

Laporan Praktikum

Algoritma dan Struktur Data

Ganjil 2025/2026

Program Studi Teknik Informatika

Institut Teknologi Sumatera



Modul : Pointer

Nama : Kenzie Sahasika Tariana

NIM : 124140103

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

Dasar Teori Materi Pointer

Pointer adalah variabel khusus yang menyimpan **alamat memori** dari variabel lain, bukan nilai langsungnya. Dengan pointer, program dapat mengakses dan memanipulasi data secara lebih fleksibel.

Pada pointer kita dapat melakukan deklarasi variabel dengan menggunakan tanda ***** pada tipe data yang ingin kita jadikan sebagai alamat.

contoh:

```
int *ptr;
```

Lalu dalam pointer kita juga dapat melakukan inisialisasi pada pointer jika melakukan pengisian alamat variabel dengan menggunakan operator **&**. Hal ini dilakukan untuk mengubah alamat asal/alamat awal pointer dengan nilai baru sesuai dengan keinginan pengguna.

Contoh :

```
int x = 10;  
int *ptr = &x;
```

Selanjutnya untuk mengakses nilai melalui variabel dari pointer yang ada, kita dapat menggunakan tanda (*****) pada awal variabel saat melakukan pemanggilan.

Contoh :

```
cout << *ptr; // menampilkan nilai x  
*ptr = 20;    // mengubah nilai x
```

Dalam pemrograman kegunaan utama dari pointer adalah untuk melakukan efisiensi memori, dengan passing by reference pada fungsi, dan pengelolaan memori dinamis. Dengan menggunakan pointer kita dengan mudah dapat mengubah nilai suatu variabel, agar sesuai dengan kebutuhan. Pengguna hanya perlu melakukan perubahan alamat dari variabel yang dipilih dengan melakukan langkah seperti contoh berikut :

Contoh singkat:

```
int a = 5;  
  
int *p = &a;  
*p = 15; // nilai a berubah menjadi 15
```

Namun beberapa kesalahan yang sering terjadi/kekeliruan dalam penggunaan pointer adalah salah dalam mengakses alamat, sehingga memanggil alamat variabel yang tidak valid dan dapat mengakibatkan pointer tidak dapat digunakan atau dapat mengakibatkan error pada pemanggilan pointer.

Source Code

Nomor 1

```
1. #include <iostream>
2. using namespace std;
3.
4.
5. //membuat subprogram void dan menggunakan pointer untuk menyimpan
   parameter dari variabel x
6. void jmlhBilGanjil(int *x, int y) {
7.     *x = 0;
8.     //menggunakan for untuk melakukan penjumlahan sesuai dengan nilai
   yang akan di masukkan oleh pengguna
9.     for (int i = 1; i <= y; i++) {
10.         if (i % 2 != 0) {
11.             *x += i;
12.         }
13.     }
14. }
15.
16. int main() {
17.     int hasil, y;
18.
19.
20.     //memberikan keluaran untuk meminta input dari pengguna untuk nilai
   y
21.     cout << "Masukkan nilai y: ";
22.     cin >> y;
23.
24.
25.     //mendeklarasikan variabel yang akan di gunakan ke dalam pointer
   yang telah di buat sebelumnya
26.     jmlhBilGanjil(&hasil, y);
27.
28.
29.     //
```

```
30.     cout << "Jumlah bilangan ganjil dari 1 sampai " << y << " = " <<
        hasil << endl;
31.     return 0;
32. }
```

