`**LAPORAN PRAKTIKUM**

**ALGORITMA DAN STRUKTUR DATA**



**Disusun oleh**

**NAMA : ROYHAN ABDURROHIM**

**NIM : 32602200018**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS ISLAM SULTAN AGUNG**

**SEMARANG**

**2022**

**HALAMAN PENGESAHAN**

**Laporan Praktikum**

**Algoritma dan Struktur Data**

Disusun Oleh :

Royhan Abdurrohim (32602200018)

Telah disetujui sebagai syarat untuk memenuhi mata kuliah Praktikum Algoritma dan Struktur Data Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

Hari :

Tanggal :

Asisten Praktikum

1. 1.................
2. 2.................
3. 3………....

Semarang, Bulan Tahun

Mengetahui,

Dosen Pengampu Praktikum Laboran

Badieah, ST., M.Kom Werdha Wilubertha H, S.Kom

**KATA PENGANTAR**

Alhamdulillah, puji syukur patut kita hantunkan kepada Allah SWT Karena atas hidayah-Nyalah laporan ini dapat terselesaikan.

Dan tidak lupa pula saya ucapkan terima kasih kepada pihak yang telah banyak membantu dan membimbing saya dalam penyusunan laporan ini. Saya ucapkan terima kasih.

.Saya menyadari bahwa banyak kekurangan dalam penyusunan laporan ini, oleh karena itu saya minta maaf apabila terdapat kesalahan dalam penyusunan laporan ini karena manusia tidak luput dari kesalahan.

Semarang, Bulan Tahun

PENYUSUN / PENULIS

( Tanda Tangan )

Nama

**DAFTAR ISI**

HALAMAN JUDUL i

LEMBAR PENGESAHAN ii

KATA PENGANTAR iii

DAFTAR ISI iv

DAFTAR GAMBAR v

DAFTAR TABEL vi

BAB I PENGENALAN BAHASA C 1

1.1.Tujuan 1

1.2.Dasar Teori 1

1.2.1. 1

1.2.2 2

1.2.3. 3

1.3.Percobaan 6

1.4.Tugas dan Latihan 10

1.5.Kesimpulan 17

DAFTAR PUSTAKA

# **DAFTAR GAMBAR**

[Gambar 1.1 Struktur Sekuensial 8](#_Toc120626153)

[Gambar 1.2 Struktur Seleksi 9](#_Toc120626154)

[Gambar 1.3 Perulangan Bentuk Pertama 10](#_Toc120626155)

[Gambar 1.4 Perulangan Bentuk Kedua 11](#_Toc120626156)

[Gambar 1.5 Flowchart Latihan 1 13](#_Toc120626157)

[Gambar 1.6 Output Latihan 1 Persegi Panjang 13](#_Toc120626158)

[Gambar 1.7 Output Latihan 1 Lingkaran 13](#_Toc120626159)

[Gambar 1.8 Flowchart Latihan 2 15](#_Toc120626160)

[Gambar 1.9 Output Latihan 2 15](#_Toc120626161)

[Gambar 1.10 Output Tugas 1 17](#_Toc120626162)

[Gambar 1.11 Flowchart Tugas 1 18](#_Toc120626163)

[Gambar 1.12 Output Tugas 2 Luas 20](#_Toc120626164)

[Gambar 1.13 Output Tugas 2 Keliling 20](#_Toc120626165)

[Gambar 1.14 Output Tugas 2 Diagonal 21](#_Toc120626166)

[Gambar 1.15 Flowchart Tugas 2 21](#_Toc120626167)

[Gambar 1,16 Output Tugas 3 Luas 23](#_Toc120626168)

[Gambar 1.17 Output Tugas 3 Keliling 23](#_Toc120626169)

[Gambar 1.18 Output Tugas 3 Diagonal 24](#_Toc120626170)

[Gambar 1.19 Flowchart Tugas 3 24](#_Toc120626171)

[Gambar 1.20 Output Tugas 4 27](#_Toc120626172)

