

## Laporan Praktikum

# Algoritma dan Struktur Data

Ganjil 2025/2026

Program Studi Teknik Informatika

Institut Teknologi Sumatera



**Modul :** Pointer

**Nama :** Adhitya Warman

**NIM :** 124140007

**Kelas (Kelas Asal) :** RD

Instruksi sederhana :

- ❖ Disarankan untuk edit menggunakan Google Docs agar tidak berantakan,
- ❖ Silahkan mengganti nama modul baik yang ada pada **cover** maupun **header** sesuai dengan materi praktikum,
- ❖ Gunakan text styling seperti Heading 1, Normal Text yang telah terformat, atau text style lainnya untuk menjaga estetika laporan,
- ❖ Gunakan Syntax Highlighter untuk merapikan kode yang sudah anda buat ke dalam laporan.

## Soal/Pertanyaan:

### 1. Prosedur Jumlah Bilangan Ganjil

Objektif: Implementasi prosedur void untuk menghitung jumlah bilangan ganjil dari 1 sampai y.

Spesifikasi:

- Subprogram memiliki parameter \*x (pointer, untuk menyimpan hasil).
- Nilai y = 20.
- Gunakan loop untuk menjumlahkan hanya bilangan ganjil.

Parameter:

- \*x → pointer yang menyimpan hasil penjumlahan.
- y → input dari pengguna, batas atas perhitungan.

Output:

Nilai \*x yang berisi hasil penjumlahan bilangan ganjil dari 1 sampai y.

Contoh:

Input y = 10

Hasil = 25 // (1 + 3 + 5 + 7 + 9)

### 2. Operasi Matematika dengan Pointer

Objektif: Implementasi persamaan matematika menggunakan pointer dengan variasi rumus.

Spesifikasi:

- Sebuah sub program memiliki parameter \*x, y, dan \*z.
- Mula-mula nilai \*x = 0 dan \*z = 1.
- Nilai y = 20.
- Hitunglah \*x dan \*z dengan rumus berikut (gunakan for loop):

Rumus Perhitungan:

- $*x = \sum(n^2)$  untuk n = 1 sampai y → (penjumlahan kuadrat bilangan)

- $*z = \prod (2n - 1)$  untuk  $n = 1$  sampai  $y \rightarrow$  (perkalian bilangan ganjil)

Parameter:

- $*x \rightarrow$  pointer, nilai awal 0
- $y \rightarrow 20$
- $*z \rightarrow$  pointer, nilai awal 1

Output:

Nilai akhir  $*x$  dan  $*z$  setelah perhitungan.

Contoh:

Input  $y = 4$

$$*x = 1^2 + 2^2 + 3^2 + 4^2 = 30$$

$$*z = (1) \times (3) \times (5) \times (7) = 105$$

### 3. Tukar Elemen Array Menggunakan Pointer

Objektif: Latihan manipulasi array dengan pointer.

Spesifikasi:

- Buatlah program C++ yang:
- Meminta pengguna memasukkan ukuran array  $n$ .
- Meminta pengguna mengisi elemen array.
- Gunakan pointer untuk menukar elemen pertama dengan terakhir, serta kedua dengan kedua terakhir, dan seterusnya (efeknya array jadi terbalik).
- Tidak boleh menggunakan indeks array secara langsung ( $arr[i]$ ), hanya pointer aritmatika ( $*(ptr + i)$ ).

Output yang Diharapkan:

Input  $n = 5$

Masukkan elemen: 10 20 30 40 50

Array sebelum dibalik: 10 20 30 40 50

Array sesudah dibalik: 50 40 30 20 10

## Dasar Teori

Pointer dalam bahasa C++ adalah variabel khusus yang menyimpan alamat memori dari variabel lain. Dengan pointer, kita tidak hanya mengakses nilai suatu variabel, tetapi juga bisa mengubah nilai tersebut secara langsung melalui alamat memorinya. Konsep ini menjadi dasar penting dalam pemrograman tingkat lanjut karena memungkinkan pengelolaan memori yang lebih efisien.