Nomor 2

```
1. #include <iostream>
2. using namespace std;
3.
4. //membuat fungsi void dan menggunakan pointer untuk menyimpan nilai dari variabel
   x dan z
5. void operasiMatematika(int *x, int y, float *z) {
6.     //mendeklarasikan nilai untuk alamat pointer x dan z
7.     *x = 0;
8.     *z = 1;
9.     // menggunakan for untuk melakukan perhitungan dari penjumlahan x dan z
   sesuai dengan ketentuannya yakni menghitung dari 1 hingga y yang akan diinputkan
   oleh pengguna
10.    for (int n = 1; n <= y; n++) {
11.        *x += (n * n);
12.        *z *= (2 * n - 1);
13.    }
14. }
15.
16.
17. int main() {
18.     //mendeklarasikan beberapa variabel yang akan di gunakan pada bagian
   selanjutnya
19.     int hasilx, y;
20.     float hasilz;
21.
22.     //memberikan keluaran untuk meminta input nilai y dari pengguna
23.     cout << "Masukkan nilai y: ";
24.     cin >> y;
25.
26.     //melakukan pemanggilan pointer dengan variabel yang di gunakan
27.     operasiMatematika(&hasilx, y, &hasilz);
28.
29.     //memberikan keluaran hasil dari perhitungan yang sesuai dengan ketentuan dan
   nilaiy yang telah di masukkanoleh pengguna
30.     cout << "Hasil penjumlahan kuadrat (*x) = " << hasilx << endl;
31.     cout << "Hasil perkalian bilangan ganjil (*z) = " << hasilz << endl;
32.
33.
34.     return 0;
35. }
```

Nomor 3

```
1. #include <iostream>
2. using namespace std;
3.
```

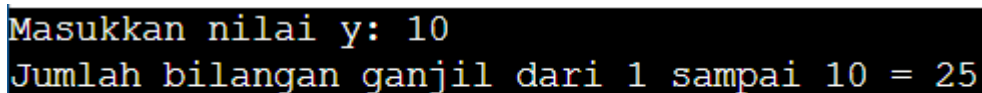
```
4. //membuat fungsi void dan menggunakan pointer untuk variabel yang akan digunakan
   dalam pengoperasian
5. void BalikArray(int *arr, int n) {
6.     //menginisialisasi variabel sebagai penanda awal dan akhir sebagai acuan pada
   saat di lakukan pemutaran
7.     int *ptrb = arr;
8.     int *ptrb = arr + n - 1;
9.     //menggunakan while untuk menjalankan program sesuai kondisi yang diinginkan
   sehingga dapat dilakukan pemutaran elemen yang sesuai dengan urutannya dari awal
   ke akhir dan akhir ke awal
10.    while (ptrb < ptrb) {
11.        int temp = *ptrb;
12.        *ptrb = *ptrb;
13.        *ptrb = temp;
14.        ptrb++;
15.        ptrb--;
16.    }
17.}
18.
19.
20.int main() {
21.
22.    //mendeklarasikan variabel yang akan digunakan dan membuat 1 array untuk
   menampung jumlah elemen yang akan dimasukkan sesuai dengan keinginan pengguna
23.    const int MAX = 100; // kapasitas maksimum array
24.    int arr[MAX];
25.    int n;
26.
27.    //meminta input ukuran array yang diinginkan oleh pengguna untuk menampung
   input elemen dari pengguna
28.    cout << "Masukkan ukuran array (maksimal 100): ";
29.    cin >> n;
30.
31.    //menggunakan if untuk mendeteksi jika pengguna memasukkan ukuran array
   melebihi batas maksimal yang telah di tentukan untuk ukuran arraynya
32.    if (n > MAX) {
33.        cout << "Ukuran array terlalu besar! Maksimal " << MAX << endl;
34.        return 1;
35.    }
36.
37.    //memberikan keluaran yang meminta pengguna untuk memasukkan elemen yang
   diinginkan sesuai dengan jumlah ukuran array sebelumnya telah diberikan
   menggunakan for
38.    cout << "Masukkan elemen array: ";
39.    for (int i = 0; i < n; i++) {
40.        cin >> arr[i];
41.    }
42.
43.    //menampilkan urutan elemen sesuai dengan jumlah array yang telah ditentukan
   sebelum dilakukan pemutaran uruan
44.    cout << "Array sebelum dibalik: ";
45.    for (int i = 0; i < n; i++) {
46.        cout << arr[i] << " ";
47.    }
48.    cout << endl;
```

```
49.    //melakukan pemanggilan fungsi void yang telah di buat pada awal program
      untuk melakukan pembaikan urutan array yang telah di berikan sebelumnya
50.    BalikArray(arr, n);
51.
52.    //menampilkan elemen sesuai array yang ada namun dengan urutan yang telah di
      balik menggunakan fungsi void dari program yang telah di buat di awal
53.    cout << "Array setelah dibalik: ";
54.    for (int i = 0; i < n; i++) {
55.        cout << arr[i] << " ";
56.    }
57.    cout << endl;
58.
59.    return 0;
60.}
```