[Gambar 1.21 Flowchart Tugas 4 27](#_Toc120626173)

[Gambar 1.22 Output Tugas 5 29](#_Toc120626174)

[Gambar 1.23 Flowchart Tugas 5 30](#_Toc120626175)

**DAFTAR TABEL**

Tabel 1.1 5

Tabel 1.2 5

Tabel 2.1 17

Tabel 2.2 19

Dst

# BAB I

# STRUKTUR DASAR ALGORITMA

1. Tujuan

Adapun tujuan dari diadakannya praktikum ini:

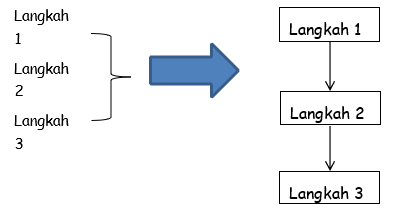
1. Mahasiswa dapat mengetahui dan memahami macam struktur dasar algoritma
2. Mahasiswa dapat mengetahui dan memahami struktur sekuensial
3. Mahasiswa dapat mengetahui dan memahami struktur seleksi
4. Mahasiswa dapat mengetahui dan memahami struktur perulangan
5. Dasar Teori

### Sekuensial

Suatu pemrograman tidaklah terlepas dari algoritma sebagai acuan untuk membuat program. Pemrograman sekuensial merupakan runtunan (sekuensial / *sequence*) satu atau lebih instruksi, yang berarti bahwa:

* Tiap instruksi dikerjakan satu per satu
* Tiap instruksi dilaksanakan tepat satu kali; tidak ada instruksi yang di ulang
* Urutan instruksi yang dilaksanakan pemroses sama dengan urutan instruksi sebagaimana yang tertulis di dalam teks algoritmanya.
* Akhir dari instruksi terakhir merupakan akhir algoritma.

Pada struktur sekuensial, langkah-langkah yang dilakukan dalam algoritma diproses secara berurutan, sebagaima diperlihatkan pada gambar berikut:



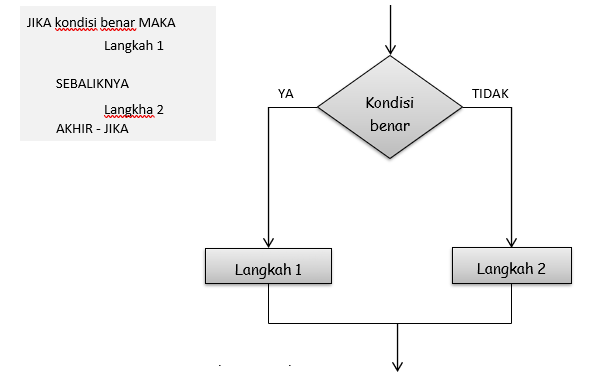
Gambar 1.1 Struktur Sekuensial

Pada contoh tersebut, langkah 1 merupakan langkah yang akan dijalankan pertama kali. Setelah itu, langkah 2 dikerjakan dan diikuti dengan langkah 3.

### Seleksi

Seleksi digunakan unuk mengarahkan pencabangan aliran eksekusi program. Eksekusi program diarahkan sesuai kondisi yang mengendalikannya. Aliran seleksi dalam C++ dinyatakan dalam bentuk if dan switch. Aliran ini digunakan untuk menentukan alur jalannya eksekusi program. Alur eksekusi program ditentukan dari nilai kebenaran kondisi yang di tentukan.

Struktur seleksi menyatakan pemilihan langkah yang didasarkan oleh suatu kondisi (pengambilan keputusan). Gambar 1.2 memperlihatkan diagram alir struktur seleksi yang melibatkan dua alternatif. Dalam hal ini, simbol belah ketupat digunakan untuk mewakili langkah pengambilan keputusan.



