statement)
- Perulangan (Looping/Iteration Statement)



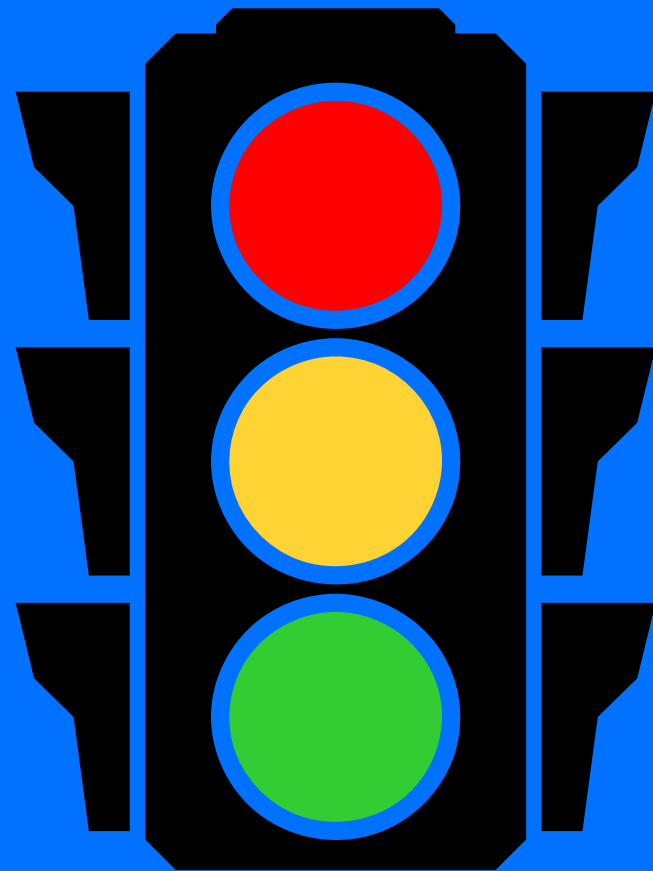
# Percabangan / Selection Statement

Percabangan adalah suatu perintah (pernyataan) yang memungkinkan suatu perintah (pernyataan) dieksekusi jika suatu kondisi terpenuhi atau tidak.

```
1  <?php
2
3  // control statement if
4
5  $a = 3;
6  if ($a == 3) {
7      echo "a sama dengan 3";
8  } else {
9      echo "a tidak sama dengan 3";
10 }
```



# Analogi Percabangan



Ketika melihat rambu lalu lintas di sebuah perempatan, pada saat kondisi lampu yang ditemui berwarna merah, maka kendaraan harus berhenti. Saat kondisi lampu yang menyala berwarna hijau, maka kendaraan masih bisa terus melaju. Bentuk statement seperti ini dinamakan struktur pemilihan (selection) atau percabangan.



# Bentuk Percabangan

- Percabangan Sederhana (Simple Way Case)
- Percabangan Majemuk/Bertingkat (Multi Way Case)
- Percabangan Bersarang (Nested Case)



# Percabangan Sederhana (Simple Way Case)

- Percabangan Tunggal (One Way Case)
- Percabangan Dua Tingkat (Two Way Case)

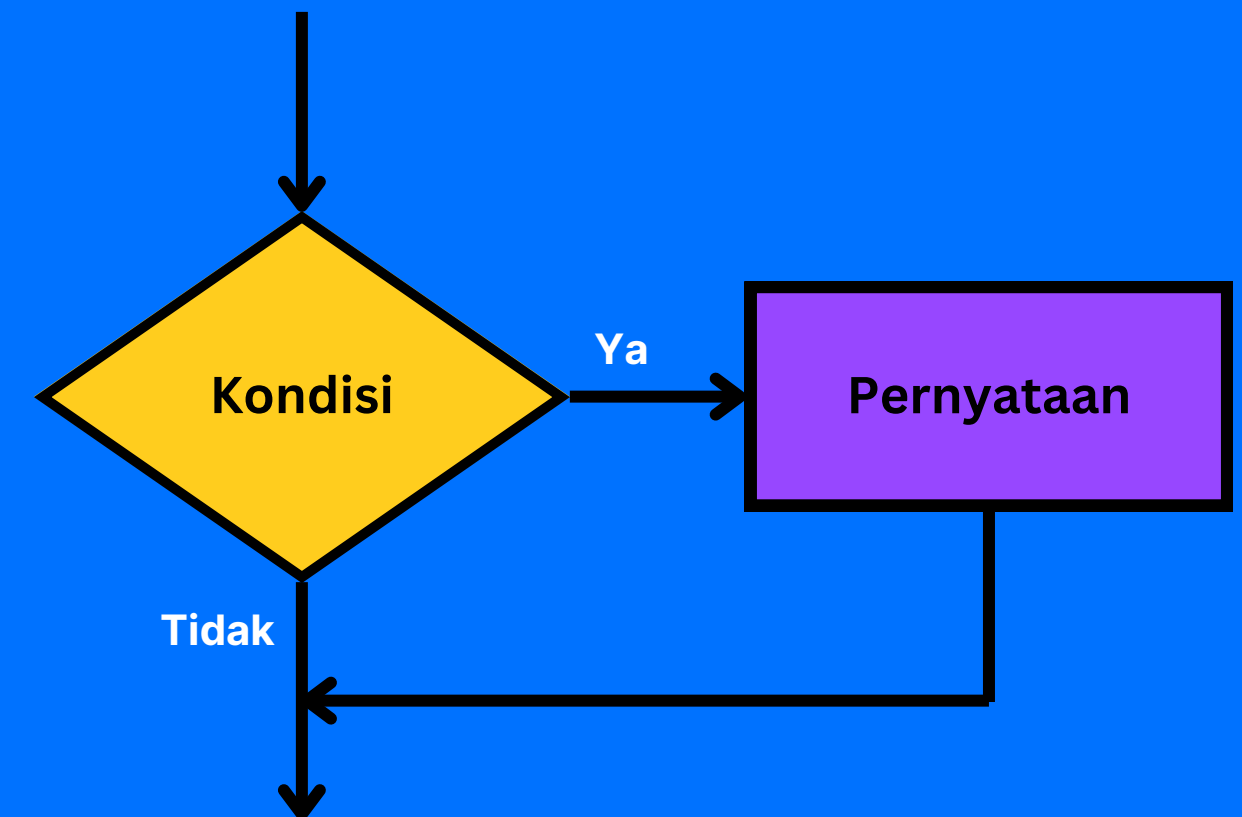


# Percabangan Tunggal / One Way Case

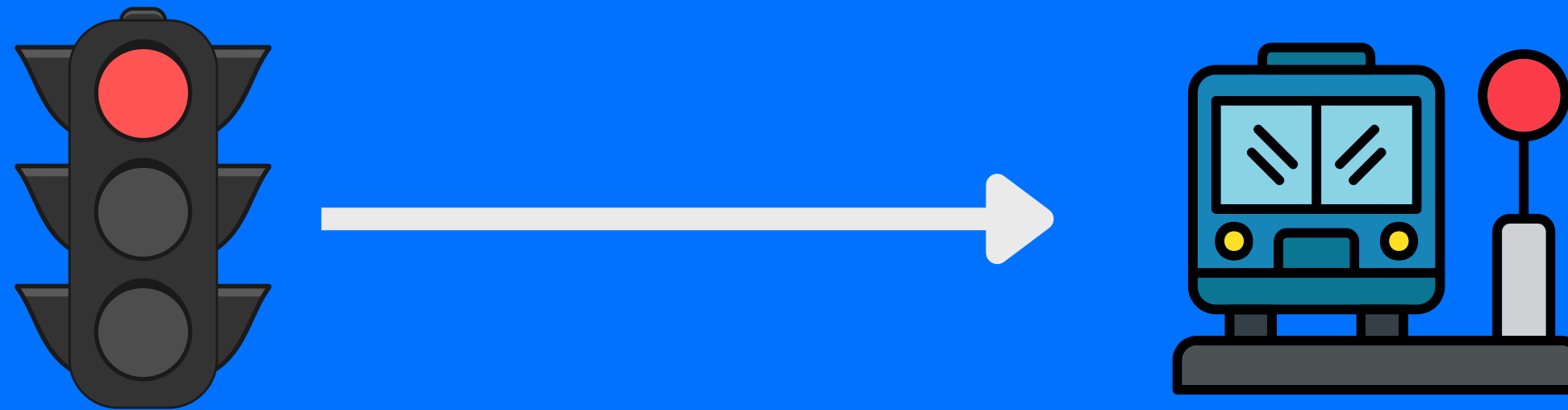
Aturan penulisan struktur percabangan tunggal atau one-way case

