**BAB III**

**PENGKODISIAN DAN PERULANGAN**

1. Tujuan
2. Dapat memahami pengertian dan konsep pengkondisian dan perulangan.
3. Dapat mengetahui penggunaan pengkondisian dengan bahasa pemograman C#, Java dan PHP.
4. Dapat mengimplementasikan pengkondisian dalam bahasa pemograman C#, Java dan PHP.
5. Dapat mengetahui perbedaan pengkondisian antara *if, if - else, if- else if - else* dan *switch case*.
6. Dapat memahami konsep dasar perulangan.
7. Dapat memahami peryataan perulangan *WHILE.*
8. Dapat memahami peryataan perulangan *DO - WHILE.*
9. Dapat memahami peryataan perulangan *FOR.*
10. Dapat membuat program sederhana pengkondisian menggunakan bahasa pemograman C#, Java dan PHP.
11. Dapat membuat program sederhana perulangan dengan menggunakan bahasa C#, Java dan PHP.
12. Dasar Teori
13. **Pengkodisian**
14. **If**

Fungsi *IF* dalam bahasa pemograman bahasa untuk menjalankan ( mengeksekusi suatu program) yang apabila syaratnya atau syarat tertentu telah terpenuhi yang mana bisa terdiri hanya atas satu keadaan (syarat) atau  banyak keadaan tergantung dari program tersebut.

(sumber: <http://kucing-gilak.blogspot.com/2017/01/belajar-fungsi-if-dalam-bahasa.html>)

1. **If.. Else**

Fungsi *IF Else* dalam bahasa pemograman adalah suatu fungsi yang mengeksekusi atau menjalan suatu program apa bisa suatu kondisi atau persyaratan terpenuhi maka program tersebut atau perintah tersebut di jalankan, tetapi apabila kondisi atau syaratnya tidak terpenuhi maka disinilah fungsi dari *else* tersebut. Yang mana *else* bisa digunakan sebagai kondisi lain apabila kondisi pertama tidak terpenuhi dan juga bisa dibuata *else* kedua yang mana akan di jalankan bisa *else* pertama tidak terpenuhi.

(sumber: <http://kucing-gilak.blogspot.com/2017/01/belajar-fungsi-if-else-dan-contohnya.html>)

1. **If.. Else If.. Else**

Pada dasarnya, kondisi *IF ELSE IF* adalah sebuah struktur logika program yang di dapat dengan cara menyambung beberapa kondisi *IF ELSE* menjadi sebuah kesatuan.

Jika kondisi pertama tidak terpenuhi atau bernilai *false*, maka kode program akan lanjut ke kondisi *IF* di bawahnya. Jika ternyata tidak juga terpenuhi, akan lanjut lagi ke kondisi *IF* di bawahnya, dan seterusnya hingga blok *ELSE* terakhir atau terdapat kondisi *IF* yang bernilai *true.*

(sumber: https://www.duniailkom.com/tutorial-belajar-c-percabangan-kondisi-if-else-if-bahasa-c/)

1. **Switch Case**

Fungsi *switch case* dalam bahasa pemograman sama dengan pengguna fungsi *IF*.Yang koding dalam *switch case* ini lebih singkat daripada menggunakan *if* dan lebih *simple* untuk digunakan tetapi di sini dalam suatu *case* hanya terdapat satu kodisi tidak seperti *if* yang dpaat kitaa masukkan dengan banyak kondisi. Dan apa bila kondisi tidak terpenuhi maka akan dialihkan kekondisi lain dan apabila tidak terpenuhi akan di masukkan ke kondisi *default* jika di dalam *if* disebut *else*.

(Sumber: <http://kucing-gilak.blogspot.com/2017/01/belajar-fungsi-switch-case-dalam-bahasa.html>)

1. **Perulangan**
2. **While**

Perulangan *While* hanya terdiri dari *condition* atau syarat dan akan terus melakukan perulangan selama syarat terpenuhi.

Berikut adalah bentuk umum dari *loop While*  
  
while(Condition)  
{  
 statement;  
}

(Sumber : <http://kucing-gilak.blogspot.com/2017/02/fungsi-loop-perulangan-while-dalam.html>)

1. **Do While**

Perulangan *(Loop)* *do-while* adalah perulangan yang melakukan perulangan terlebih dulu kemudian menentukan apakah perulanganya masih memenuhi syarat.

Yang mana bentuk perulangan *Do-While* secara umum adalah:

do  
{ statement;}  
while(condition);

(Sumber ; <http://kucing-gilak.blogspot.com/2017/02/fungsi-pernyataan-perulangan-do-while.html>)

1. **For**

Struktur pengulangan / perulangan jenis *for* biasanya digunakan untuk melakukan perulangan yang telah diketahui banyaknya. Biasanya jenis perulangan *for* dianggap sebagai jenis perulangan yang paling mudah dipahami.

Untuk melakukan perulangan dengan menggunakan struktur perulangan *for*, harus memiliki sebuah variabel sebagai indeksnya. Namun perlu sekali untuk diperhatikan bahwa tipe data dari variabel yang akan digunakan sebagai indeks haruslah tipe data yang mempunyai urutan yang teratur, misalnya tipe data int.

Sebagai catatan bahwa jika melakukan perulangan yang sifatnya menaik (*increment*) maka nilai awal dari variabel yang didefinisikan haruslah lebih kecil dari nilai akhir yang dituliskan dalam kondisi (kondisi saat berjalan). Sebaliknya jika akan melakukan perulangan yang sifatnya menurun (*decrement*) maka nilai awal harus lebih besar dari nilai akhir.