Gambar 1.2 Struktur Seleksi

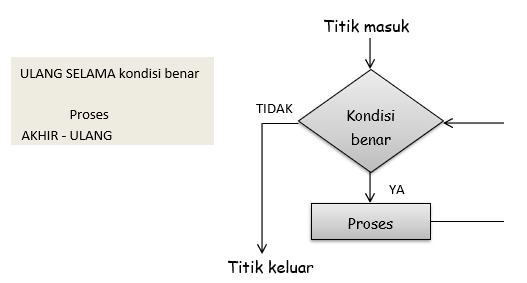
Pada struktur tersebut, langkah 1 hanya akan dijalankan jika kondisi bernilai benar, sedangkan langkah 2 hanya akan dijalankan jika kondisi bernilai salah.

### Pengulangan

Pengulangan adalah suatu tindakan melakukan hal yang sama berulang-ulang kali. Penting untuk dilakukan suatu pengulangan atau perulangan dalam pembuatan suatu program dengan tujuan untuk mengefisienkan kerja program ataupun mengefisienkan listing dari coding dari program tersebut, karena programer tidak perlu mengulang proses yang sama. Untuk perulangan pada c++ dapat dilakukan dengan menggunakan pernyataan for, while dan do-while.

Pengulangan menyatakan suatu tindakan atau langkah yang dijalankan beberapa kali. Struktur pengulangan menyatakan perwujudan keadaan seperti itu. Sebagai contoh, jika anda ingin menampilkan 10 tulisan “SELAMAT BELAJAR”, Anda bisa menuliskannya dengan menggunakan struktur sekuensial. Hal itu berarti Anda memberikan 10 instruksi untuk menuliskan kesepuluh tulisan tersebut. Cara seperti itu memang praktis untuk jumlah pengulangan yang sedikit (misalnya 2 atau 3 pengulangan), tetapi tidak cocok untuk jumlah yang besar. Agar lebih praktis, Anda bisa menggunakan struktur pengulangan.

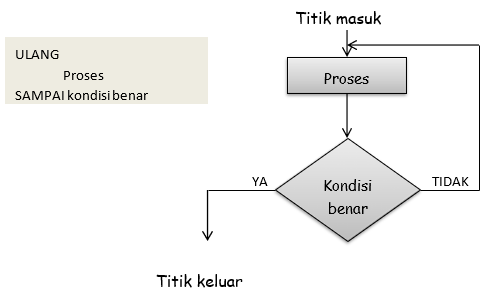
**Struktur pengulangan pertama**



Gambar 1.3 Perulangan Bentuk Pertama

Pada struktur tersebut, proses dapat berupa satu atau beberapa langkah. Pada bentuk ini, ada kemungkinan proses tidak dijalankan sama sekali sekiranya di awal kondisi bernilai salah. Diagram alir pada Gambar 1.3 menunjukkan bahwa sebelum proses dijalankan pertama kali, kondisi diuji terlebih dahulu. Sekiranya kondisi bernilai benar maka proses dijalankan, kemudian kondisi diuji lagi. Sepanjang kondisi masih bernilai benar proses akan tetap dijalankan. Namun, begitu kondisi bernilai salah maka pengulangan berakhir.

**Struktur pengulangan kedua**



Gambar 1.4 Perulangan Bentuk Kedua

Pada bentuk kedua, proses paling tidak dijalankan sekali.

### Flowchart

Flowchart adalah urutan proses kegiatan yang digambarkan dalam bentuk simbol. *Flowchart* juga didefinisikan sebagai diagram yang menyatakan aliran proses dengan menggunakan anotasianotasi.