Deklarasi pointer menggunakan tanda bintang (\*). Misalnya:

```
int *p;
```

Kode di atas menyatakan bahwa `p` adalah pointer yang dapat menyimpan alamat dari variabel bertipe `int`. Untuk menyimpan alamat suatu variabel, digunakan operator `&` (address-of), sedangkan untuk mengakses nilai yang ditunjuk pointer digunakan operator `*` (dereference).

Contoh sederhana:

```
int x = 10;
int *p = &x;    // p menyimpan alamat x
cout << *p;     // menampilkan isi variabel x yaitu 10
```

Dalam konteks praktikum ini, pointer digunakan untuk:

1. Menyimpan hasil perhitungan pada parameter fungsi (contoh: penjumlahan bilangan ganjil).
2. Melakukan operasi matematis kompleks, seperti penjumlahan kuadrat bilangan dan perkalian bilangan ganjil.
3. Melakukan manipulasi array tanpa menggunakan indeks langsung, melainkan dengan aritmetika pointer.

Melalui penggunaan pointer, program menjadi lebih fleksibel dan dapat mengakses data secara langsung di memori. Namun, penggunaan pointer harus hati-hati, karena kesalahan dapat menyebabkan error serius seperti **dangling pointer** atau **memory leak**.

## Link repl.it Source Code

1. <https://onlinegdb.com/x-6weGCrn>
2. <https://onlinegdb.com/Am1RvX9BM>
3. <https://onlinegdb.com/KsLS9hEHg>

## Source Code

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

// Prosedur untuk menghitung jumlah bilangan ganjil dari 1 sampai y
// Param: *x -> pointer untuk menyimpan hasil
//        y -> batas atas bilangan
void jumlahGanjil(int *x, int y) {
    *x = 0; // inisialisasi hasil penjumlahan ke 0

    // loop dari 1 sampai y
    for (int i = 1; i <= y; i++) {
        if (i % 2 != 0) { // cek apakah bilangan ganjil
            *x += i;      // jika ganjil, tambahkan ke hasil (*x)
        }
    }
}

int main() {
    int y, hasil;

    // minta input batas atas (y) dari pengguna
    cout << "Masukkan nilai y: ";
    cin >> y;

    // panggil prosedur jumlahGanjil, kirim alamat variabel hasil
    jumlahGanjil(&hasil, y);

    // tampilkan hasil penjumlahan bilangan ganjil
    cout << "Hasil penjumlahan bilangan ganjil 1 sampai "
         << y << " adalah: " << hasil << endl;

    return 0; // program selesai
}

2. #include <iostream>
using namespace std;

// Prosedur untuk menghitung:
// *x = jumlah kuadrat bilangan dari 1 sampai y
// *z = hasil perkalian bilangan ganjil dari 1 sampai y
// Param:
//   *x -> pointer untuk menyimpan hasil penjumlahan kuadrat
//   y -> batas perhitungan
//   *z -> pointer untuk menyimpan hasil perkalian bilangan ganjil
void operasiMatematika(int *x, int y, long long *z) {
    *x = 0; // inisialisasi hasil penjumlahan = 0
```

```
*z = 1;      // inisialisasi hasil perkalian = 1

// perulangan dari n = 1 sampai y
for (int n = 1; n <= y; n++) {
    *x += n * n;      // tambahkan kuadrat n ke *x
    *z *= (2 * n - 1); // kalikan bilangan ganjil (2n-1) ke *z
}

int main() {
    int y, hasilX;
    long long hasilZ; // pakai long long agar hasil perkalian besar bisa ditampung

    // minta input dari user
    cout << "Masukkan nilai y: ";
    cin >> y;

    // panggil prosedur dengan mengirim alamat variabel hasil
    operasiMatematika(&hasilX, y, &hasilZ);

    // tampilkan hasil
    cout << "Hasil penjumlahan kuadrat (x): " << hasilX << endl;
    cout << "Hasil perkalian bilangan ganjil (z): " << hasilZ << endl;

    return 0; // program selesai
}
```

3. **#include <iostream>**  
**using namespace std;**

```
int main() {
    int n;
    cout << "Masukkan ukuran array: ";
    cin >> n; // input jumlah elemen array

    int arr[n];      // deklarasi array dengan ukuran n
    int *ptr = arr;   // pointer menunjuk ke array (alamat elemen pertama)