- Notasi penulisannya menggunakan if (kondisi) -> aksi
- Kondisi pecabangan untuk menentukan nilai benar atau salahnya pada sebuah statement
- Aksi dari sebuah statement yang di kerjakan hanya jika kondisi bernilai benar, dan sebaliknya

```
if <kondisi> then  
    pernyataan aksi atau instruksi
```

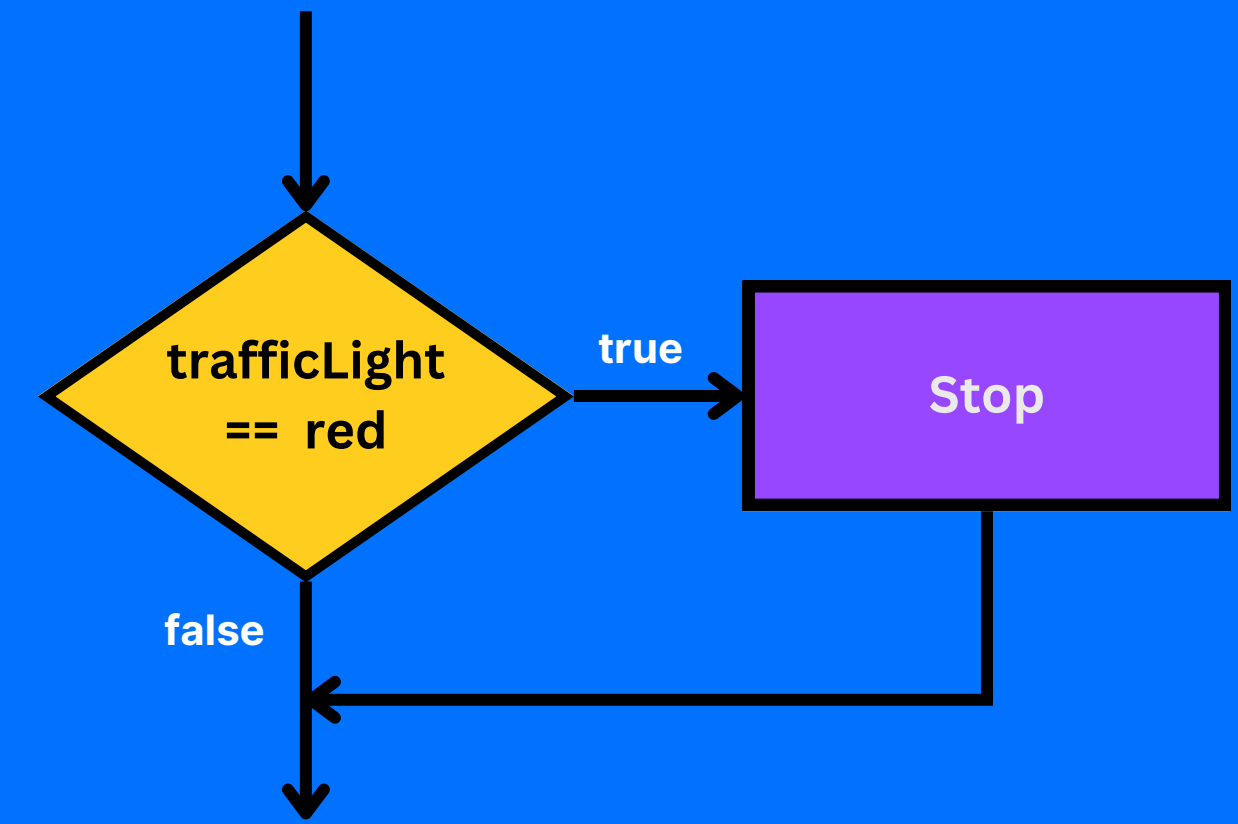


# Contoh Percabangan Tunggal



● ● ● mardirabootcamp - SelectionOneWayCase.php

```
1  <?php
2
3  $trafficLight = "red";
4
5  if ($trafficLight == "red") {
6      echo "Stop";
7  }
```



# Implementasi Percabangan Tunggal

● ● ● mardirabootcamp - SelectionOneWayCase.php

```
1  <?php
2
3  // selection statement one way case
4
5  $number = 10;
6  if ($number > 9) {
7      echo "Nilai lebih dari 9";
8  }
```

Nilai lebih dari 9

# Alternatif Percabangan Tunggal

```
mardirabootcamp - SelectionOneWayAlt.php

1  <?php
2
3  $isBoolean = true;
4
5  // cek nilai benar atau salah
6  if ($isBoolean = true) {
7      echo "Benar"; #Benar
8  }
9
10 if ($isBoolean = false) {
11     echo "Salah"; #output kosong
12 }
13
14 // shorthand cek nilai benar atau salah
15 if ($isBoolean) {
16     echo "Benar"; #Benar
17 }
18
19 if (!$isBoolean) {
20     echo "Salah"; #output kosong
21 }
```

Kondisi percabangan tunggal menggunakan operator perbandingan sama dengan

BenarBenar

# Contoh Percabangan Tunggal Lainnya

```
mardirabootcamp - SelectionOneWayAlt.php

1  <?php
2
3  // Kondisi menggunakan operator logika AND
4  $waterTemperature = 10;
5
6  if ($waterTemperature > 0 && $waterTemperature < 30) {
7      echo "Hasil Suhur Air : Dingin<br>"; #TRUE && TRUE
8  }
9
10 // Kondisi menggunakan operator logika OR
11 $trafficLight = "red";
12
13 if ($trafficLight = "red" || $trafficLight = "yellow") {
14     echo "Rambu Lalu Lintas : Berhenti <br>"; #TRUE || FALSE
15 }
16
17 // Kondisi menggunakan operator logika NOT
18 $waterTemperature = 10;
19
20 if (!($waterTemperature > 0 && $waterTemperature < 30)) {
21     echo "Hasil Suhur Air : Dingin"; #FALSE && TRUE
22 }
23
24 // Kondisi menggunakan operator logika XOR
25 $trafficLight = "red";
26
27 if ($trafficLight = "red" xor $trafficLight = "red") {
28     echo "Rambu Lalu Lintas : Berhenti <br>"; #TRUE XOR TRUE
29 }
```