1. Hasil dan Analisa Percobaan
2. **Pengkodisian dalam Java**
3. **If**

package if\_26;

public class If\_26 {

public static void main(String[] args) {

int i = 20;

if(i > 10){

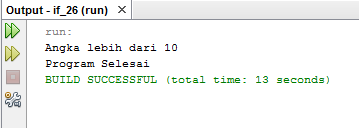
System.out.println("Angka lebih dari 10");

}

System.out.println("Program Selesai");

}

}

****

Gambar 3. 1 output if pada java

Pertama deklarasikan variabel int dan diberi nilai 20 kemudian dengan kondisi *if* variabel tersebut dibandingkan menggunakan operator lebih dari. Karena kondisi tersebut benar maka *statement* didalam *if* akan dieksekusi yaitu mencetak *output* “Angka lebih dari 10”.

1. **If.. Else**

package if\_else\_26;

import java.util.Scanner;

public class If\_else\_26 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Masukkan Angka = ");

int i = sc.nextInt();

if(i > 10){

System.out.println("Angka lebih besar dari 10 yaitu "+i);

}

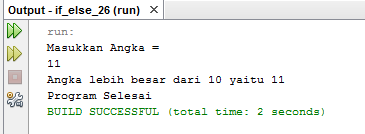
else {

System.out.println("Angka kurang dari 10 yaitu "+i);

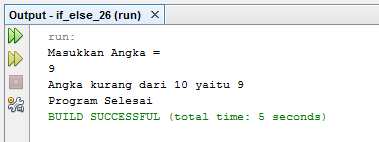
}

System.out.println("Program Selesai");

}

****

Gambar 3. 2 output If .. Else pada java



Gambar 3. 3 Output If .. Else pada java

Pertama kita menggunakan *scanner* berfungsi untuk menangkap *input*-an dari program yang dibuat. Lalu mencetak tulisan untuk *input*-an dan memasukkan nilainya kedalam variabel i. Kemudian menggunakan perintah *if* untuk mengecek kondisinya jika benar maka akan mencetak “Angka lebih besar dari 10” dan jika kondisinya salah akan mencetak “Angka kurang dari 10 yaitu”+i dimana i bernilai angka yang dimasukkan ke program.

1. **If.. Else If.. Else**

package if\_else\_if\_else\_26;

import java.util.Scanner;

public class If\_else\_if\_else\_26 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Masukkan Angka = ");

int i = sc.nextInt();

if(i == 0){

System.out.println("Angka yang dimasukkan adalah 0");

}

else if(i > 0){

System.out.println("Bilangan Bulat Positif yaitu "+i);

}

else if(i < 0){

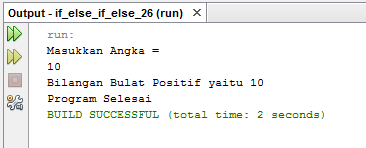
System.out.println("Bilangan Bulat Negatif yaitu "+i);

}

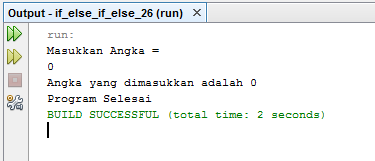
System.out.println("Program Selesai");

}

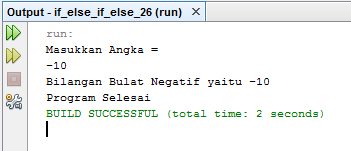
}

****

Gambar 3. 4 Output If else if else pada java



Gambar 3. 5 Output If else if else pada java

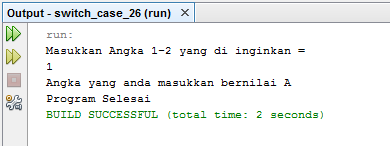


Gambar 3. 6 Output If else if else pada java

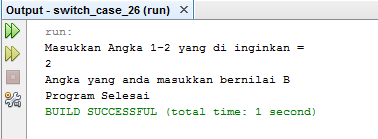
Pertama kita menggunakan *scanner* berfungsi untuk menangkap *input*-an dari program yang dibuat. Lalu mencetak tulisan untuk *input*-an dan memasukkan nilainya kedalam variabel i. Kemudian menggunakan perintah if(i == 0)untuk mengecek kondisinya jika benar maka akan mencetak “Angka yang dimasukkan adalah 0” kemudian jika kondisinya tidak terpenuhi akan diperiksa oleh else if (i > 0) jika kondisi benar akan mencetak “Bilangan Bulat Positif yaitu”+i dimana i bernilai angka yang dimasukkan ke program. Lalu jika kondisinya tidak terpenuhi juga akan diperiksa oleh else if (i < 0) jika kondisi benar akan mencetak “Bilangan Bulat Negatif yaitu”+i dimana i bernilai angka yang diinputkan ke program.

1. **Switch Case**

|  |
| --- |
| package switch\_case\_26;  import java.util.Scanner;  public class Switch\_case\_26 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.println("Masukkan Angka 1-2 yang di inginkan = ");  int i = sc.nextInt();  switch(i){  case(1):System.out.println("Angka yang anda masukkan bernilai A");break;  case(2):System.out.println("Angka yang anda masukkan bernilai B");break;  default:System.out.println("Angka yang anda masukkan tak bernilai A ataupun B");break;  }  System.out.println("Program Selesai");  }  } |

****

Gambar 3. 7 Ouput switch case pada java



Gambar 3. 8 Output switch case pada java

Pertama kita menggunakan *scanner* berfungsi untuk menangkap *input*-an dari program yang dibuat. Lalu mencetak tulisan untuk *input*-an dan memasukkan nilainya kedalam variabel i. Kemudian dengan perintah *switch* akan memeriksa variabel i, jika variabel i nilainya sama dengan salah satu *case* yang ada maka akan dieksekusi *statement* yang ada pada *case* tersebut, case(1) yang akan mencetak “Angka yang anda masukkan bernilai A” lalu case(2) yang akan mencetak “Angka yang anda masukkan bernilai B” dan *default* yang akan dieksekusi ketika semua *case* tidak ada yang terpenuhi. Fungsi break; untuk menghentikan perintah agar *case* dibawahnya tidak ikut dieksekusi.