    // Input elemen array
    cout << "Masukkan elemen array: ";
    for (int i = 0; i < n; i++) {
        cin >> *(ptr + i); // input menggunakan pointer, bukan arr[i]
    }

    // Tampilkan array sebelum dibalik
    cout << "Array sebelum dibalik: ";
    for (int i = 0; i < n; i++) {
        cout << *(ptr + i) << " "; // akses elemen dengan pointer
    }
}
```

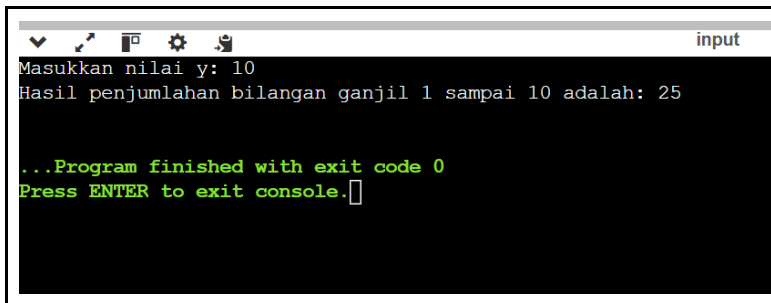
```
cout << endl;

// Proses membalik array dengan pointer
for (int i = 0; i < n / 2; i++) {
    // tukar elemen pertama dengan terakhir, kedua dengan kedua terakhir, dst.
    int temp = *(ptr + i);           // simpan elemen ke-i
    *(ptr + i) = *(ptr + (n - 1 - i)); // ganti dengan elemen dari belakang
    *(ptr + (n - 1 - i)) = temp;     // isi elemen belakang dengan temp
}

// Tampilkan array setelah dibalik
cout << "Array sesudah dibalik: ";
for (int i = 0; i < n; i++) {
    cout << *(ptr + i) << " ";
}
cout << endl;

return 0; // program selesai
}
```

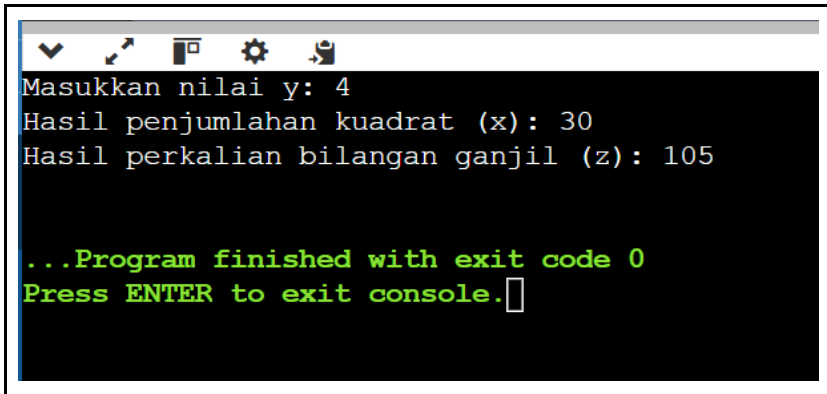
## Dokumentasi Hasil Running



```
input
Masukkan nilai y: 10
Hasil penjumlahan bilangan ganjil 1 sampai 10 adalah: 25

...Program finished with exit code 0
Press ENTER to exit console.
```

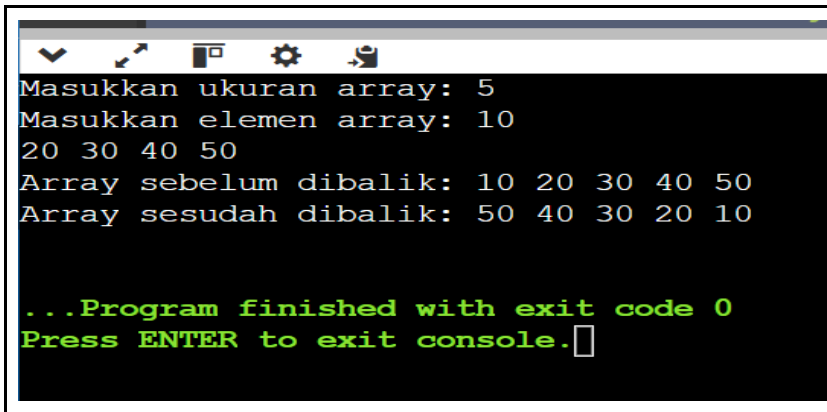
**Gambar 1.** Hasil Running yang menjumlahkan semua bilangan ganjil dari 1 sampai ke y