Beberapa simbol yang digunakan didalam *flowchart* adalah sebagai berikut:

1. **(Terminator**) menyatakan titik awal atau titik akhir diagram alir.
2. (**Proses)** menyatakan sebarang proses, misal untuk menyatakan suatu operasi aritmatika.
3. (**Input/output** atau kadang disebut data) digunakan untuk menyatakan operasi pemasukan data atau penampilann data.
4. **(Keputusan**) dilakukan untuk melakukan pengambilan keputusan. Dalam hal ini, yang ada dalam simbol ini berupa suatu pertanyaan yang jawabannya berupa dua kemungkinan, yaitu YA atau TIDAK.
5. (**Konektor**) digunakan untuk menghubungkan ke berbagai bagian dalam diagram alir. (Kadir, 2010)
6. Latihan
7. Contoh seleksi
8. Kode Program

#include <iostream>

using namespace std;

main()

{

cout << "Nama: Royhan Abdurrohim\n";

cout << "NIM: 32602200018\n\n";

char jenis;

int panjang, lebar, luas, jari;

float luaslingkar;

cout << "Pilih Persegi panjang<P> atau Lingkaran<L>: ";

cin >> jenis;

if (jenis == 'P' || jenis == 'p')

{

cout << "panjang = ";

cin >> panjang;

cout << "lebar = ";

cin >> lebar;

luas = panjang \* lebar;

cout << "Luas = " << luas << endl;

}

else if (jenis == 'L' || jenis == 'l')

{

cout << "jari-jari = ";

cin >> jari;

luaslingkar = 3.14 \* jari \* jari;

cout << "Luas = " << luaslingkar << endl;

}

else

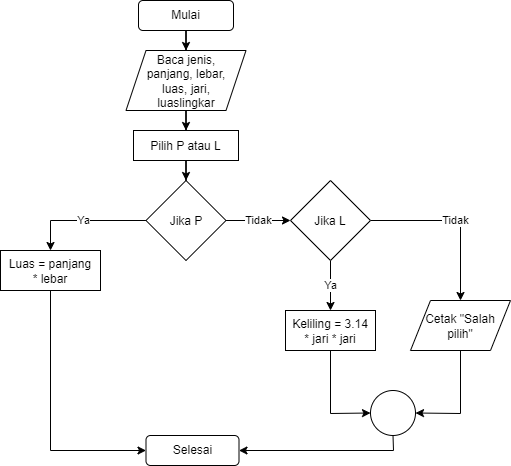
}

cout << "salah pilih" << endl;

}

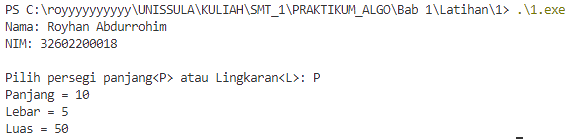
}

1. Flowchart

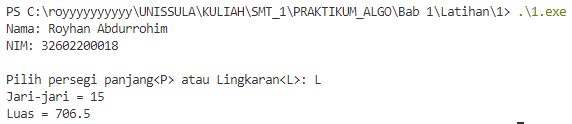


Gambar 1.5 Flowchart Latihan 1

1. Output



Gambar 1.6 Output Latihan 1 Persegi Panjang



Gambar 1.7 Output Latihan 1 Lingkaran

1. Penjelasan

Gambar 1.6 dan gambar 1.7 menunjukkan contoh percabangan

branching (if) antara pilihan P untuk perhitungan luas persegi dan L untuk perhitungan luas lingkaran. Dalam kode perhitungan tersebut menggunakan deklarasi variable dengan tipe data integer, float dan char.

Variabel dengan tipe data char merupakan variabel yang akan menampung pilihan dari masukan pengguna. Jika kita ingin menghitung luas persegi kita masukkan pilihan P. Sehingga output yang akan keluar adalah perhitungan persegi dengan integer panjang dan lebar.

Namun, apabila kita ingin menghitung luas lingkaran, kita masukkan pilihan L. Sehingga output yang keluar adalah perhitungan luas lingkaran dengan integer jari-jari.

