

Laporan Praktikum

Algoritma dan Struktur Data

Ganjil 2025/2026

Program Studi Teknik Informatika
Institut Teknologi Sumatera



Modul : Pointer

Nama : Christopher Leon Saputra

NIM : 124140097

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 adalah variabel yang menyimpan alamat memori dari variabel lain, bukan nilainya langsung. Dengan pointer, kita dapat mengakses atau mengubah nilai variabel melalui alamat tersebut

Perbedaan pointer dan variabel biasa adalah:

- Variabel biasa menyimpan data secara langsung.
- Pointer menyimpan alamat tempat data itu berada.

Fungsi Pointer:

1. Mengubah nilai asli variabel melalui fungsi.
2. Efisiensi memori, karena cukup mengirimkan alamat saja.
3. Menjadi dasar untuk struktur data yang lebih kompleks.

Cara penggunaan pointer ada dua, yaitu:

1. Operator & (reference) untuk mendapatkan alamat variabel.
2. Operator * (dereference) untuk mengakses atau mengubah nilai yang di tunjuk.

Source Code

Gunakan [Syntax Highlighter](#) untuk merapikan source code yang dipindahkan dari text editor anda ke dokumen ini. Jangan lupa untuk memberikan penjelasan yang rinci pada code anda, dengan menggunakan comment

SOAL 1

```
1. #include <iostream>
2. using namespace std;
3.
4. // Fungsi untuk menghitung jumlah bilangan ganjil 1 sampai y
5. // Hasil disimpan di pointer *x
6. void jumlahGanjil(int *x, int y) {
7.     *x = 0;
8.     for(int i = 1; i <= y; i+=2) // loop hanya bilangan ganjil
9.         *x += i;                // tambahkan ke hasil
10. }
11.
12. int main() {
13.     int y;
14.     cout << "Masukkan nilai y: ";
```

```
15.     cin >> y;
16.
17.     int hasil = 0;
18.     jumlahGanjil(&hasil, y);           // panggil fungsi dengan alamat hasil
19.     cout << "Hasil: " << hasil;
20.     return 0;
21. }
```

SOAL 2

```
1. #include <iostream>
2. using namespace std;
3.
4. // Fungsi operasi: hitung jumlah kuadrat dan hasil kali bilangan ganjil
5. void operasi(int *x, int y, int *z) {
6.     *x = 0;           // inisialisasi jumlah kuadrat
7.     *z = 1;           // inisialisasi hasil kali ganjil
8.     for(int n = 1; n <= y; n++) {
9.         *x += n * n;   // jumlah kuadrat dari 1^2 + 2^2 + ... + y^2
10.        *z *= (2 * n - 1); // kali semua bilangan ganjil: 1 * 3 * 5 * ...
11.    }
12. }
13.
14. int main() {
15.     int x, y, z;
16.     cout << "y: "; cin >> y;
17.
18.     operasi(&x, y, &z);           // panggil fungsi dengan pointer
19.
20.     cout << "x: " << x << endl;
21.     cout << "z: " << z << endl;
22.     return 0;
23. }
```

SOAL 3

```
1. #include <iostream>
2. using namespace std;
3.
4. void balikArray(int *ptr, int n) {
5.     // ptr menunjuk ke alamat array
6.     // *(ptr + i) artinya ambil elemen ke-i
7.     for (int i = 0; i < n / 2; ++i) {
8.         int tmp = *(ptr + i);           // simpan elemen ke-i
9.         *(ptr + i) = *(ptr + (n - 1 - i)); // tukar dengan elemen dari
        belakang
10.        *(ptr + (n - 1 - i)) = tmp;
11.    }
12. }
13.
14. void printArray(int *ptr, int n) {
15.     // akses elemen array lewat pointer
16.     for (int i = 0; i < n; ++i) {
17.         cout << *(ptr + i);           // *(ptr + i) sama dengan arr[i]
18.         if (i < n - 1) cout << " ";
19.     }
}
```

```
20.     cout << '\n';
21.}
22.
23.int main() {
24.     const int maxN = 1000;
25.     int arr[maxN];
26.     int n;
27.
28.     cout << "Masukkan ukuran array n: ";
29.     cin >> n;
30.     if (n <= 0 || n > maxN) {
31.         cout << "Nilai n tidak valid. Pakai 1.." << maxN << '\n';
32.         return 0;
33.     }
34.
35.     cout << "Masukkan " << n << " elemen: ";
36.     for (int i = 0; i < n; ++i) {
37.         cin >> *(arr + i); // arr digunakan sebagai pointer ke elemen pertama
38.     }
39.
40.     cout << "Array sebelum dibalik: ";
41.     printArray(arr, n);
42.
43.     balikArray(arr, n); // kirim alamat array ke fungsi
44.
45.     cout << "Array setelah dibalik: ";
46.     printArray(arr, n);
47.
48.     return 0;
49.}
```

Dokumentasi Hasil Running

Berikan penjelasan tentang gambar hasil running tersebut. Gunakan Tabel 1x2 dengan border 0pt agar gambar yang dilampirkan rapih. Jika sulit, silakan copy tabel di bawah ini,

NOMOR 1

Jika $y = 10$, fungsi akan menambahkan semua bilangan ganjil dari 1 sampai 10. Hasil penjumlahan adalah $1 + 3 + 5 + 7 + 9 = 25$. Maka program akan menampilkan output Hasil: 25.



```
Masukkan nilai y: 10
Hasil: 25

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

Gambar 1.1 Running code nomor 1

NOMOR 2

Jika $y = 4$, maka program menghitung jumlah kuadrat dari 1 sampai 4. Hasilnya yaitu $1^2 + 2^2 + 3^2 + 4^2 = 30$. Untuk perkalian bilangan ganjil, dihitung $(1 \times 3 \times 5 \times 7) = 105$. Jadi output yang ditampilkan adalah $x: 30$ dan $z: 105$.



```
y: 4
x: 30
z: 105
```

Gambar 1.2 Running code nomor 2

NOMOR 3

Jika $n = 5$ dengan elemen array `10 20 30 40 50`, program mula-mula menampilkan array asli yaitu `10 20 30 40 50`. Lalu fungsi `balikArray` bekerja dengan pointer untuk menukar elemen pertama dengan terakhir, kedua dengan keempat, dan seterusnya. Proses ini membalik urutan array tanpa memakai indeks langsung, hanya lewat aritmatika pointer. Hasil akhirnya array berubah menjadi `50 40 30 20 10` dan ditampilkan sebagai output.

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

Gambar 1.3 Running code nomor 3

Link GitHub/GDB Online:

1. <https://github.com/leec1za/TUGASPRAKTIKUMASD.git>

Referensi

1. <https://praktikum-asd-rd.vercel.app/pointer>