

LAPORAN PRAKTIKUM

“Pertemuan ke-1: Post Test - CLASS DAN REKURSI”

Diajukan untuk memenuhi salah satu praktikum Mata Struktur Data Informatika yang di
ampu oleh:

Dr., Ardiansyah, S.T., M.Cs.



Disusun Oleh:

Mohammad Farid Hendianto 2200018401

A / Rabu 10.30 – 13.30 Lab. Jaringan

PROGRAM STUDI INFORMATIKA
UNIVERSITAS AHMAD DAHLAN
FAKULTAS TEKNOLOGI INDUSTRI
TAHUN 2023

Cetaklah secara terbalik bilangan INTEGER maksimal 5 digit secara iteratif dan rekursif.
Sebagai contoh :

input : 12345

output : 54321

Gunakan class untuk menampung data integernya dan operasi (fungsi).

Jawab:


Berikut adalah source code kodingan untuk membalikkan integer secara iteratif dan rekursif

```
1  #include <iostream>
2
3  class Number {
4      long long int num;
5  public:
6      Number() : num(0) {}
7
8      void input() {
9
10         std::cout << "Input: ";
11         std::cin >> num;
12         if (num < 1 || num > 99999) {
13             std::cout << "Must five digit number... (click enter to continue)...\n";
14             system("pause");system("cls");input();
15         }
16     }
17
18     void printReverseIterative() const {
19         int rev = 0;
20         for (int temp = num; temp; temp /= 10)
21             rev = rev * 10 + temp % 10;
22         std::cout << "Output (Iterative): " << rev << '\n';
23     }
24
25     void printReverseRecursive() const {
26         printReverseRecursive(num, 0);
27     }
28
29 private:
30     static void printReverseRecursive(int n, int rev) {
31         if(n == 0) {
32             std::cout << "Output (Recursive): " << rev;
33             return;
34         }
35
36         printReverseRecursive(n / 10, rev * 10 + n % 10);
37     }
38 };
39
```

```
40 int main(){  
41     Number n;  
42  
43     n.input();  
44     n.printReverseIterative();  
45     n.printReverseRecursive();  
46  
47     return 0;  
48 }  
49
```

Penjelasan:

Program C++ di atas merupakan program untuk menerima input angka dari pengguna dan mencetak kebalikannya dalam dua cara, yaitu secara iteratif dan rekursif.

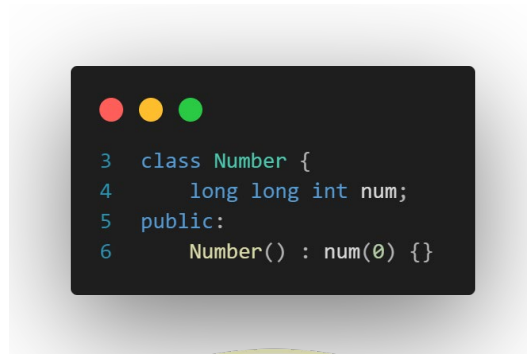


```
1 #include <iostream>  
2 using namespace std;
```

Gambar 1 header <iostream>.