1. Contoh perulangan
2. Kode Program

#include <iostream>

using namespace std;

main()

{

cout << "Nama: Royhan Abdurrohim\n";

cout << "NIM: 32602200018\n\n";

int l=0;

for (int j = 0; j <= 10; j++)

{

for (int k = j; k <=l; k++)

{

cout<<k<<' ';

}

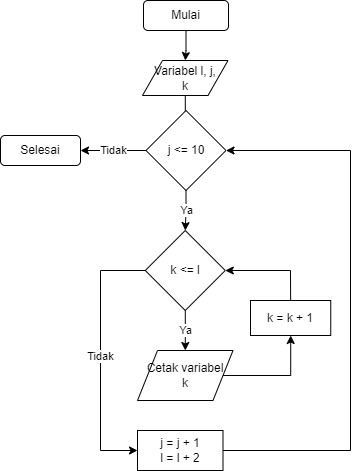
l+=2;

cout<<"\n";

}

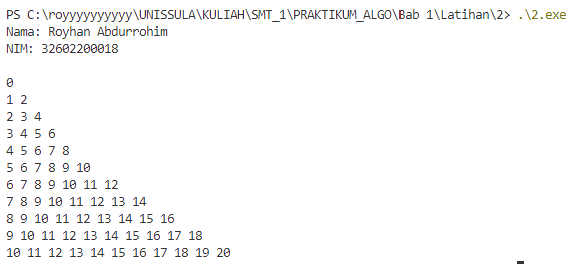
}

1. Flowchart



Gambar 1.8 Flowchart Latihan 2

1. Output



Gambar 1.9 Output Latihan 2

1. Penjelasan

Pada gambar 1.9 merupakan hasil dari perulangan atau *looping.* Untuk melakukan *looping*, kita bisa menggunakan *for* dan *while.* Diatas merupakan contoh *looping* dengan menggunakan for. Bisa kita lihat disini dalam membentuk perulangan for berlaku komposisi for(inisialisasi (nilai awal); condition (syarat perulangan); pengubah nilai pecacahan).

Inisialisasi adalah bagian untuk memberikan nlai awal untul variabel-variabel tertentu. Contohnya ketika kita ingin memulai perhitungan dari angka 9 maka pada bagian inisialisasi dimasukan angka 9 (variabel=9).

Kemudian ada syarat pengulangan. Syarat pengulangan adalah sampai mana kita ingin melakukan *looping* atau bisa di artikan angka terakhir yang kita inginkan itu apa. Contoh ketika kita ingin melakukan *looping* sampai angka 10 kita masukan tanda lebih dari sama dengan <=10 (variabel<=10).

Dan yang terakhir adalah pengubah nilai pecacahan. Pengubah nilai pecacahan adalah mengatur hasil *looping* itu naik atau turun. Contoh kita ingin melakukan *looping* dari terkecil ke terbesar kita masukan variabel++ misal kita ingin kelipatan 2 maka masukan variable +=2. Sedangkan jika ingin dari terbesar ke terkecil masukan variable — jika kita ingin berkelipatan 3 maka masukan variabel -=3.

1. Tugas
2. Modifikasilah program perulangan pada contoh 2 dengan struktur while menggunakan bahasa C.

Jawaban:

1. Kode Program

#include <stdio.h>

int main()

{

printf("Nama: Royhan Abdurrohim\n");

printf("NIM: 32602200018\n\n");

int l = 0, j = 0, k = 0;

while (j <= 10)

{

k = j;

while (k <= l)

{

printf("%i ", k);

k++;

}

printf("\n");

l += 2;

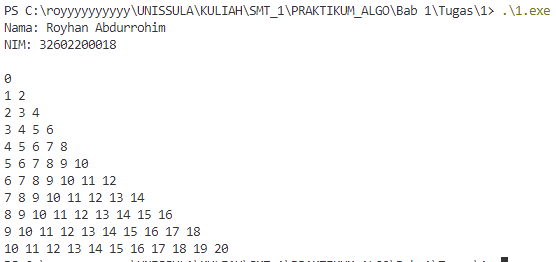
j++;

}

return 0;

