**BAB IV**

**PERULANGAN**

## 4.3 Hasil dan Analisa Percobaan

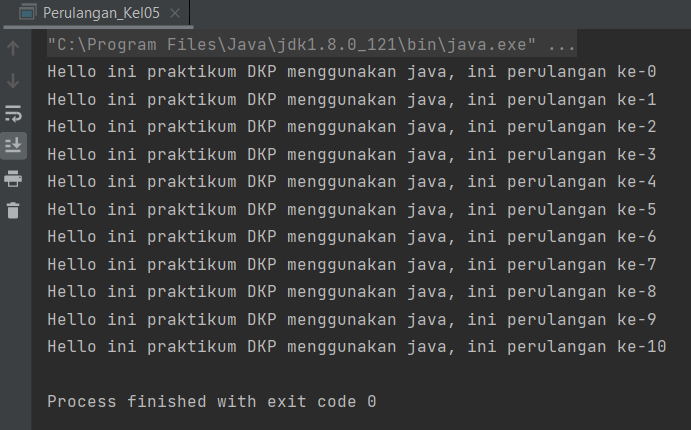
### 4.3.1 Perulangan dalam Java

* 1. While

Berikut ini adalah *source code* perulangan *while* pada Java:

|  |
| --- |
| public class Perulangan\_Kel05 {  public static void main(String[] args) {  int i=0;  while (i<=10)  {  System.out.println("Hello ini praktikum DKP menggunakan java, ini perulangan ke-"+i);  i++;  }  }  } |

Berikut adalah output dari *source code* di atas:



Gambar 4. x Output perulangan while Java

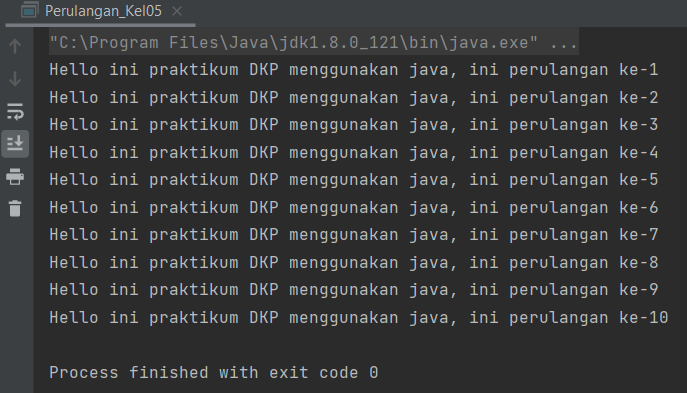
Pada *source code* ini *public static void main* adalah *main method* yang merupakan metode utama untuk menjalankan program. *Source code* dimulai dengan memasukkan variabel *integer* i yang bernilai nol. Kemudian pada baris berikutnya, membuat deklarasi perulangan *while* dengan variabel integer i sebelumnya, bernilai kurang dari sama dengan sepuluh supaya output akan membuat *loop* dari nol hingga sepuluh. Lalu membuat output *string* sebagai keterangan jumlah loop ditambah variabel i yang kemudian setelahnya diikuti *increment* (i++) supaya variabel dalam perulangan akan terulang dan bertambah sesuai pada deklarasi while, yaitu kurang dari sama dengan sepuluh, sehingga program akan berulang dan bertambah dari nol hingga sepuluh.

* 1. Do While

Berikut ini adalah *source code* perulangan *do while* pada Java:

|  |
| --- |
| public class Perulangan\_Kel05 {  public static void main(String[] args) {  int i=1;  do  {  System.out.println("Hello ini praktikum DKP menggunakan java, ini perulangan ke-"+i);  i++;  } while (i <=10);  }  } |

Berikut adalah *output* dari *source code* di atas:



Gambar 4. X Output perulangan do while Java

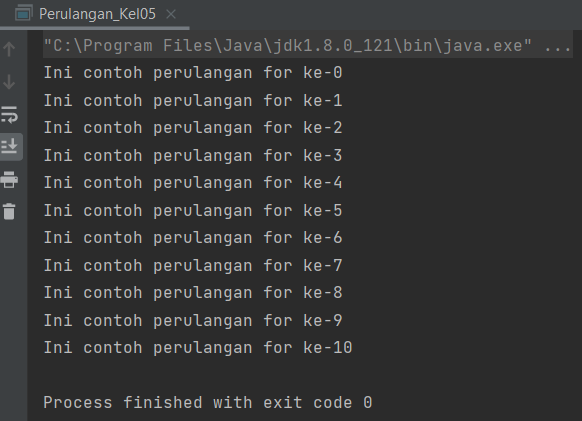
Pada *source code* ini *public static void main* adalah *main method* yang merupakan metode utama untuk menjalankan program. Program di atas dimulai dengan membuat variabel *integer* i yang bernilai satu. Kemudian membuat deklarasi *do* yang diikuti output *string* keterangan perulangan lalu ditambah variabel i. Setelah itu untuk membuat perulangan menjadi bertambah, menggunakan *increment* setelah *output.* Lalu di akhir membuat deklarasi *while* dengan variabel i kurang dari sama dengan sepuluh, untuk membuat perulangan akan berhenti sampai nilai sepuluh. Perbedaan *while* dan *do* *while* pada contoh sebelumnya mungkin tidak terlihat karena bentuk output yang sama, namun pada *do* *while* pernyataan dalam *do* akan langsung dieksekusi tanpa melihat *boolean* *expression* seperti pada *while*.

* 1. For

Berikut ini adalah *source code* perulangan *for* pada Java:

|  |
| --- |
| public class Perulangan\_Kel05 {  public static void main(String[] args) {  for (int i=0;i<=10;i++){  System.out.println("Ini contoh perulangan for ke-"+i);  }  }  } |

Berikut adalah *output* dari *source code* diatas:



Gambar 4. X Output perulangan for Java

Pada *source code* ini *public static void main* adalah *main method* yang merupakan metode utama untuk menjalankan program. *Source code* dimulai langsung dengan deklarasi perulangan *for* dengan pernyataan berisi variabel *integer* i yang bernilai nol dan kurang dari sama dengan sepuluh, lalu diikuti *increment* supaya program *loop* akan bertambah. Deklarasi pada *for* langsung berisi pernyataan penuh untuk batas *loop*, nilai variabel, dan *increment* tanpa harus dipisah baris seperti pada *do while* dan *while*. Lalu terakhir membuat deklarasi *output string* yang diikuti variabel i sebagai hasil *for loop* yang telah berjalan.

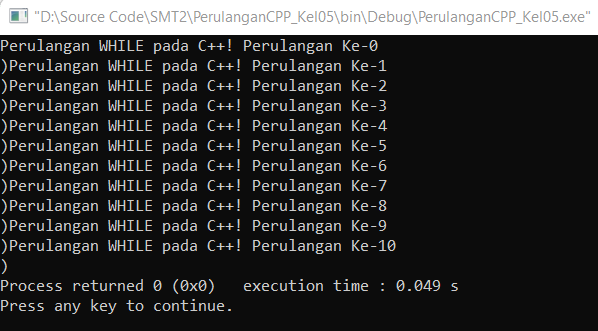
### Perulangan dalam C++

* + - * 1. While

Berikut adalah *source code* perulangan *while* pada C++:

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int i = 0;  while(i<=10){  std::cout<<"Perulangan WHILE pada C++! Perulangan Ke-"<<i<<"\n)";  i++;  }  } |

Berikut adalah *output* dari *source code* diatas:



Gambar 4. X Output perulangan while C++

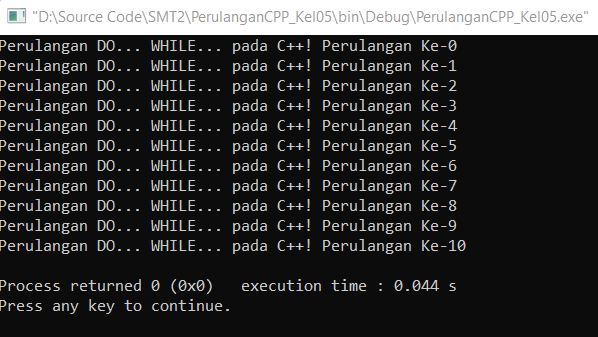
Dalam *source code* ini, #include <iostream> berfungsi untuk meng-*import* fungsi *input* dan fungsi *output* pada bahasa C++. Lalu using namespace std; berfungsi menambahkan *namespace* saat kita menggunakan fungsi *cin* atau fungsi *cout dan int main()* berfungsi memberikan nilai balik menurut tipe datanya. Program dimulai dengan membuat variabel *integer* i yang bernilai nol, lalu deklarasi *while* dengan pernyataan variabel i kurang dari sama dengan sepuluh sebagai batasan *loop* yang akan diproses. Lalu membuat deklarasi *output* std::cout supaya *output* dapat berupa *value* atau nilai dan *output* berupa *string*  diikuti nilai variabel i dan \n untuk memindahkan ke baris selanjutnya. Terakhir, membuat *increment* supaya program *loop* akan bertambah hingga batas yang ditentukan oleh deklarasi *while.* Karena nilai variabel i adalah nol dan batasnya adalah sepuluh, maka program akan membuat *loop* yang bertambah dari nol hingga sepuluh.

* + - * 1. Do While

Berikut adalah *source code* perulangan *do while* pada C++:

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int i = 0;  do{  std::cout<<"Perulangan DO... WHILE... pada C++! Perulangan Ke-"<<i<<"\n";  i++;  } while(i<=10);  } |

Berikut adalah *output* dari *source code* diatas:



Gambar 4. X Output perulangan do while C++

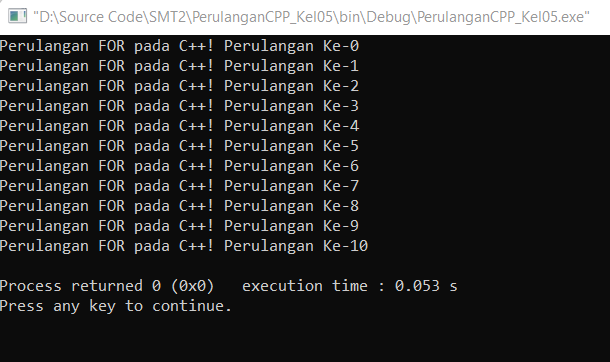
Dalam *source code* ini, #include <iostream> berfungsi untuk meng-*import* fungsi *input* dan fungsi *output* pada bahasa C++. Lalu using namespace std; berfungsi menambahkan *namespace* saat kita menggunakan fungsi *cin* atau fungsi *cout dan int main()* berfungsi memberikan nilai balik menurut tipe datanya. Program dimulai dengan membuat variabel *integer* i yang bernilai nol, lalu langsung deklarasi *do* dengan pernyataan *output* std::cout supaya *output* dapat berupa *value* atau nilai dan *output* berupa *string* diikuti nilai variabel i dan \n untuk memindahkan ke baris selanjutnya. Lalu menambahkan *increment* supaya nilai variabel dalam program *loop* dapat bertambah. Sama seperti pada Java, do while terlebih dahulu mengeksekusi pernyataan pada do lalu membuat deklarasi while dengan batas nilai variabel i hingga sepuluh. Kemudian program akan berjalan dengan membuat loop dari nilai nol hingga sepuluh.

* + - * 1. For

Berikut adalah *source* *code* perulangan *for* pada C++:

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  for(int i=0;i<=10;i++){  std::cout << "Perulangan FOR pada C++! Perulangan Ke-"<<i<<"\n";  }  } |

Berikut adalah output dari source code di atas:



Gambar 4. X Output perulangan for C++

Pada source code di atas, #include <iostream> berfungsi untuk meng-*import* fungsi *input* dan fungsi *output* pada bahasa C++. Lalu using namespace std; berfungsi menambahkan *namespace* saat kita menggunakan fungsi *cin* atau fungsi *cout dan int main()* berfungsi memberikan nilai balik menurut tipe datanya. Program diawali langsung dengan deklarasi *for* yang berisi variabel *integer* i yang bernilai nol, perulangan dengan jumlah kurang dari sama dengan sepuluh, lalu *increment* untuk penambahan nilai variabel dalam *looping*. Lalu membuat std::cout supaya *output* dapat berupa *value* atau nilai dan *output* berupa *string* diikuti nilai variabel i dan \n untuk memindahkan ke baris selanjutnya. Program akan berjalan dengan output berulangnya nilai variabel i yang bertambah dari nol hingga sepuluh.