1. **Perulangan dalam Java**
2. **DO WHILE**

package do\_while\_26;

public class Do\_while\_26 {

public static void main(String[] args) {

int i=1;

do{

System.out.println("Selamat Datang di Praktikum DKP dengan perulangan DO WHILE ke-"+i);

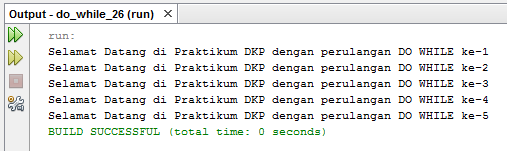
i++;

}

while(i<=5);

}

}

****

Gambar 3. 9 Output DO pada java

Pertama deklarasi variabel i yang bernilai 1 karena ingin memulai perulangan dari nomor 1. Lalu perintah *do* akan mengeksekusi perintah yang akan mencetak “Selamat Datang di Praktikum DKP dengan perulangan DO WHILE ke-+i” dimana i akan dimulai dari 1 kemudian i++ akan di-*increment* selama kondisi masih terpenuhi yaitu i<=5.

1. **WHILE**

package while\_26;

public class While\_26 {

public static void main(String[] args) {

int i=1;

while(i<=5){

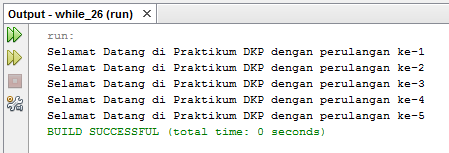
System.out.println("Selamat Datang di Praktikum DKP dengan perulangan ke-"+i);

i++;

}

}

}

****

Gambar 3. 10 Output WHILE pada java

Pertama deklarasi variabel i yang bernilai 1 karena ingin memulai perulangan dari nomor 1. Selama kondisi *while* masih terpenuhi yaitu i<=5 akan dieksekusi perintah untuk mencetak “Selamat Datang di Praktikum DKP dengan perulangan ke-”+i dimana i dimulai dari satu kemudian i++ untuk melakukan *increment* pada variabel i.

1. **FOR**

package for\_26;

public class For\_26 {

public static void main(String[] args) {

// TODO code application logic here

int i;

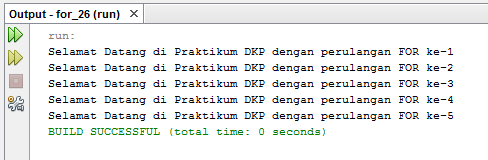
for(i=1; i<=5; i++){

System.out.println("Selamat Datang di Praktikum DKP dengan perulangan FOR ke-"+i);

}

}

}

****

Gambar 3. 11 Output FOR pada java

Pertama deklarasi variabel i untuk menampung nilai inisialisasi, lalu menggunakan perintah *for* dengan inisialisasi 1 dan syarat i<=5 maka i++ nilai i akan terus bertambah (*increment*) selama syarat masih terpenuhi. Kemudian akan mencetak *output* "Selamat Datang di Praktikum DKP dengan perulangan FOR ke-"+i dengan i dimulai dari angka 1.

1. **Pengkodisian dalam C#**
2. **If**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace if\_26

{

class Program

{

static void Main(string[] args)

{

int nilai;

nilai = 45;

if(nilai > 0 && nilai < 80)

{

Console.WriteLine("Success Condition");

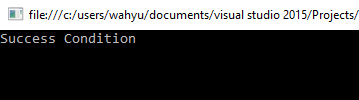
Console.ReadLine();

}

}

}

}

****

Gambar 3. 12 Output if pada C#

Pertama deklarasikan variabel int bernama nilai dan diberi nilai 20 kemudian dengan kondisi *if* variabel tersebut dibandingkan menggunakan operator lebih dari 0 dan kurang dari 80. Karena kondisi tersebut benar maka *statement* didalam *if* akan dieksekusi yaitu mencetak *output* “*Success Condition*”.

1. **If.. Else**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace if\_else\_26

{

class Program

{

static void Main(string[] args)

{

float a;

string angka;

Console.WriteLine("PROGRAM BILANGAN BULAT");

Console.Write("Masukkan Angka = ");

angka = Console.ReadLine();

float.TryParse(angka, out a);

if(a>0)

{

Console.WriteLine("BILANGAN BULAT POSITIF");

Console.ReadLine();

}

else

{

Console.WriteLine("BILANGAN BULAT NEGATIF");

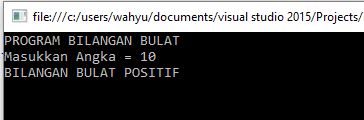
Console.ReadLine();

}

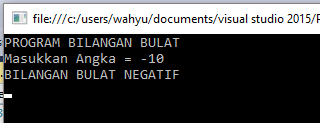
}

}

}

****

Gambar 3. 13 Output If .. Else pada C#

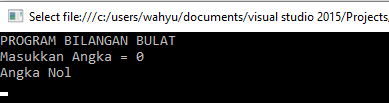


Gambar 3. 14 Output If .. Else pada C#

Pertama deklarasikan variabel a bertipe *float* dan angka bertipe *string*. Lalu mencetak tulisan untuk *input*-an dan memasukkan nilainya kedalam variabel angka dengan perintah Console.ReadLine(). Kemudian menggunakan perintah *Tryparse* untuk mengkonversi tipe *string* ke tipe *float* dengan variabel a. Kemudian menggunakan perintah if(a>0) akan diperiksa varibel a, jika kondisi tersebut benar maka akan mencetak *output* "BILANGAN BULAT POSITIF" sedangkan jika kondisi salah maka akan dieksekusi perintah *else* yang akan mencetak "BILANGAN BULAT NEGATIF".