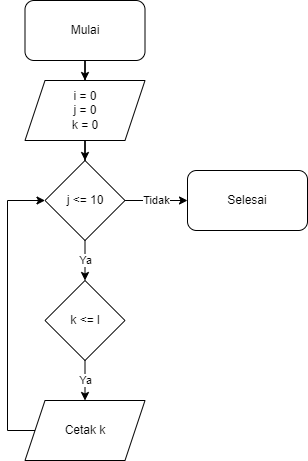
}

1. Output



Gambar 1.10 Output Tugas 1

1. Flowchart



Gambar 1.11 Flowchart Tugas 1

1. Penjelasan

Untuk output dari program ini sama seperti pada contoh nomor 2. Namun perbedaannya ada di Bahasa pemrograman dan struktur perulangan yang digunakan. Pada program ini, saya menggunakan Bahasa pemrograman C dan struktur perulangan while.

Secara umum, syntax pada Bahasa pemrograman C dan C++ sebenarnya cukup mirip. Jika di C++ kita menggunakan header iostream untuk melakukan input dan output, di C kita menggunakan header stdio.h. Kemudian jika di C++ kita menggunakan fungsi cout untuk mengeluarkan ouput, di C kita menggunakan fungsi printf.

1. Buatlah program yang dapat menghitung luas, keliling, dan diagonal persegipanjang. Diagonal dihitung dengan rumus:

diagonal = akar(panjang2 + lebar2). akar dapat dikerjakan dengan fungsi sqrt().

Masukan berupa panjang dan lebar. Program berupa pilihan untuk luas, keliling, dan diagonal.

Jawaban:

1. Kode Program

#include <iostream>

#include <cmath>

using namespace std;

main()

{

cout << "Nama: Royhan Abdurrohim\n";

cout << "NIM: 32602200018\n\n";

char jenis;

int panjang, lebar;

cout << "Fungsi persegi panjang" << endl;

cout << "Luas (L)" << endl;

cout << "Keliling (K)" << endl;

cout << "Diagonal (D)" << endl;

cout << "Pilih fungsi persegi panjang: ";

cin >> jenis;

jenis = tolower(jenis);

if (jenis != 'l' && jenis != 'k' && jenis != 'd')

{

cout << "Fungsi yang anda masukkan salah";

return 0;

}

cout << "Masukkan panjang: ";

cin >> panjang;

cout << "Masukkan lebar: ";

cin >> lebar;

if (jenis == 'l')

{

int luas = panjang \* lebar;

cout << "Luas = " << luas;

}

else if (jenis == 'k')

{

int keliling = (2 \* panjang) + (2 \* lebar);

cout << "Keliling = " << keliling;

}

else if (jenis == 'd')

{

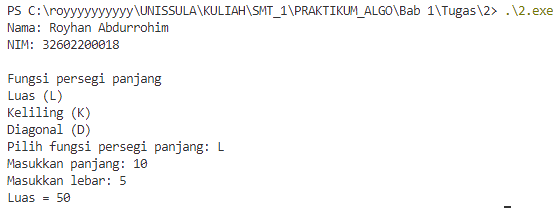
float diagonal = sqrt((panjang \* panjang) + (lebar \* lebar));

cout << "Diagonal = " << diagonal;