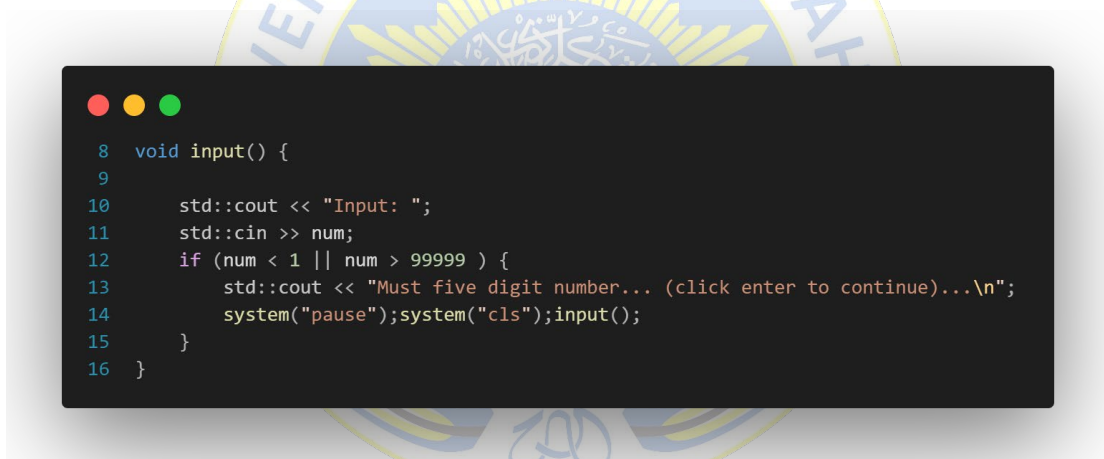
Baris ini adalah preprocessor directive dalam C++. `#include <iostream>` digunakan untuk memasukkan file header standar Input/Output Stream yang mendefinisikan objek seperti `cin`, `cout`, `cerr` dll. Pernyataan `using namespace std;` adalah cara untuk memberi tahu kompiler bahwa kita ingin menggunakan semua fungsi, objek, dan variabel yang didefinisikan dalam ruang nama standar (`std`). Hal ini bisa sangat membantu dalam program kecil di mana kita tidak peduli tentang konflik nama dan hanya ingin mengetik sebanyak mungkin. Namun, dalam program yang lebih besar atau ketika bekerja dengan banyak pustaka berbeda, penggunaan `using namespace std;` dapat menyebabkan masalah. Alasannya adalah bahwa ruang nama standar memiliki banyak fungsi, objek, dan variabel. Jika kita memiliki fungsi atau variabel dengan nama yang sama di tempat lain di kode kita (atau dalam pustaka lain yang kita gunakan), maka akan ada konflik. Misalnya, jika kita memiliki fungsi sendiri bernama `count` dan Anda juga menggunakan pernyataan `using namespace std;`, maka akan ada konflik antara fungsi Anda dan `std::count` dari pustaka standar. Karena alasan ini, programmer senior sering kali memilih untuk tidak menggunakan pernyataan `using namespace std;`, tetapi lebih suka mengetikkan prefix `std::` sebelum setiap penggunaan dari ruang nama standar. Ini membuat jelas bahwa mereka mengacu pada versi standar dari setiap fungsi atau objek, dan menghindari

kemungkinan konflik. Ini juga merupakan praktik baik karena membuat kode lebih mudah dibaca oleh orang lain.



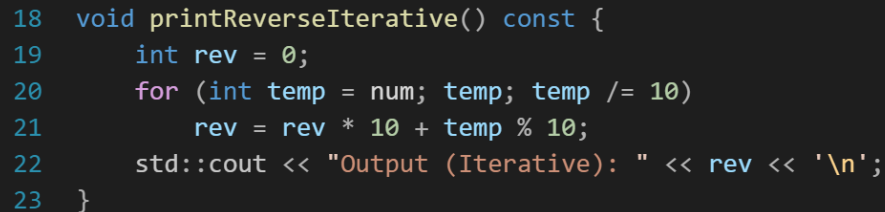
Gambar 2 Inisialisasi variabel num pada class.

Baris ini adalah preprocessor directive dalam C++. `#include <iostream>` digunakan untuk memasukkan file header standar Input/Output Stream yang mendefinisikan objek seperti `cin`, `cout`, `cerr` dll.



Gambar 3 input().