1. **If.. Else If.. Else**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace if\_else\_if\_26  {  class Program  {  static void Main(string[] args)  {  float a;  string angka;  Console.WriteLine("PROGRAM BILANGAN BULAT");  Console.Write("Masukkan Angka = ");  angka = Console.ReadLine();  float.TryParse(angka, out a);  if (a > 0)  {  Console.WriteLine("BILANGAN BULAT POSITIF");  Console.ReadLine();  }  else if (a == 0)  {  Console.WriteLine("Angka Nol");  Console.ReadLine();  }  else  {  Console.WriteLine("BILANGAN BULAT NEGATIF");  Console.ReadLine();  }  }  }  } |

****

Gambar 3. 15 Output If .. Else If .. Else pada C#

Pertama deklarasikan variabel a bertipe *float* dan angka bertipe *string*. Lalu mencetak tulisan untuk *input*-an dan memasukkan nilainya kedalam variabel angka dengan perintah Console.ReadLine(). Kemudian menggunakan perintah *Tryparse* untuk mengkonversi tipe *string* ke tipe *float* dengan variabel a. Kemudian menggunakan perintah if (a>0) akan diperiksa varibel a, jika kondisi tersebut benar maka akan mencetak *output* "BILANGAN BULAT POSITIF". Apabila kondisi belum terpenuhi akan lanjut pada perintah else if (a == 0)jika kondisi tersebut benar akan dicetak *output* “Angka Nol” sedangkan jika semua kondisi diatas salah maka akan dieksekusi perintah *else* yang akan mencetak "BILANGAN BULAT NEGATIF".

1. **Switch Case**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace switch\_case\_26

{

class Program

{

static void Main(string[] args)

{

string nilai;

Console.Write("Masukkan Nilai (A-E) : ");

nilai = Console.ReadLine();

switch (nilai)

{

case "A":

Console.WriteLine("Nilai Anda\t: {0} (80 - 100)", nilai);

Console.ReadLine();break;

case "B":

Console.WriteLine("Nilai Anda\t: {0} (70 - 79)", nilai);

Console.ReadLine(); break;

case "C":

Console.WriteLine("Nilai Anda\t: {0} (60 - 69)", nilai);

Console.ReadLine(); break;

case "D":

Console.WriteLine("Nilai Anda\t: {0} (50 - 59)", nilai);

Console.ReadLine(); break;

case "E":

Console.WriteLine("Nilai Anda\t: {0} (0 - 49)", nilai);

Console.ReadLine(); break;

default:

Console.WriteLine("Nilai yang Anda Masukkan SALAH");

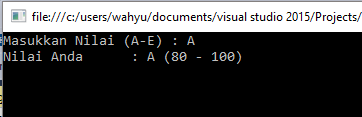
Console.ReadLine(); break;

}

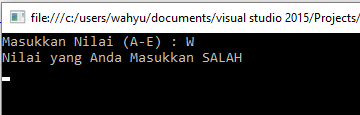
}

}

}

****

Gambar 3. 16 Output switch case pada C#



Gambar 3. 17 Output switch case pad a C#

Pertama deklarasi variabel *string* dengan nama nilai. Kemudian membuat tulisan untuk *input*-an dan memasukkan inputan tersebut kedalam variabel nilai menggunakan Console.ReadLine(). Lalu perintah *switch* akan memeriksa variabel nilai dengan semua *case* yang ada apabila terdapat *case* yang terpenuhi akan mencetak berdasarkan *case* variabel tersebut, contohnya jika variabel A maka akan terpenuhi *case* A yang akan mencetak("Nilai Anda\t: {0} (80 - 100)", nilai) dimana {0} akan diisi dengan variabel nilai. Sedangkan jika semua *case* tidak terpenuhi akan dieksekusi *default* yang akan mencetak ("Nilai yang Anda Masukkan SALAH");

**3.3.4. Perulangan dalam C#**

1. **While**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace while\_26

{

class Program

{

static void Main(string[] args)

{

int angka = 1;

while (angka<=5)

{

Console.WriteLine("PERULANGAN WHILE KE -" + angka);

angka++;

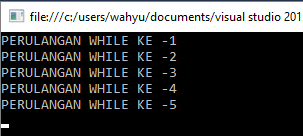
}

Console.ReadLine();

}

}

}



Gambar 3. 18 Output while pada C#

Pertama deklarasi variabel angka yang bernilai 1 karena ingin memulai perulangan dari nomor 1. Selama kondisi *while* masih terpenuhi yaitu i<=5 akan dieksekusi perintah untuk mencetak ”PERULANGAN WHILE KE –“ + angka dimana angka dimulai dari satu kemudian angka++ untuk melakukan *increment* pada variabel angka.