Dokumentasi Hasil Running

NOMOR 1

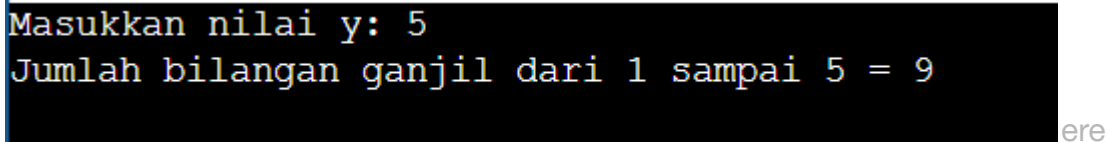
Pada kode program dari soal pertama, setelah di running mendapatkan hasil yang sama dengan contoh yang diberikan dari soal. Dimana saat diinputkan nilai 10 menghasilkan input 25, ini menandakan program berjalan sesuai dengan instruksi yang telah diberikan. Program menghitung/menjumlahkan hanya angka ganjil dalam deret 1 sampai dengan n. Dalam running pertama ini kita melihat penjumlahan angka ganjil yang terdapat dari urutan baris 1 sampai 10 ($1+3+5+7+9$) dan menghasilkan output 25.



```
Masukkan nilai y: 10
Jumlah bilangan ganjil dari 1 sampai 10 = 25
```

Gambar 1.1 Running code nomor 1 dengan input sesuai contoh

Dan ketika program di running lalu pada bagian input di ganti dengan angka 5 maka akan memberikan hasil output 9. Karena program melakukan penjumlahan hanya angka ganjil dari 1 hingga 5 yakni $(1+3+5)$ dan hasilnya 9.

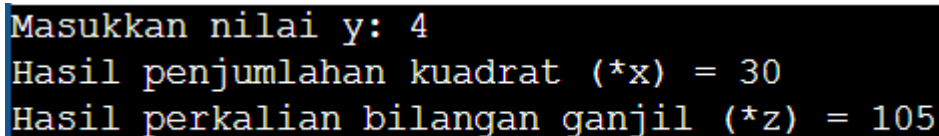


```
Masukkan nilai y: 5
Jumlah bilangan ganjil dari 1 sampai 5 = 9
```

Gambar 1.2 Running code nomor 1 dengan input 5

NOMOR 2

Pada kode program nomor 2 ini kita dapat melihat bahwa program berjalan dengan baik sesuai dengan ketentuan, di dalam program terdapat 3 variabel yakni x,y,dan z. Untuk variabel x dan z adalah pointer yang melakukan 2 tugas yang berbeda yakni, pada variabel x akan melakukan penjumlahan bilangan berpangkat seperti pada contohnya variabel x akan melakukan penjumlahan kuadrat bilangan dari 1 sampai nilai y sesuai input pengguna dan untuk variabel z akan melakukan perkalian bilangan ganjil dari 1 sampai nilai y sesuai input pengguna. Dan pada running pertama ini kita menggunakan nilai 4 sesuai dengan contoh pada soal, sehingga memberikan input yang sama dengan soal. Untuk variabel diberikan nilai 4 sehingga untuk variabel x memberikan output 30 karena melakukan penjumlahan bilangan berpangkat ($*x = 1^2 + 2^2 + 3^2 + 4^2$) yang memiliki hasil 30. Pada variabel z program melakukan pengkalian 4 bilangan ganjil, yang dimana di ambil dari urutan awal yakni ($*z = 1 \times 3 \times 5 \times 7$) dan memberikan hasil 105. Berbeda dengan kode program pada nomor 1 yang menjumlahkan angka ganjil yang terdapat di dalam urutan 1 sampai y,pada kode program nomor 2 ini untuk variabel z pengambilan nilainya dia akan menghitung angka ganjil dari 1 akan menjadi angka pertama dan 7 akan menjadi angka ke 4, karena program hanya mencari 4 bilangan ganjil bukan mencari bilangan ganjil yan terdapat di antara angka 1 sampai 4.