Kondisi percabangan tunggal menggunakan operator logika

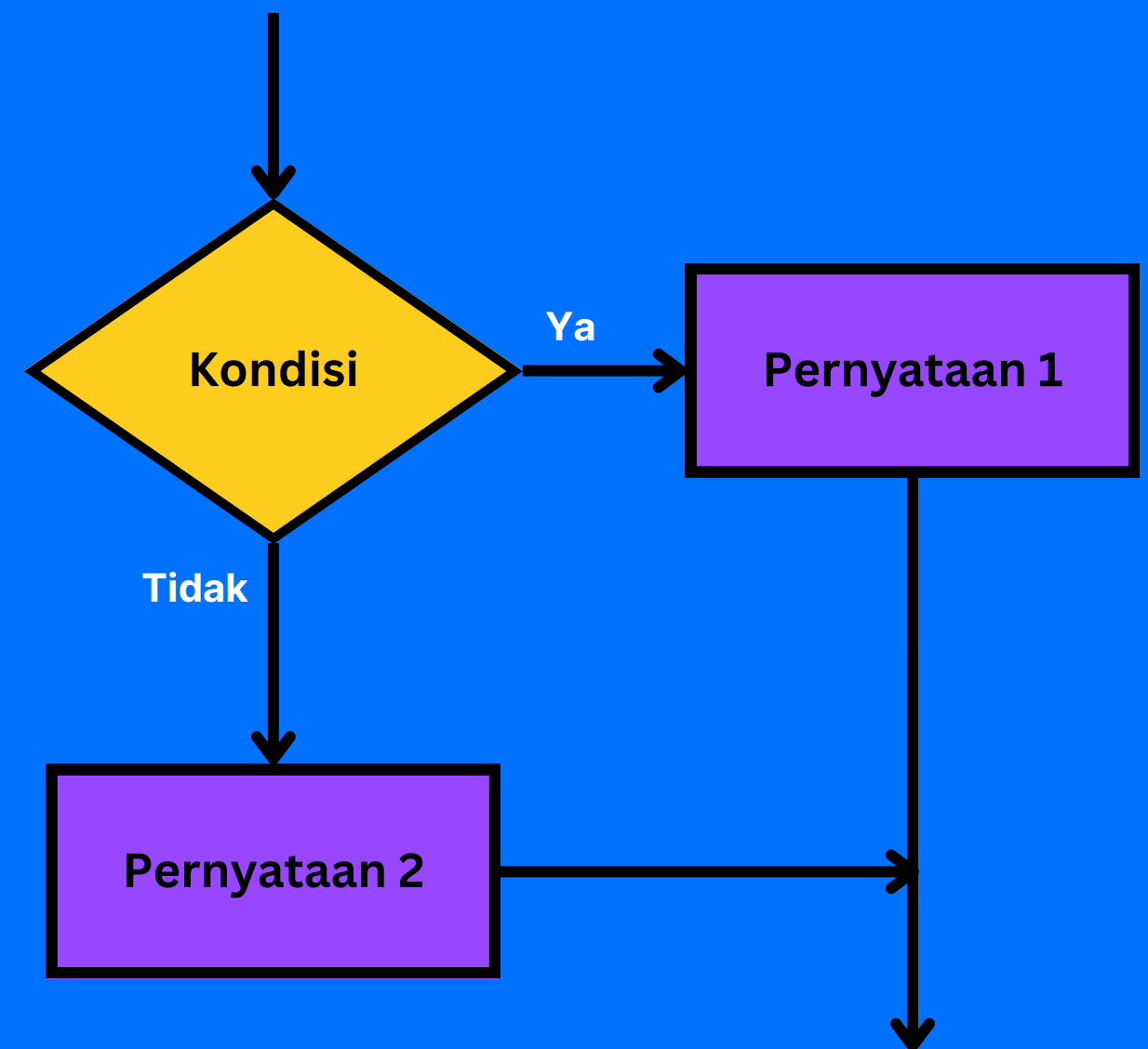
Hasil Suhur Air : Dingin  
Rambu Lalu Lintas : Berhenti

# Percabangan Dua Tingkat/ Two Way Case

Aturan penulisan struktur percabangan dua tingkat atau two-way case

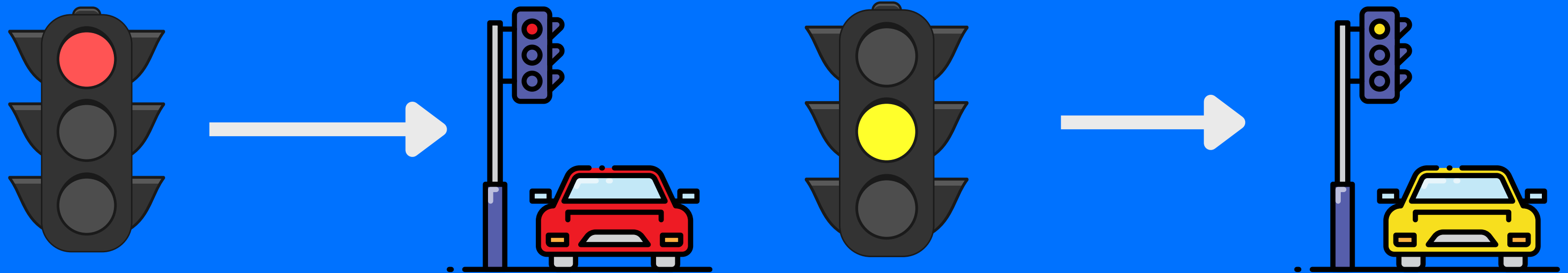
- Notasi penulisannya menggunakan if (kondisi) -> pernyataan aksi 1 -> else -> pernyataan aksi 2
- “else” artinya “jika tidak” (menyatakan negasi dari kondisi)
- Memilih/mejalankan satu dari dua buah pernyataan aksi sesuai dengan nilai kondisi yang dipenuhi
- Jika kondisi terpenuhi/ bernilai benar maka pernyataan aksi 1 akan dikerjakan, tetapi jika tidak, maka pernyataan aksi 2 yang akan dikerjakan

```
if <kondisi> then  
    pernyataan aksi 1  
else  
    pernyataan aksi 2
```



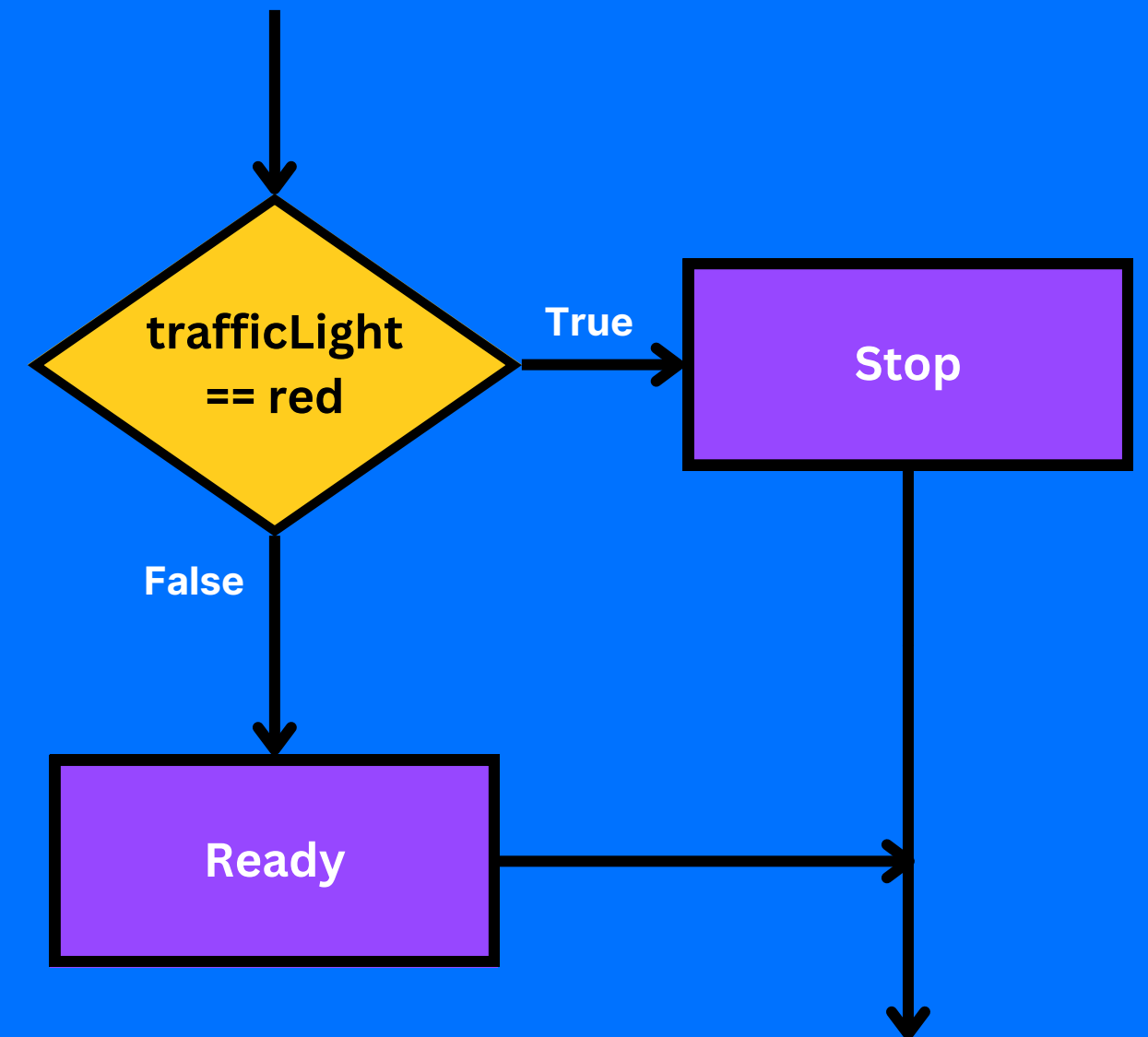


# Contoh Percabangan Dua Tingkat



● ● ● mardirabootcamp - SelectionTwoWayCase.php

```
1  <?php
2
3  $trafficLight = "red";
4
5  if ($trafficLight == "red") {
6      echo "Stop";
7  } else {
8      echo "Ready";
9  }
```