1. **Do While**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace do\_while\_26

{

class Program

{

static void Main(string[] args)

{

int angka = 1;

do

{

Console.WriteLine("PERULANGAN DO WHILE KE-" + angka);

angka++;

}

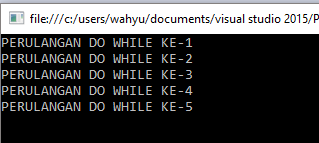
while (angka <= 5);

Console.ReadLine();

}

}

}



Gambar 3. 19 Output Do While pada C#

Pertama deklarasi variabel angka yang bernilai 1 karena ingin memulai perulangan dari nomor 1. Lalu perintah *do* akan mengeksekusi perintah yang akan mencetak “PERULANGAN DO WHILE KE-” + angka dimana angka akan dimulai dari 1 kemudian angka++ akan di-*increment* selama kondisi masih terpenuhi yaitu i<=5.

1. **For**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace for\_26

{

class Program

{

static void Main(string[] args)

{

int jumlah;

for (jumlah=1; jumlah<=5; jumlah++)

{

Console.WriteLine("PERULANGAN FOR KE-" + jumlah);

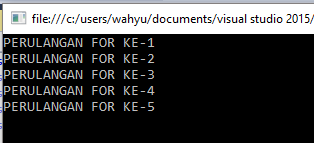
}

Console.ReadLine();

}

}

}

****

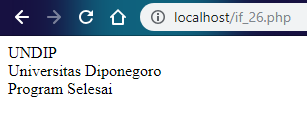
Gambar 3. 20 Output For pada C#

Pertama deklarasi variabel jumlah untuk menampung nilai inisialisasi, lalu menggunakan perintah *for* dengan inisialisasi 1 dan syarat jumlah<=5 maka jumlah++ nilai jumlah akan terus bertambah (*increment*) selama syarat masih terpenuhi. Kemudian dengan perintah Console.WriteLine akan mencetak ("PERULANGAN FOR KE-" + jumlah); dengan jumlah dimulai dari angka 1.

**3.3.5. Pengkodisian dalam PHP**

1. **If**

|  |
| --- |
| <?php  $ptn = "UNDIP";  if($ptn == "UNDIP"){  echo $ptn, "<br>Universitas Diponegoro<br>";  }  echo "program selesai";  ?> |

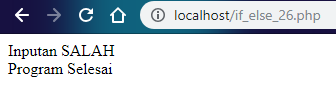
****

Gambar 3. 21 Output If pada PHP

Pada program ini pengkondisian menggunakan perintah *if* yang seperti tertera pada *source code*. Disana dijelaskan bahwa jika variabel nya yaitu UNDIP, maka akan memunculkan output yaitu Universitas Diponegoro. Karena diawal telah di deklarasikan bahwa $ptn = "UNDIP";, maka pengkondisian nya akan sesuai dan akan memunculkan *output* “Universitas Diponegoro” seperti pada Gambar 3.21

1. **If.. Else**

|  |
| --- |
| <?php  $ptn = "UGM";  if($ptn == "UNDIP"){  echo $ptn, "<br>Universitas Diponegoro<br>";  }  else{  echo "Inputan SALAH<br>";  }  echo "program selesai";  ?> |

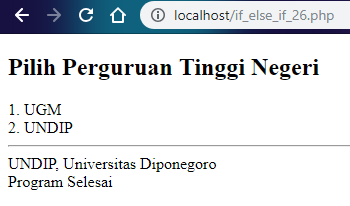
****

Gambar 3. 22 Output If .. Else pada PHP

Pada program ini pengkondisian *if* menjelaskan bahwa apabila variabel ptn sama dengan undip, maka nantinya output akan memunculkan “Universitas Diponegoro”. Lalu juga menggunakan *else*, maksudnya yaitu selain dari yang dikondisikan oleh if, maka akan masuk ke pengkondisian *else*. Disana dijelaskan jika ada pengkondisian yang tidak termasuk dalam pengkondisian *if,* maka akan memunculkan output “Inputan SALAH”. Dan di awal telah di deklarasikan bahwa $ptn = "UGM";, yang menjelaskan bahwa kondisi ini tidak sesuai dengan syarat *if* , maka akan di masuk ke pengkondisian else dan akan memunculkan output seperti pada Gambar 3.22

1. **If.. Else If.. Else**

|  |
| --- |
| <?php  echo "<h2> Pilih Perguruan Tinggi Negeri </h2>";  echo "1. UGM<br>";  echo "2. UNDIP<hr>";  $ptn = "2";  if($ptn == "1"){  echo "UGM, Universitas Gajah Mada<br>";  }  else if ($ptn == "2"){  echo "UNDIP, Universitas Diponegoro<br>";  }  echo "Program Selesai";  ?> |

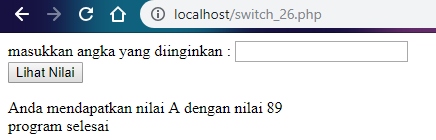
****

Gambar 3. 23 Output If.. Else If.. Else

Pertama memunculkan terlebih dahulu menu pilihan untuk dijadikan *output* nya agar nanti muncul 2 pilihan. Lalu memulai dengan pengkondisian *if*, jika variabelnya yaitu ptn sama dengan 1, maka akan tertulis “UGM, Universitas Gajah mada. Lalu menggunakan pengkondisian *else if* untuk syarat apabila variabel ptn nya sama dengan 2 maka akan muncul output “UNDIP, Universitas Diponegoro”. Karena diawal telah di deklarasikan bahwa variabel ptn nya sama dengan 2, maka akn memunculkan *Output* seperti Gambar 3.23.