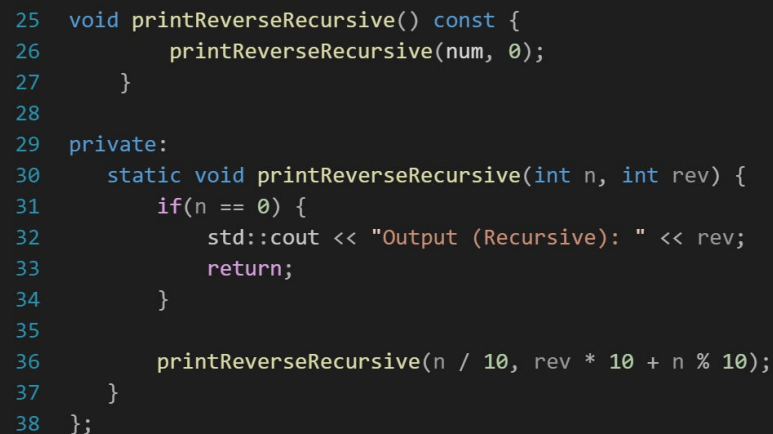
Fungsi `input()` ini digunakan untuk menerima input dari pengguna. Fungsi ini akan meminta pengguna untuk memasukkan angka dan menyimpannya dalam variabel anggota 'num'. Jika pengguna memasukkan nilai di luar rentang [1 hingga 99999] artinya di antara 1 digit hingga 5 digit, maka fungsi ini akan menampilkan pesan kesalahan dan menghentikan eksekusi program sampai pengguna memberi masukan yang valid.



```
18 void printReverseIterative() const {
19     int rev = 0;
20     for (int temp = num; temp; temp /= 10)
21         rev = rev * 10 + temp % 10;
22     std::cout << "Output (Iterative): " << rev << '\n';
23 }
```

Gambar 4 printReverseIterative().

Fungsi printReverseIterative() mencetak kebalikan dari nomor yang dimasukkan oleh pengguna secara iteratif. Itu menggunakan loop for dimana setiap iterasi menghapus digit terakhir dari 'temp' dan menambahkannya ke akhir 'rev'. Proses ini berlanjut sampai tidak ada digit tersisa dalam 'temp'.

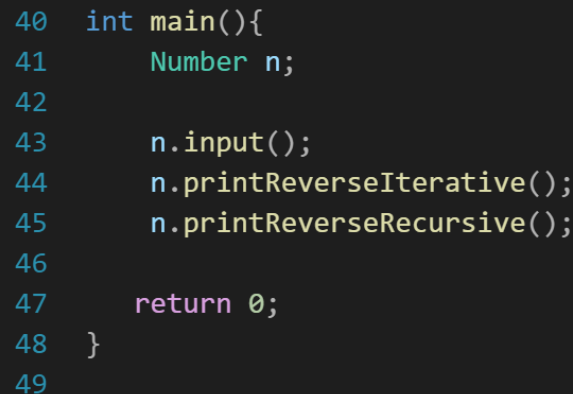


```
25 void printReverseRecursive() const {
26     printReverseRecursive(num, 0);
27 }
28
29 private:
30 static void printReverseRecursive(int n, int rev) {
31     if(n == 0) {
32         std::cout << "Output (Recursive): " << rev;
33         return;
34     }
35     printReverseRecursive(n / 10, rev * 10 + n % 10);
36 }
37
38 };
```

Gambar 5 printReverseRecursive().

Metode publik printReverseRecursive() adalah wrapper sekitar metode pribadi dengan nama yang sama. Metode pribadi adalah fungsi rekursif yang mencetak kebalikan dari nomor

yang dimasukkan oleh pengguna secara rekursif. Fungsi ini membagi 'n' dengan 10 (menghapus digit terakhir) dan menambahkannya ke akhir 'rev' dalam setiap pemanggilan rekursif. Jika 'n' menjadi 0, maka fungsi mencetak 'rev' dan kembali.



```
40 int main(){
41     Number n;
42
43     n.input();
44     n.printReverseIterative();
45     n.printReverseRecursive();
46
47     return 0;
48 }
49
```

Gambar 6 Fungsi utama.

Fungsi `main()` adalah titik masuk program. Pertama, ia membuat objek dari kelas `Number`, kemudian memanggil metode `input()`, `printReverseIterative()`, dan `printReverseRecursive()` pada objek tersebut secara berurutan.

Jadi, program ini adalah contoh yang baik tentang bagaimana menggunakan teknik iteratif dan rekursif untuk memecahkan masalah yang sama dalam pemrograman C++. Teknik rekursif sering digunakan ketika solusi suatu masalah dapat dinyatakan dalam istilah solusi submasalah yang lebih kecil dari jenis yang sama. Sebaliknya, teknik iteratif digunakan ketika kita memiliki loop atau struktur berulang di mana setiap langkah solusi bergantung pada hasil sebelumnya.