# Implementasi Percabangan Dua Tingkat

```
mardirabootcamp - SelectionTwoWayCase.php

1  <?php
2
3  $waterTemperature = 10;
4
5  if ($waterTemperature > 0 && $waterTemperature < 30) {
6      echo "Hasil Suhur Air : Dingin<br>"; #TRUE && TRUE
7  } else {
8      echo "Hasil Suhur Air : Panas<br>"; #FALSE && TRUE
9  }
10
11  // Kondisi Terbalik
12  if ($waterTemperature > 0 && $waterTemperature > 30) {
13      echo "Hasil Suhur Air : Panas<br>"; #TRUE && FALSE
14  } else {
15      echo "Hasil Suhur Air : Dingin<br>"; #FALSE && FALSE
16  }
```

Hasil Suhur Air : Dingin  
Hasil Suhur Air : Dingin

# Alternatif Percabangan Dua Tingkat

```
mardirabootcamp - SelectionTwoWayCaseAlt.php

1  <?php
2
3  $waterTemperature = 10;
4
5
6  if ($waterTemperature > 0 && $waterTemperature < 30 && $waterTemperature ≠ NULL) {
7      echo "Hasil Suhur Air : Dingin<br>"; #TRUE && TRUE && TRUE
8  }
9
10 // Kondisi menggunakan operator logika OR
11
12 $trafficLight = "red";
13
14 if ($trafficLight = "red" || $trafficLight = "yellow" || $trafficLight = "green") {
15     echo "Rambu Lalu Lintas : Berhenti <br>"; #TRUE || FALSE || FALSE
16 }
```

Hasil Suhur Air : Dingin  
Rambu Lalu Lintas : Berhenti

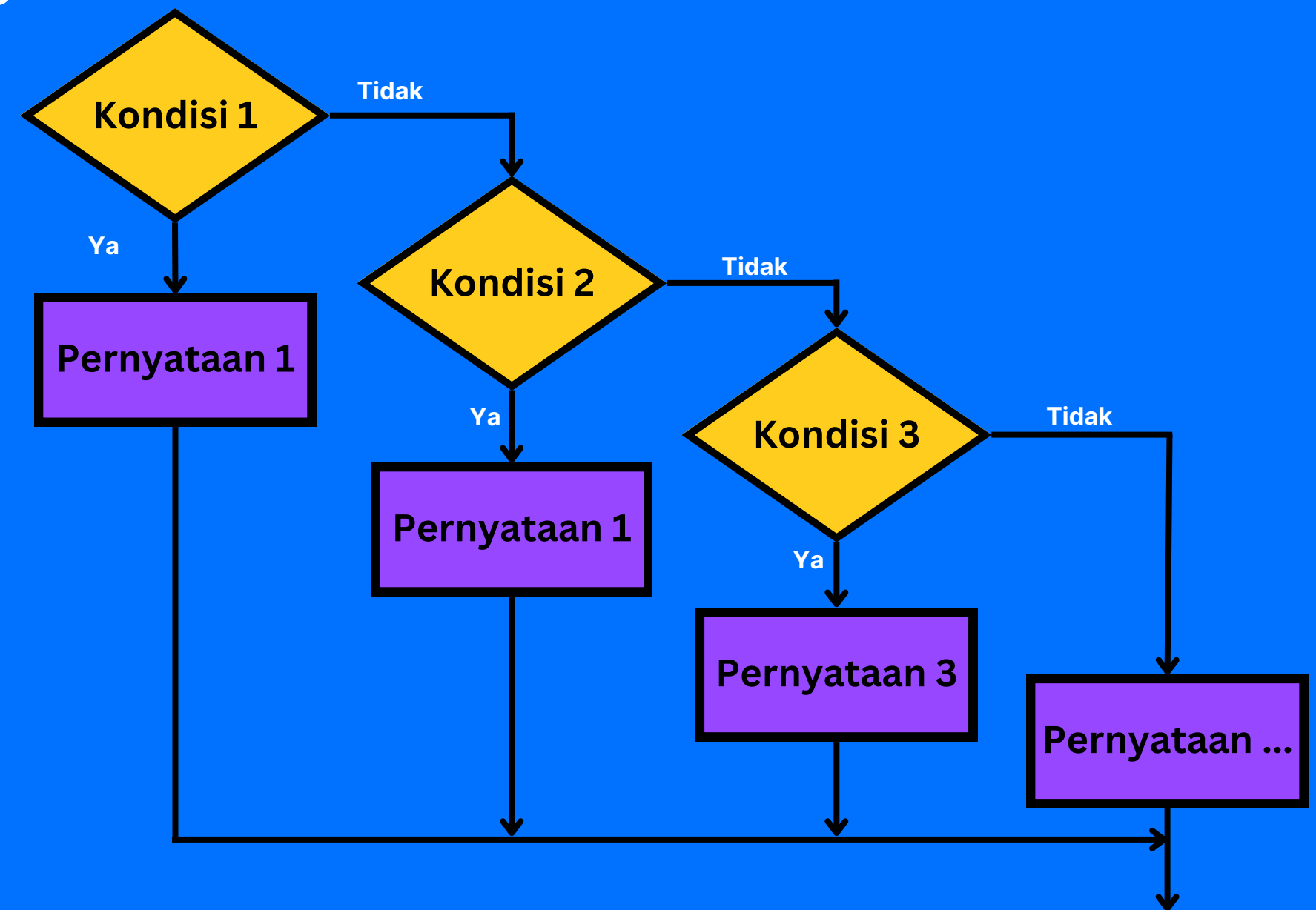
Percabangan dua tingkat menggunakan lebih dari 2 operator logika

# Percabangan Majemuk/Multi Way Case

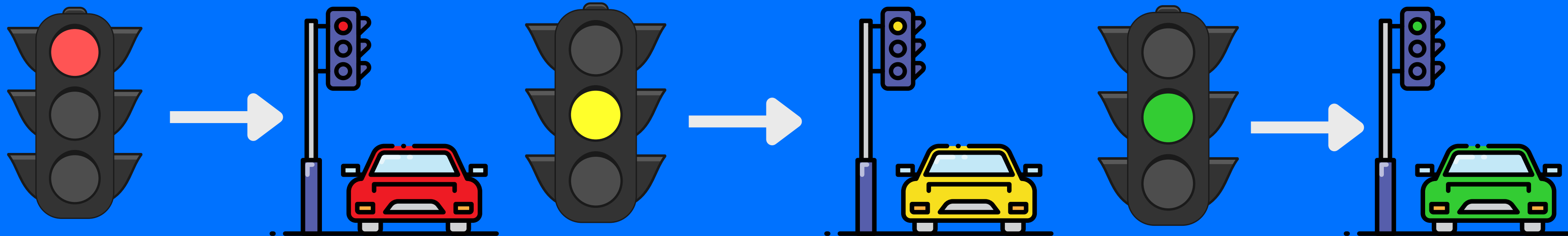
Aturan penulisan struktur percabangan majemuk atau multi-way-case

- Notasi penulisannya menggunakan if (kondisi) -> pernyataan aksi 1 -> elseif -> pernyataan aksi 2 -> seterusnya .. kemudian diakhiri else -> pernyataan aksi terakhir
- Struktur percabangan bertingkat digunakan bila ada dua buah kondisi atau lebih yang dapat dianalisis

```
if <kondisi 1> then  
    pernyataan aksi 1  
elseif <kondisi 2>  
    pernyataan aksi 2  
    . . .  
    . . .  
else  
    pernyataan aksi 3
```