1. **Switch Case**

|  |
| --- |
| <form method="post">  masukkan angka yang diinginkan : <input type="text" name="angka"> <br>  <input type="submit" value="Lihat Nilai">  </form>  <?php  $angka = isset($\_POST['angka'])?abs((int)$\_POST['angka']):'';  switch ($angka) {  case ($angka >=80):  echo "Anda mendapatkan nilai A dengan nilai ".$angka; break;  case ($angka >=70):  echo "Anda mendapatkan nilai B dengan nilai ".$angka; break;  case ($angka >=60):  echo "Anda mendapatkan nilai C dengan nilai ".$angka; break;    default:  echo "Anda harus mengulang! Semangat! Karena nilai anda ".$angka; break;  }  echo "<br>program selesai<br>";  ?> |

****

Gambar 3. 24 Output switch case pada PHP

Pada *syntax* HTML diawal dijelaskan bahwa nanti *inputan*-nya akan berbentuk *post* dengan judulnya yaitu “masukkan angka yang dinginkan : ” dan menggunakan *input* tipe *submit* dengan bertuliskan “lihat Nilai” yang berfungsi untuk memasukkan nilai yang telah diisi pada kolom teks sebelumnya.

Selanjutnya pada *syntax* php setelah menerima *input*an tadi, maka akan diproses menggunakan pengkondisian *switch*. Dijelaskan bahwa apabila variabel angka yang dimasukkan >= 80, maka akan muncul output “Anda Mendapatkan nilai A dengan nilai “ lalu memuncukan variabel yang telah di *input*kan tadi.

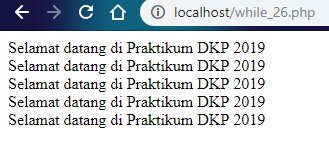
Begitu juga apabila variabel angka >= 70 atau variabel angka >= 60, maka *output*nya akan muncul sesuai dengan yang tertera pada *echo* nya.

Lalu untuk *Syntax* *default* , berfungsi sebagai persyaratan lainnya apabila dari 3 syarat sebelumnya tidak ada yang memenuhi, dan akan memunculkan *output* “ Anda harus mengulang! Semangat! Karena nilai anda (variabel angka yang telah dimasukkan sebelumnya)”.

**3.3.6. Perulangan PHP**

1. **While**

|  |
| --- |
| <?php  $i=1;  while ($i <= 5) {  echo "Selamat datang di Praktikum DKP 2019";  echo "<br />";  $i++;  }  ?> |

****

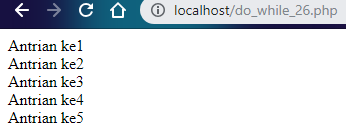
Gambar 3. 25 Output While pada PHP

Pada perulangan *while* dijelaskan pada *syntax-*nya apabila variabel i kurang atau sama dengan 5, maka akan terjadi perulangan sampai pada i = 5. Disana tertera $i++; yang bermakna, variabel i akan terus ditambah 1 sampai pada kondisi yang diingikan yang pada program ini yaitu 5.

Perulangan yang dimaksud yaitu perulangan teks yang tertulis pada *syntax* yakni “Selamat datang di Praktikum DKP 2019. Dan karena diawal di deklarasikan bahwa i=1 maka akan muncul *output* yang tertera seperti pada Gambar 3.25

1. **Do While**

|  |
| --- |
| <?php  $i=1;  do {  echo "Antrian ke",$i;  echo "<br />";  $i++;  } while ($i <= 5);  ?> |

****

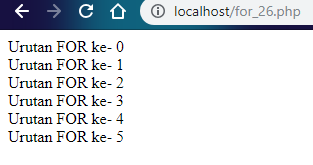
Gambar 3. 26 Output Do While pada PHP

Pada metode perulangan *do while, do* berguna untuk apa yang nantinya akan di lakukan jika variabel telah memenuhi *while*. dan *while* berfungsi sebagai syarat untuk menentukan apakah *do* harus dilaksanakan atau tidak. Pada perintah *do* dituliskan bahwa akan mengeluarkan output “Antrian ke(variabel angka yang dimasukkan)” dan *while* menjelaskan bahwa *do* akan dilaksanakan apabila variabel i<= 5

Diawal telah dideklarasikan bahwa Variabel i=1 yang nanti nya program akan berjalan seperti pada Gambar 3.26

1. **For**

|  |
| --- |
| <?php  $i=1;  for ($i=0; $i <=5 ; $i++) {  echo "Urutan FOR ke- ",$i;  echo "<br />";  }  ?> |



Gambar 3. 27 Output For pada PHP

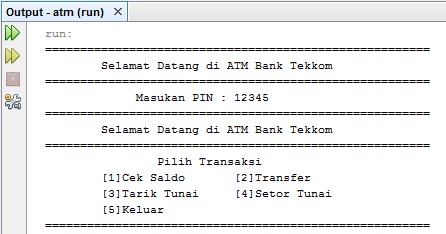
Untuk perulangan *For*, semua syaratnya langsung di masukkan pada 1 parameter seperti yang tertulis pada *syntax* for ($i=0; $i <=5 ; $i++) {, disana dijelaskan bahwa ini diperuntukkan dari i=0 dan I<=5. Dan juga nantinya setiap perulangan, variabelnya akan ditambah dengan 1.

Lalu apabila variabel memenuhi syarat tadi, maka akan muncul output “Urutan FOR ke – (variabel yang telah dimasukkan tadi)”.

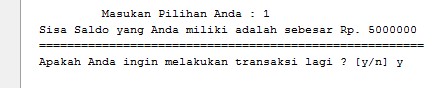
Karena diawal telah di deklarasikan bahwa variabel i=1. Maka akan muncul *output* seperti yang tertera pada Gambar 3.27.