```
Masukkan nilai y: 4
Hasil penjumlahan kuadrat (x): 30
Hasil perkalian bilangan ganjil (z): 105

...Program finished with exit code 0
Press ENTER to exit console.
```

**Gambar 2.** Hasil Running yang menjumlahkan hasil kuadrat dan menampilkan hasil perkalian dari beberapa logic pada nilai y yang di inputkan



```
Masukkan ukuran array: 5
Masukkan elemen array: 10
20 30 40 50
Array sebelum dibalik: 10 20 30 40 50
Array sesudah dibalik: 50 40 30 20 10

...Program finished with exit code 0
Press ENTER to exit console.
```

**Gambar 3.** Hasil Running yang menampilkan inputan yang di inputkan decara berurutan dari awal dan kebalikkannya

## Contoh Penggunaan:

### 1. Prosedur Jumlah Bilangan Ganjil

Pada program jumlah bilangan ganjil, pengguna diminta untuk memasukkan sebuah nilai batas atas y. Program kemudian akan menghitung jumlah seluruh bilangan ganjil dari angka 1 hingga y. Misalnya jika pengguna memasukkan nilai y = 10, maka program akan menampilkan hasil penjumlahan bilangan ganjil yaitu  $1 + 3 + 5 + 7 + 9 = 25$ . Perhitungan ini dilakukan dengan bantuan pointer yang menyimpan hasil penjumlahan dan kemudian ditampilkan kembali ke layar.

### 2. Operasi Matematika dengan Pointer

Pada program operasi matematika, pengguna juga diminta untuk memasukkan nilai batas atas y. Program akan menghitung dua hasil sekaligus, yaitu jumlah kuadrat bilangan dari 1 sampai y serta hasil



perkalian semua bilangan ganjil dari 1 sampai  $2y - 1$ . Sebagai contoh, jika pengguna memasukkan  $y = 4$ , maka hasil penjumlahan kuadrat adalah  $1^2 + 2^2 + 3^2 + 4^2 = 30$ , dan hasil perkalian bilangan ganjil adalah  $1 \times 3 \times 5 \times 7 = 105$ . Nilai akhir ini disimpan dalam variabel melalui pointer dan kemudian ditampilkan di layar.

### 3. Tukar Elemen Array Menggunakan Pointer

program manipulasi array, pengguna diminta untuk memasukkan ukuran array  $n$ , kemudian mengisi elemen-elemen array tersebut. Program akan menampilkan array sebelum dibalik, lalu menukar urutan elemen sehingga array menjadi terbalik dengan memanfaatkan aritmetika pointer, tanpa menggunakan indeks array secara langsung. Misalnya jika pengguna memasukkan ukuran  $n = 5$  dengan elemen 10 20 30 40 50, maka output program akan menampilkan “Array sebelum dibalik: 10 20 30 40 50” dan “Array sesudah dibalik: 50 40 30 20 10”. Proses pertukaran dilakukan dengan menukar elemen pertama dengan terakhir, kedua dengan kedua terakhir, hingga seluruh array terbalik.

## Link GitHub/GDB Online:

Link GitHub : <https://github.com/AW2606/LAPRAK-ASD-RD-007.git>

## Referensi

Deitel, H. M., & Deitel, P. J. (2017). *C++ How to Program* (10th ed.). Pearson Education.

Stroustrup, B. (2013). *The C++ Programming Language* (4th ed.). Addison-Wesley.

Malik, D. S. (2018). *C++ Programming: From Problem Analysis to Program Design* (8th ed.). Cengage Learning.

GeeksforGeeks. (2024). *Pointers in C++*. Diakses dari: <https://www.geeksforgeeks.org/cpp-pointers>

TutorialsPoint. (2024). *C++ Pointers*. Diakses dari: [https://www.tutorialspoint.com/cplusplus/cpp\\_pointers.htm](https://www.tutorialspoint.com/cplusplus/cpp_pointers.htm)

ChatGPT