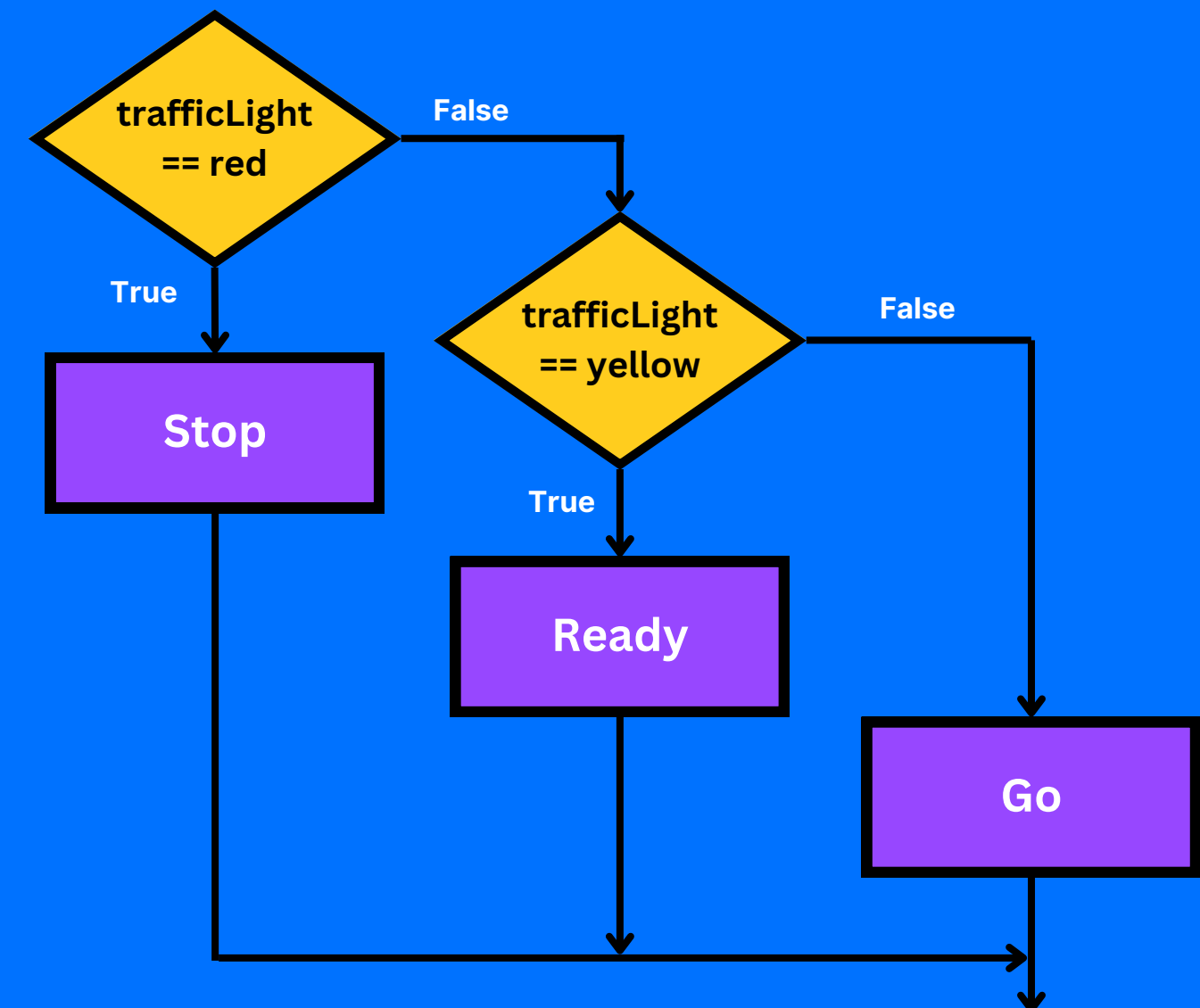


# Contoh Percabangan Majemuk



● ● ● mardirabootcamp - SelectionMultiWayCase.php

```
1  <?php
2
3  $trafficLight = 'red';
4
5  if ($trafficLight == 'red') {
6      echo 'Stop';
7  } elseif ($trafficLight == 'yellow') {
8      echo 'Ready';
9  } else {
10     echo 'Go';
11 }
```



# Implementasi Percabangan Majemuk

```
mardirabootcamp - SelectionMultiWayCase.php

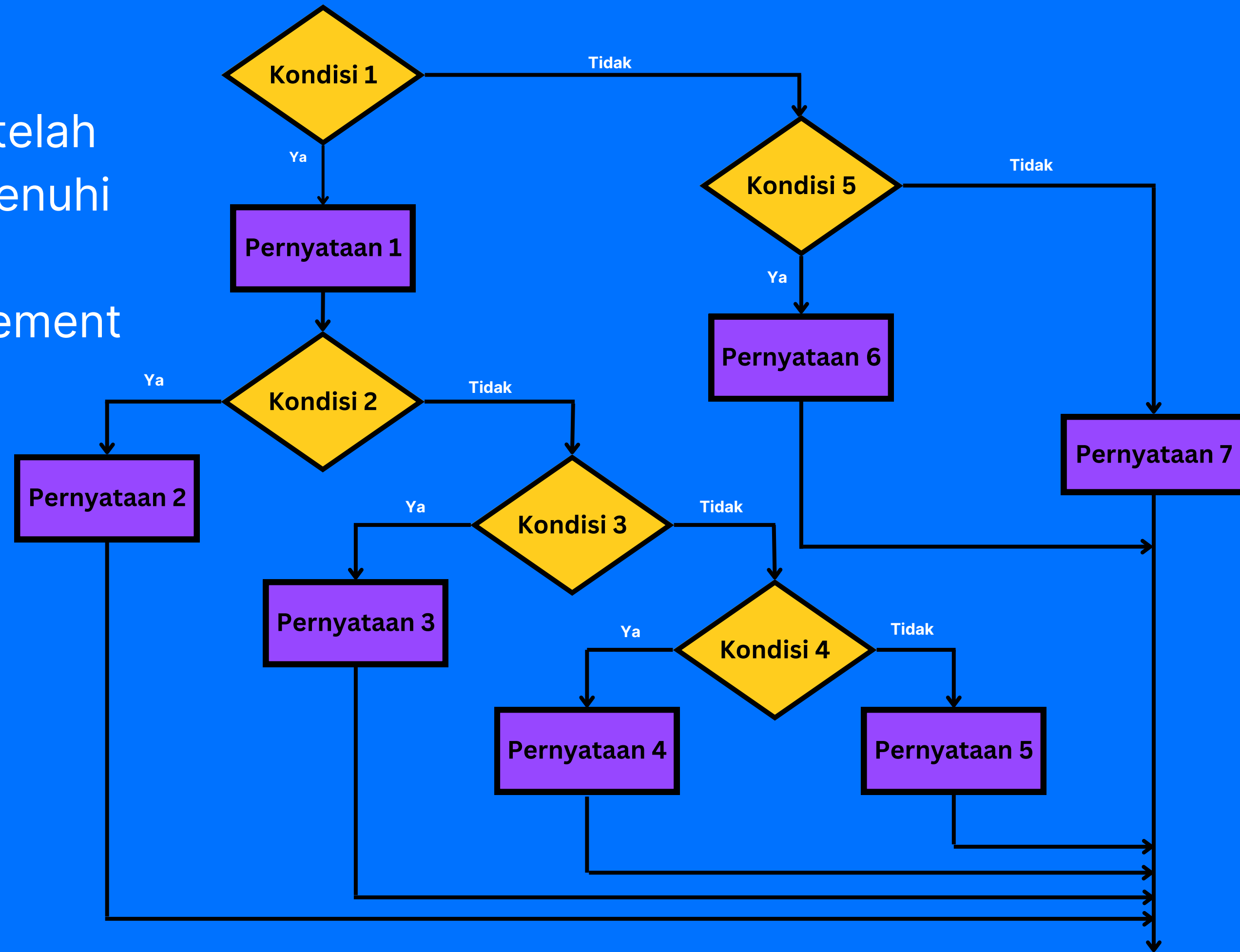
1  <?php
2
3  $waterTemperature = 10;
4
5  if ($waterTemperature > 0 && $waterTemperature < 30 && $waterTemperature ≠ NULL) {
6      echo "Hasil Suhur Air : Dingin<br>"; #TRUE && TRUE && TRUE
7  } elseif ($waterTemperature > 30 && $waterTemperature < 50 && $waterTemperature ≠ NULL) {
8      echo "Hasil Suhur Air : Hangat<br>"; #FALSE && TRUE && TRUE
9  } elseif ($waterTemperature > 50 && $waterTemperature < 100 && $waterTemperature ≠ NULL) {
10     echo "Hasil Suhur Air : Panas<br>"; #FALSE && TRUE && TRUE
11 } else {
12     echo "Hasil Suhur Air : Tidak Diketahui<br>"; #FALSE
13 }
14
15 // Kondisi menggunakan operator logika OR
16
17 $trafficLight = "red";
18
19 if ($trafficLight = "red" || $trafficLight = "yellow" || $trafficLight = "green") {
20     echo "Rambu Lalu Lintas : Berhenti <br>"; #TRUE || FALSE || FALSE
21 } elseif ($trafficLight = "yellow" || $trafficLight = "green" || $trafficLight = "red") {
22     echo "Rambu Lalu Lintas : Hati-hati <br>"; #FALSE || FALSE || TRUE
23 } elseif ($trafficLight = "green" || $trafficLight = "red" || $trafficLight = "yellow") {
24     echo "Rambu Lalu Lintas : Maju <br>"; #FALSE || TRUE || FALSE
25 } else {
26     echo "Rambu Lalu Lintas : Tidak Diketahui <br>"; # FALSE
27 }
```