1. Tugas
2. **Tugas**

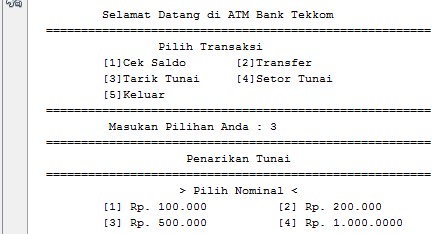
|  |
| --- |
| package atm;  import java.util.Scanner;  public class Atm {    Scanner input = new Scanner(System.in);  int pil;  int saldo = 5000000;  int minSaldo = 50000;  public void login() {  int pin;  System.out.println("=======================================================");  System.out.println("\tSelamat Datang di ATM Bank Tekkom");  System.out.println("=======================================================");  System.out.print("\t Masukan PIN : ");  pin = input.nextInt();  if (pin == 12345) {  menu();  } else {  System.out.println("Gagal");  login();  }  }  public void menu(){  System.out.println("=======================================================");  System.out.println("\tSelamat Datang di ATM Bank Tekkom");  System.out.println("=======================================================");  System.out.println("\t\tPilih Transaksi");  System.out.println("\t[1]Cek Saldo [2]Transfer");  System.out.println("\t[3]Tarik Tunai [4]Setor Tunai");  System.out.println("\t[5]Keluar");  System.out.println("=======================================================");  System.out.print("\t Masukan Pilihan Anda : ");  pil = input.nextInt();    switch (pil) {  case 1:  System.out.println("Sisa Saldo yang Anda miliki adalah sebesar Rp. " + saldo);  transaksiLagi();  break;  case 2:  int norek;  int nom;  System.out.println("=======================================================");  System.out.print("\t\tTransfer");  System.out.println();  System.out.println("=======================================================");  System.out.print("\tMasukan Nomor Rekening : ");  norek = input.nextInt();  System.out.println();  System.out.print("\tMasukan Nominal Transfer : ");  nom = input.nextInt();    if (saldo < 50000) {  System.out.println("\tMaaf Saldo Anda tidak mencukupi.");  } else {  saldo -= nom;  System.out.println("Anda telah berhasil melakukan transfer sebesar : " + nom);  System.out.println("Ke nomor rekening : " + norek);  System.out.println("\t Sisa Saldo adalah :" + saldo);  }  transaksiLagi();  break;  case 3:  tarikTunai();  break;  case 4:  int uang;  System.out.println("=======================================================");  System.out.println("\t Setor Tunai");  System.out.println("=======================================================");  System.out.print("\tMasukan Nominal : ");  uang = input.nextInt();  saldo += uang;  System.out.println("Anda telah berhasil melakukan setoran sebesar : " + uang);  System.out.println("\t Sisa Saldo adalah :" + saldo);  transaksiLagi();  break;  case 5:  System.out.println("=======================================================");  System.out.println("\tTerimakasih telah menggunakan layanan kami.");  System.out.println("\t Silahkan Ambil kartu Anda.");  break;  default:  System.out.println("Pilihan Yang Anda masukan salah. Silahkan Login kembali.");  login();  break;  }  }  public void tarikTunai(){  System.out.println("=======================================================");  System.out.print("\t Penarikan Tunai");  System.out.println();  System.out.println("=======================================================");  System.out.print("\t > Pilih Nominal <");  System.out.println();  System.out.print("\t[1] Rp. 100.000 [2] Rp. 200.000 ");  System.out.println();  System.out.print("\t[3] Rp. 500.000 [4] Rp. 1.000.0000 ");  System.out.println();  System.out.println("=======================================================");  System.out.print("\t Masukan pilihan Anda : ");  pil = input.nextInt();  System.out.println("=======================================================");  switch (pil) {  case 1:  tarikNominal(100000);  break;  case 2:  tarikNominal(200000);  break;  case 3:  tarikNominal(500000);  break;  case 4:  tarikNominal(1000000);  break;  default:  System.out.println("\tPilihan Yang Anda masukan salah");  tarikTunai();  }  }  public void tarikNominal(double nominal) {  if (saldo < 50000) {  System.out.println("\tMaaf Saldo Anda tidak mencukupi.");  } else {  saldo -= nominal;  System.out.println("Anda telah berhasil melakukan penarikan sebesar : " + nominal);  System.out.println("\t Sisa Saldo adalah :" + saldo);  }  transaksiLagi();  }  public void transaksiLagi(){  String text;  Scanner scan = new Scanner(System.in);  System.out.println("=======================================================");  System.out.print("Apakah Anda ingin melakukan transaksi lagi ? [y/n] ");  text = scan.nextLine();  System.out.println();  if (text.equalsIgnoreCase("y")) {  menu();  } else if (text.equalsIgnoreCase("n")) {  System.out.println("=======================================================");  System.out.println("\tTerimakasih telah menggunakan layanan kami.");  System.out.println("\t Silahkan Ambil kartu Anda.");  }  }  public static void main(String[] args) {  Atm jalan = new Atm();  jalan.login();  }    } |

****

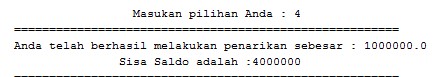
Gambar 3. 28 Tampilan program ATM



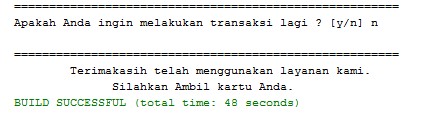
Gambar 3. 29 Menu Cek Saldo



Gambar 3. 30 Menu Penarikan Tunai



Gambar 3. 31 Penarikan Tunai berhasil



Gambar 3. 32 Menu Keluar ATM

Pertama *import scanner* untuk mengambil masukkan dari program, lalu deklarasikan variabel yang dibutuhkan yaitu pil untuk memilih menu saldo untuk memberikan saldo awal.