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

Gambar 2.1 Running code nomor 2 dengan input sesuai contoh

Dan ketika kita mengganti input menjadi 3 akan memberikan hasil $x = 14$ dan $z = 15$. Karena pada variabel pointer $*x$ perhitungannya menjadi seperti ini ($*x = 1^2 + 2^2 + 3^2$) yang hasilnya 14. Lalu pada variabel pointer $*z$ perhitungannya akan menjadi seperti ini ($*z = 1 \times 3 \times 5$) yang hasilnya adalah 15, sehingga membuktikan bahwa program berjalan sesuai dengan ketentuan yang berikan.

```
Masukkan nilai y: 3
Hasil penjumlahan kuadrat (*x) = 14
Hasil perkalian bilangan ganjil (*z) = 15
```

Gambar 2.2 Running code nomor 2 dengan input 3

NOMOR 3

Pada kode program nomor 3 ini terlihat berjalan dengan baik dan sesuai dengan ketentuan, yakni melakukan pembalikan urutan elemen pada sebuah array. Ketika input yang diberikan sesuai dengan contoh pada soal yakni memiliki array yang berisikan 5 elemen (10 20 30 40 50) lalu array tersebut dipanggil ulang yang memberikan keluaran untuk menunjukkan urutan array sebelum di balik. Lalu berikutnya program memberikan keluaran setelah array tersebut di proses dan memberikan keluaran (50 40 30 20 10), yang dimana kita bisa lihat bahwa urutan dari elemen didalam array tersebut telah diubah/dibalik dengan baik sesuai dengan yang kita inginkan dan ketentuan.

```
Masukkan ukuran array (maksimal 100): 5
Masukkan elemen array: 10 20 30 40 50
Array sebelum dibalik: 10 20 30 40 50
Array setelah dibalik: 50 40 30 20 10
```

Gambar 3.1 Running code nomor 3 dengan input sesuai contoh

Dan ketika kita mengganti inputnya yakni array berukuran 6 dengan isi elemen di dalamnya (60 50 40 30 20 10), output untuk bagian array yang belum di balik sesuai dengan urutan awalnya pada saat di inputkan. Dan selanjutnya setelah dilakukan perubahan urutan/dibalik, isi dari array tersebut menjadi (10 20 30 40 50 60), yang menandakan program berjalan dengan baik serta sesuai dengan ketentuan.

```
Masukkan ukuran array (maksimal 100): 6
Masukkan elemen array: 60 50 40 30 20 10
Array sebelum dibalik: 60 50 40 30 20 10
Array setelah dibalik: 10 20 30 40 50 60
```

Gambar 3.2 Running code nomor 3 dengan input ukuran array 6

Link GitHub/GDB Online:

1. <https://github.com/kenzie781/PRAKTIKUM-ASD-RD-124140103.git>

Referensi

KALAU KALIAN PAKE GEN AI (CHAT GPT, GEMINI, CLAUDE, DLL. KALIAN BISA SHARE LINK PERCAKAPAN LINK GEN AI KALIAN DI SINI SEBAGAI BAHAN REFERENSI CODE KALIAN)