Hasil Suhur Air : Dingin  
Rambu Lalu Lintas : Berhenti

# Percabangan Bersarang / Nested Case

- Situasi dimana kita ingin memeriksa kondisi lain setelah kondisi sebelumnya terpenuhi (bernilai TRUE).
- Konsep: terdapat IF statement di dalam IF statement

```
if <kondisi 1>  
  Pernyataan aksi 1  
  if <kondisi 2>  
    Pernyataan aksi 2  
  elseif <kondisi 3>  
    Pernyataan aksi 3  
  elseif <kondisi 4>  
    Pernyataan aksi 4  
  else  
    Pernyataan aksi 5  
elseif <kondisi 5>  
  Pernyataan aksi 6  
else  
  Pernyataan aksi 7
```





# Latihan Percabangan Bersarang



Buatlah program sederhana dengan percabangan berikut ini



# Jenis Percabangan

- If Else
- Switch Case
- Ternary Operator
- Null Coalescing




# Percabangan IF

Percabangan if ada 3 jenis penulisan, yang pertama menggunakan kurung kurawal sebagai pembatas (block statement), yang kedua tidak menggunakan kurung kurawal, dan yang terakhir menggunakan tag dan penutup tag php dan di akhiri dengan endif sebagai pembatas pada selection statement.

 Disarankan

```
1 <?php
2
3 $trafficLight = "red";
4
5 if ($trafficLight = "red") {
6     echo "Berhenti";
7 }
```

 Alternatif

```
1 <?php
2
3 $trafficLight = "red";
4
5 if ($trafficLight = "red")
6     echo "Berhenti";
```

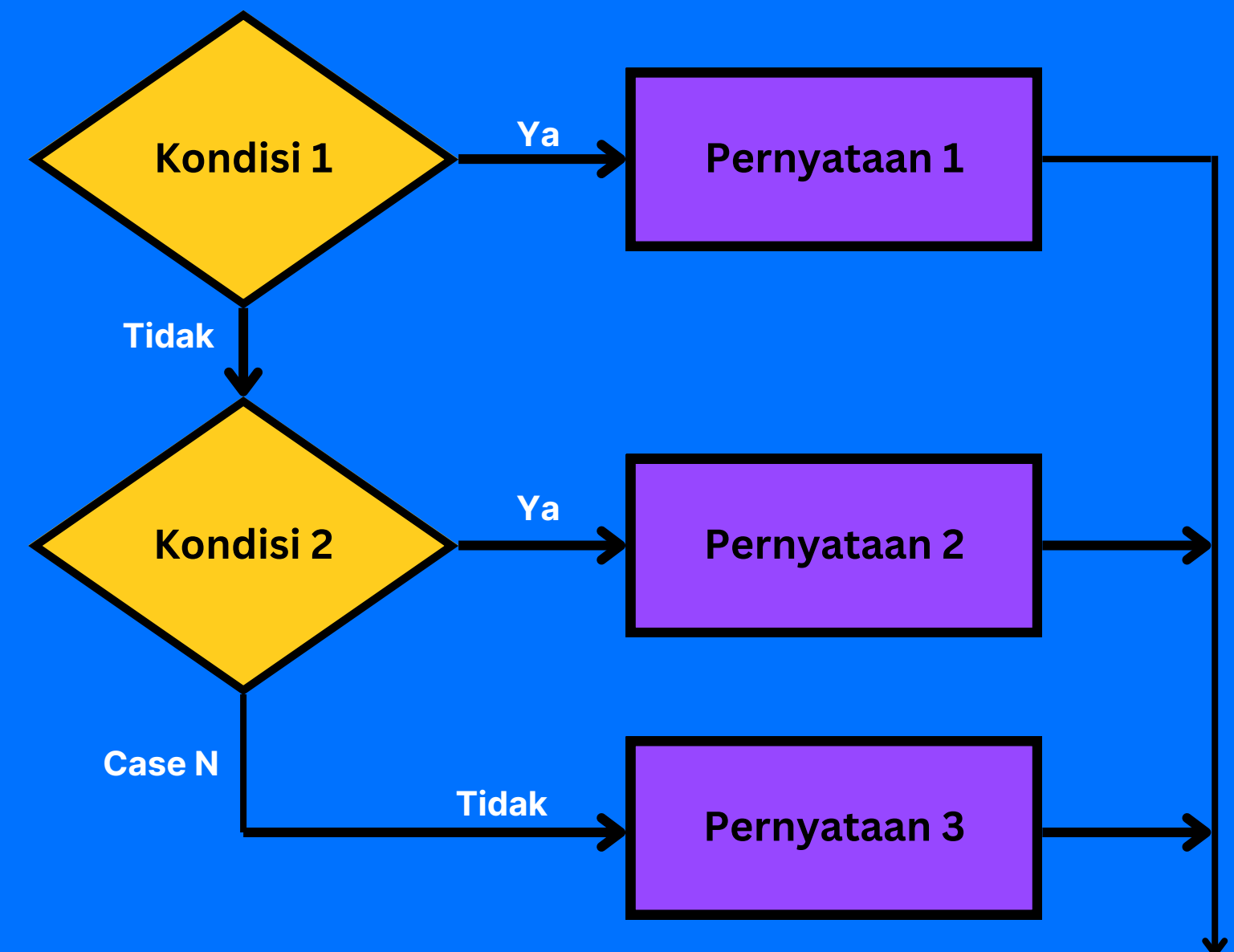
 Alternatif

```
1 <?php
2
3 $trafficLight = "red";
4
5 if ($trafficLight = "red") : ?>
6     <p>Berhenti</p>
7 <?php endif; ?>
```

# Switch Case

Switch case merupakan statement percabangan selain if else. Statement switch case akan mencocokkan suatu nilai terhadap serangkaian nilai yang menjadi kondisi sampai menemukan kecocokan tersebut

```
switch <nilai>  
    case <kondisi 1>  
        Pernyataan Aksi 1  
        break;  
    case <kondisi 2>  
        Pernyataan Aksi 2;  
        break;  
    default  
        Pernyataan Aksi 3;  
        break
```



# Implementasi Switch Case

mardirabootcamp - SwitchCase.php

```
1  <?php
2
3  $marks = 50;
4
5  switch ($marks) {
6      case ($marks ≥ 85):
7          echo "Indeks = A";
8          break;
9      case ($marks ≥ 75):
10         echo "Indeks = B";
11         break;
12     case ($marks ≥ 65):
13         echo "Indeks = C";
14         break;
15     case ($marks ≥ 55):
16         echo "Indeks = D";
17         break;
18     case ($marks ≥ 45):
19         echo "Indeks = E";
20         break;
21     default:
22         echo "Indeks = F";
23         break;
24 }
```

Indeks = E

# Alternatif Switch Case

```
mardirabootcamp - SwitchCaseAlt.php
1  <?php
2
3  $x = 5;
4  switch ($x) {
5      case 1:
6          echo "x bernilai 1";
7          break;
8      case 2:
9          echo "x bernilai 2";
10         break;
11     case 3:
12         switch ($x) {
13             case 3:
14                 echo "x bernilai 3";
15                 break;
16             case 4:
17                 echo "x bernilai 4";
18                 break;
19             default:
20                 echo "x tidak bernilai 3 atau 4";
21                 break;
22         }
23         break;
24     default:
25         echo "x tidak bernilai 1";
26         break;
27 }
```

Switch Case bisa menggunakan metode nested case.

x tidak bernilai 1

# Perbedaan IF dan Switch

Perbandingan	If Else	Switch
Dasar	Statement yang akan dieksekusi tergantung pada output dari ekspresi di dalam pernyataan if.	Statement yang akan dieksekusi ditentukan oleh pengguna.
Ekspresi	Statement if-else menggunakan banyak pernyataan untuk beberapa pilihan	Statement switch menggunakan ekspresi tunggal untuk beberapa pilihan.
Pengujian	Statement if-else mampu menggunakan ekspresi perbandingan dan ekspresi logika.	Statement switch pengujiannya hanya untuk ekspresi perbandingan.
Output	Output If-else dapat berupa integer,float, string, array, dan boolean.	Statement switch hanya menghasilkan output berupa string atau integer.
Urutan Eksekusi	Statement if-else akan mengeksekusi jika ada kondisi yang sesuai dengan permintaan, jika tidak maka akan berjalan kondisi yang lainnya	Statement switch mengeksekusi satu kasus demi kasus sampai break statement muncul atau statement akhir tercapai.