Kemudian fungsi main() akan memanggil fungsi login() untuk membuat program harus memasukkan pin sebelum berjalan, jika benar pinnya maka akan ditampilkan menu jika salah maka akan mencetak "Gagal" dan harus kembali *login.*

Lalu fungsi menu() untuk menampilkan menu yang dapat diberi *input* jika salah maka akan dialihkan untuk login kembali. Ketika memberi *input* akan tampil salah satu menu:

1. Menampilkan saldo

Mencetak variabel saldo dengan System.out.println();

1. Menampilkan transfer

Mendeklarasikan variabel norek sebagai nomor rekening tujuan dan variabel nom sebagai nominal yang akan ditransfer.

Kemudian menampilkan menu transfer untuk memasukkan nomor rekening yang akan dimasukkan kedalam variabel norek dan memasukkan nominal transfer untuk dimasukkan kedalam variabel nom.

Jika saldo kurang dari 50000 maka akan ditampilkan pesan "Maaf Saldo Anda tidak mencukupi" jika saldo lebih dari 50000 maka akan dijalankan else yang akan mengeksekusi pengurangan saldo berdasar nominal yang diinputkan. Lalu akan menampilkan pesan "Anda telah berhasil melakukan transfer sebesar: " + nom dimana nom adalah jumlah nominal yang ditransfer.

1. Menampilkan tarik tunai

Akan memanggil fungsi tarikTunai() yang berfungsi untuk menampilkan menu Penarikan Tunai yang dapat menampilkan pilihan uang yang dapat diambil.

Ada beberapa pilihan yaitu: Rp. 100000, Rp. 200000, Rp. 500000, Rp. 1000000

Pilihan diatas akan dimasukkan kedalam fungsi tarikNominal() sebagai parameter yang diinputkan. Lalu fungsi tarikNomial() akan mengecek apakah saldo kurang dari 50000 jika benar maka akan mencetak "Maaf Saldo Anda tidak mencukupi" namun jika saldo lebih dari 50000 akan dieksekusi else yang akan melakukan pengurangan saldo sebanyak nominal yang dimasukkan sebagai parameter dan mencetak "Anda telah berhasil melakukan penarikan sebesar : " + nominal dimana nominal adalah besar uang yang pilih ketika melakukan penarikan. Kemudian akan mengecek sisa saldo dengan memanggil variabel saldo.

1. Menampilkan setor tunai

Mendeklarasikan variabel uang untuk menampung jumlah uang yang akan disetor kemudian akan menampilkan menu setor tunai. Lalu akan mencetak tampilan untuk memasukkan nominal yang akan disetor dan memasukkan nilai *input*-an kedalam variabel uang.

Lalu akan dilakukan penambahan saldo berdasarkan *input*-an variabel uang. Kemudian akan mencetak "Anda telah berhasil melakukan setoran sebesar : " + uang dimana uang bernilai angka yang dimasukkan. Lalu akan mengecek sisa saldo dengan memanggil variabel saldo.

1. Menampilkan pesan keluar

Mencetak tulisan untuk menandakan telah berhasil keluar dan program akan berhenti.

Pada setiap pilihan menu diatas terdapat fungsi transaksiLagi() yang fungsinya menampilkan pesan apakah ingin melakukan transaksi lagi dan menangkap *input* dari program jika diinput "y" maka program akan kembali ke fungsi menu jika diinput "n" maka program akan berhenti.

1. Kesimpulan
2. Kondisi *Boolean* (kondisi benar atau salah) menjadi patokan dalam eksekusi pengkondisian.
3. Perulangan dalam bahasa C#, Java, maupun PHP mempunyai tujuan yang sama yakni mengerjakan pernyataan jika keadaan bernilai benar *(true)* atau memenuhi syarat untuk mengerjakan pernyataan tersebut dan tidak mengeksekusi ketika keadaan bernilai salah *(false).*
4. *If* akan mengeksekusi perintah yang kondisinya bernilai benar maka akan dijalankan perintah yang ada didalamnya.
5. *If - else* akan mengeksekusi perintah yang kondisinya bernilai benar maka akan dijalankan perintah yang ada didalam *if*. Jika salah akan dijalankan perintah didalam *else.*
6. *If – else if - else* sama seperti sintaks *if - else* namun dapat diberikan banyak kondisi. Jika kondisi pertama (*if*) tidak terpenuhi akan mengecek kondisi berikutnya (*else if*) dan seterusnya jika semua tidak terpenuhi akan dijalankan perintah *else.*
7. *Switch case* digunakan untuk pengkondisian dimana ada banyak kondisi yang berbeda. jika tidak ada kondisi yang memenuhi maka perintah *default* yang akan dijalankan.
8. Pada *switch case*, setelah *case* selalu ada *break* yang fungsinya untuk mengakhiri proses saat *statement*  dalam *case* telah dijalankan.
9. Jika tidak diakhiri dengan *break* maka proses pada *switch case* akan terus dilanjutkan sampai *case* terakhir.
10. *Syntax* pada C#, Java, dan PHP untuk penulisan *while* sama, dengan menggunakan kata *while* dengan kondisi untuk menjalankan perulangan tersebut, serta perintah untuk tiap perulangan.
11. *Syntax* pada C#, Java, dan PHP untuk penulisan *for* sama, menggunakan kata *for* diikuti inisialisasi variabel, kemudian kondisi agar perulangan berjalan, dan *increment* atau *decrement* variabel yang dijalankan tiap selesai perulangan.