# Ternary Operator

Ternary Operator merupakan cara lain menulis suatu perkondisian dengan teknik penulisan yang lebih sederhana. Operator ini diwakilkan dengan tanda "?" dan tanda ":"



```
1  <?php
2
3  $a = 1;
4
5  $a = $a ? 'Angka 1' : 'Bukan Angka 1';
6
```

# implementasi Ternary Operator

mardirabootcamp - TernaryOperator.php

```
1  <?php
2
3  $marks = 80;
4
5  echo $marks > 80 ? "Nilai Anda A" : "Nilai Anda B";
```


Nilai Anda B



# Perbandingan Ternary Operator Dengan If-Else



```
1  <?php
2
3  // if else
4
5  $mark = 80;
6
7  if ($mark > 80) {
8      echo "Nilai Anda A";
9  } else {
10     echo "Nilai Anda B";
11 }
12
```



```
1  <?php
2
3  // ternary operator
4
5  $marks = 80;
6
7  echo $marks > 80 ? "Nilai Anda A" : "Nilai Anda B";
8
```

# Null Coalescing

```
mardirabootcamp - AssignmentCoalescing.php

1  <?php
2
3  $string = null;
4
5  // Cara biasa mengecek nilai null dengan statement
6  if (isset($string)) {
7      $string = 'hi';
8  } else {
9      $string = 'Hello World';
10 }
11
12 // Null Coalescing standar
13 $string = $string ?? 'Hello World';
14
15 // Null Coalescing dengan operator penugasan
16 $string ??= 'Hello World';
17
18 echo "Hasil : {$string}";
```

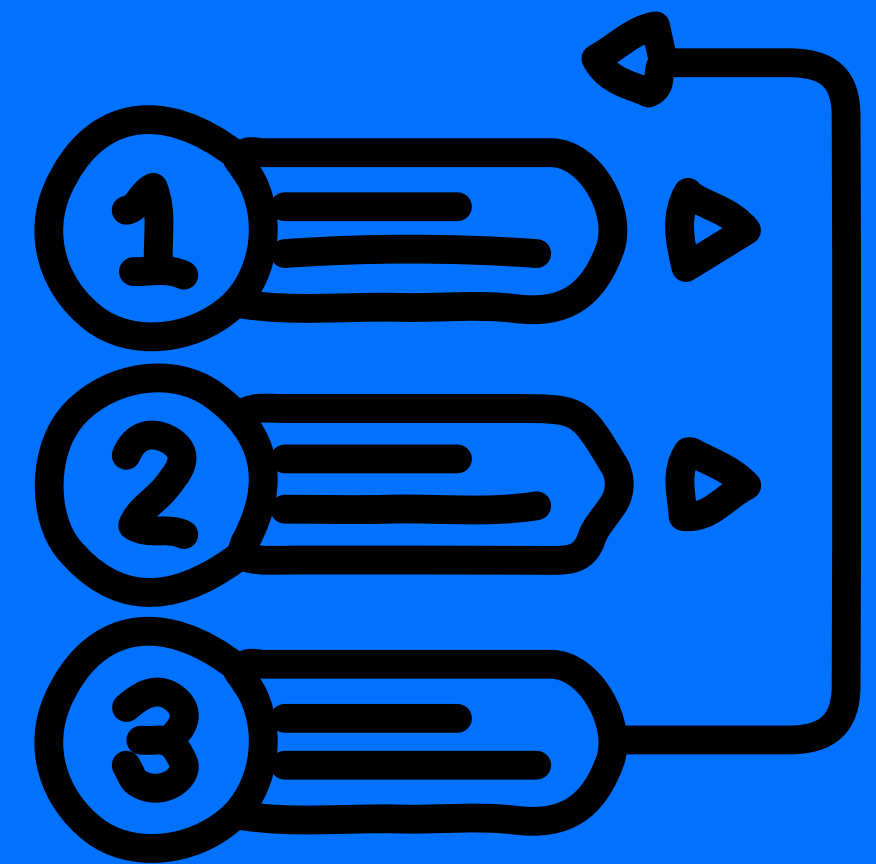
Hasil : Hello World

# Perulangan / Looping / Iteration Statement

Perulangan adalah sebuah kondisi dimana satu atau beberapa baris kode program di eksekusi secara berulang-ulang.

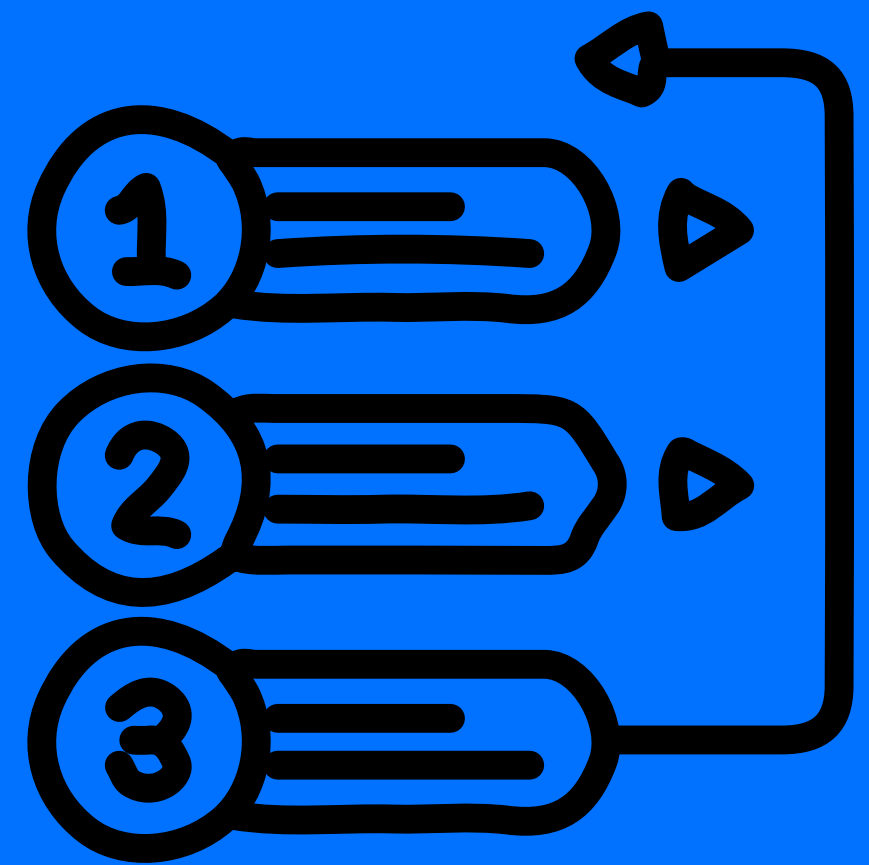


```
1  <?php
2
3  for ($i = 0; $i < 10; $i++) {
4      echo $i;
5  }
```



# Jenis Perulangan

- While Loop
- Do While
- For Loop
- Foreach



# While Loop

While digunakan untuk mengeksekusi kode program secara berulang-ulang selama kondisi yang ditentukan benar.



```
1  while (condition is true) {  
2      code to be executed;  
3  }
```

# Implementasi While Loop

```
mardirabootcamp - WhileStatement.php

1  <?php
2  $x = 1;
3  while ($x ≤ 5) {
4      echo "Belajar Back-End Development Menggunakan PHP <br>";
5      $x++;
6  }
```

Belajar Back-End Development Menggunakan PHP  
Belajar Back-End Development Menggunakan PHP  
Belajar Back-End Development Menggunakan PHP  
Belajar Back-End Development Menggunakan PHP  
Belajar Back-End Development Menggunakan PHP

# Do While Loop

Perintah do...while bekerja dengan mengeksekusi blok kode terlebih dulu, kemudian memeriksa kondisi, dan menjalankan blok kode ketika kondisi yang ditentukan benar



```
1  do {  
2      code to be executed;  
3  } while (condition is true);
```

# Implementasi Do While Loop

● ● ● mardirabootcamp - DoWhile.php

```
1  <?php
2  $x = 1;
3  do {
4      echo "$x <br>";
5      $x++;
6  } while ($x ≤ 5);
```

1  
2  
3  
4  
5



# For Loop

Struktur perulangan ini akan mengulang-ulang eksekusi beberapa baris kode program di dalamnya berdasarkan perubahan isi sebuah variabel. Jika isi variabel yang di acu sudah sampai titik akhir, maka perulangan akan berhenti.



```
1  for (init counter; test counter; increment counter) {  
2      code to be executed for each iteration;  
3  }
```

# Implementasi For Loop

- **init counter:** Inisialisasi nilai penghitung loop kondisi
- **test counter:** Dievaluasi untuk setiap iterasi loop. Jika bernilai TRUE, loop berlanjut. Jika bernilai FALSE, loop berakhir.
- **increment counter:** Meningkatkan nilai penghitung loop

● ● ● mardirabootcamp - ForLoop.php

```
1  <?php
2
3  for ($x = 0; $x ≤ 10; $x++) {
4      echo "$x <br>";
5  }
```

# Foreach

Foreach Loop digunakan untuk menampilkan nilai array, dan perulangan yang ditampilkan sesuai dengan nomor indeks / nilai dalam array.



```
1  foreach ($array as $value) {  
2      code to be executed;  
3  }
```

# Implementasi Foreach Loop



```
1  <?php
2
3  $colors = ["red", "green", "blue", "yellow"];
4
5  foreach ($colors as $value) {
6      echo "$value <br>";
7  }
```

red

green

blue

yellow

# Alternatif Foreach Loop

Cara lain menggunakan foreach dengan memanggil key dari array dan value

```
mardirabootcamp - ForachAlt.php
1  <?php
2
3  $age = ["Peter" => "35", "Ben" => "37", "Joe" => "43"];
4
5  foreach ($age as $x => $val) {
6      echo "$x = $val<br>";
7  }
```

Peter = 35  
Ben = 37  
Joe = 43

# Perbedaan Statement dan Expression

The diagram illustrates the difference between a Statement and an Expression using PHP code. The code is shown in a dark-themed editor with line numbers 1 through 12. The code is as follows:

```
1 <?php
2
3 // if else
4
5 $mark = 80;
6
7 if ($mark > 80) {
8     echo "Nilai Anda A";
9 } else {
10     echo "Nilai Anda B";
11 }
12
```

Annotations and boxes highlight specific parts of the code:

- Expression:** A green arrow points to the expression `$mark > 80` inside the `if` statement on line 7.
- Control Statement:** A white arrow points to the `if` statement structure on line 7.
- Compound Statement:** A red arrow points to the entire `if-else` block, which is enclosed in a red box.
- Expression Statement:** A yellow arrow points to the statement `echo "Nilai Anda B";` on line 10, which is enclosed in a yellow box.

# Function

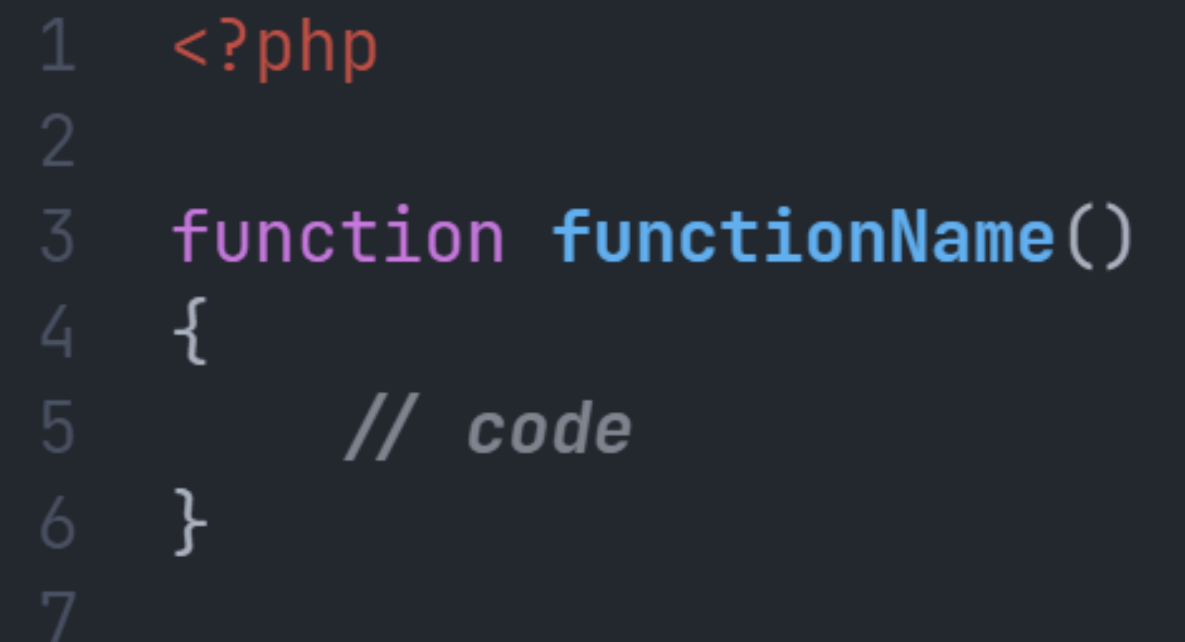
Function atau fungsi atau dalam beberapa bahasa lain dipanggil method merupakan blok kode yang dapat digunakan berulang kali dalam pemrograman. Function umumnya berisi kode perintah untuk menyelesaikan tugas tertentu.



# Penulisan Function

Dari sintaks kode di atas kita bisa simpulkan untuk membuat function umumnya membutuhkan:


- keyword bernama function
- nama function-nya
- tanda kurung () untuk menampung parameter
- kurung kurawal untuk membuat blok kode



```
1  <?php
2
3  function functionName()
4  {
5      // code
6  }
7
```



# Penamaan Function



```
1  <?php
2
3  // camel case
4  function functionName()
5  {
6      // code
7  }
8
9  // snake case
10
11 function function_name()
12 {
13     //
14 }
```

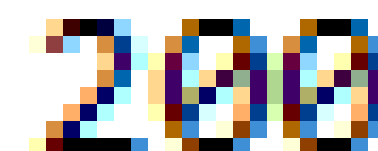
Perlu diperhatikan, khusus untuk memberi nama function ada beberapa aturan yang harus dipatuhi seperti:

- nama function boleh mengandung angka [0-9], huruf [A-Z] [a-z], atau tanda underscore. Selain di atas akan menyebabkan error.
- namun untuk awalan nama function harus menggunakan huruf atau tanda underscore, selain itu seperti angka atau tanda lain tidak diizinkan.
- sifat dari nama function adalah case-insensitive atau huruf kecil dan besar akan dianggap sama.

# Implementasi Function

● ● ● mardirabootcamp - function.php

```
1  <?php
2
3  function sum()
4  {
5      echo 100 + 100;
6  }
7
8  sum();
```

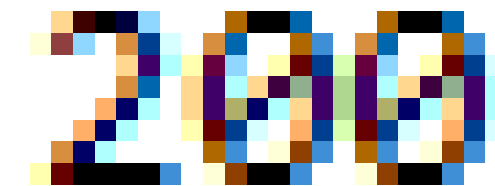
A pixelated, multi-colored representation of the number 200, displayed on a white background within a window-like frame.

# Alternatif Function

Function bisa menggunakan parameter berupa variable sebagai penampung nilai pada fungsi yang akan dijalankan

● ● ● mardirabootcamp - function.php

```
1  <?php
2
3  function sum($a, $b)
4  {
5      echo $a + $b;
6  }
7
8  sum(100, 100);
```



**Terimakasih**