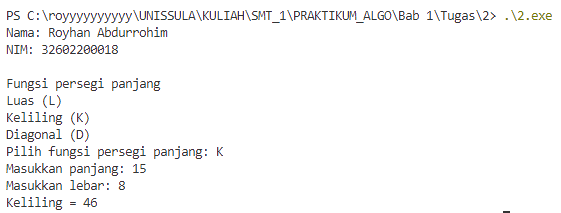
}

}

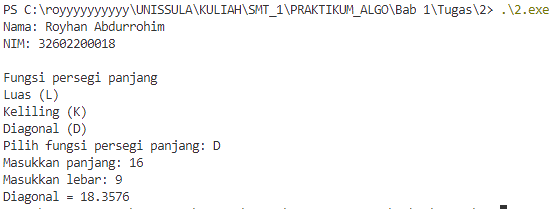
1. Output



Gambar 1.12 Output Tugas 2 Luas

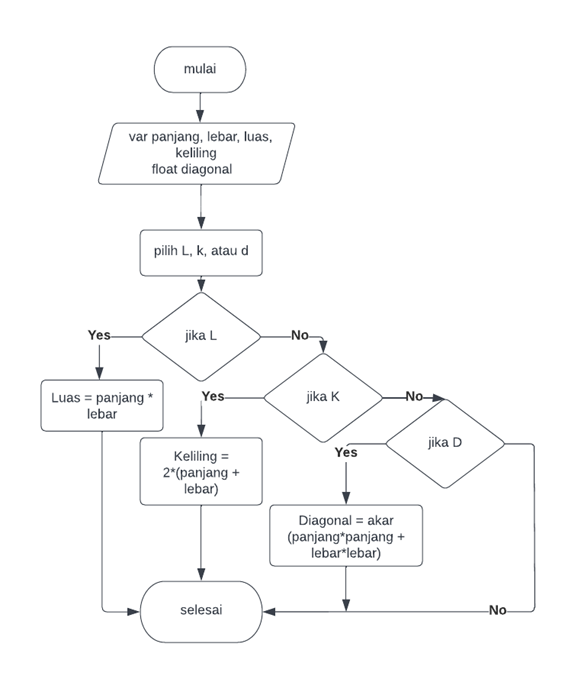


Gambar 1.13 Output Tugas 2 Keliling



Gambar 1.14 Output Tugas 2 Diagonal

1. Flowchart



Gambar 1.15 Flowchart Tugas 2

1. Penjelasan

Program ini dimulai dengan memasukkan header, header yang saya gunakan adalah iostream dan cmath. Header cmath digunakan untuk melakukan perhitungan matematika yang lebih kompleks. Pada program ini sendiri, nantinya akan digunakan untuk menghitung diagonal persegi panjang.

Saya menggunakan beberapa tipe data di program ini, yaitu: char untuk input tipe fungsi persegi panjang, int untuk input panjang dan lebar; hasil perhitungan luas; hasil perhitungan keliling, float untuk hasil perhitungan diagonal.

Struktur seleksi digunakan pada program ini untuk menyeleksi fungsi persegi panjang yang akan digunakan sesuai dengan input dari pengguna.

1. Modifikasilah program nomor 2 menjadi bahasa C

Jawaban:

1. Kode Program

#include <stdio.h>

#include <ctype.h>

#include <math.h>

int main()

{

printf("Nama: Royhan Abdurrohim\n");

printf("NIM: 32602200018\n\n");

char jenis;

int panjang, lebar;

printf("Fungsi persegi panjang\n");

printf("Luas (L)\n");

printf("Keliling (K)\n");

printf("Diagonal (D)\n");

printf("Pilih fungsi persegi panjang: ");

scanf("%s", &jenis);

jenis = tolower(jenis);

if (jenis != 'l' && jenis != 'k' && jenis != 'd')

{

printf("Fungsi yang anda masukkan salah");

return 0;

}

printf("Masukkan panjang: ");

scanf("%d", &panjang);

printf("Masukkan lebar: ");

scanf("%d", &lebar);

if (jenis == 'l')

{

int luas = panjang \* lebar;

printf("Luas = %d", luas);

}

else if (jenis == 'k')

{

int keliling = (2 \* panjang) + (2 \* lebar);

printf("Keliling = %d", keliling);

}

else if (jenis == 'd')

{

float diagonal = sqrt((panjang \* panjang) + (lebar \* lebar));

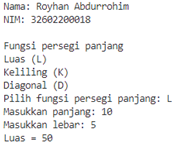
printf("Diagonal = %f", diagonal);

}

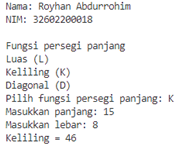
return 0;

}

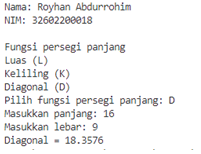
1. Output



Gambar 1,16 Output Tugas 3 Luas

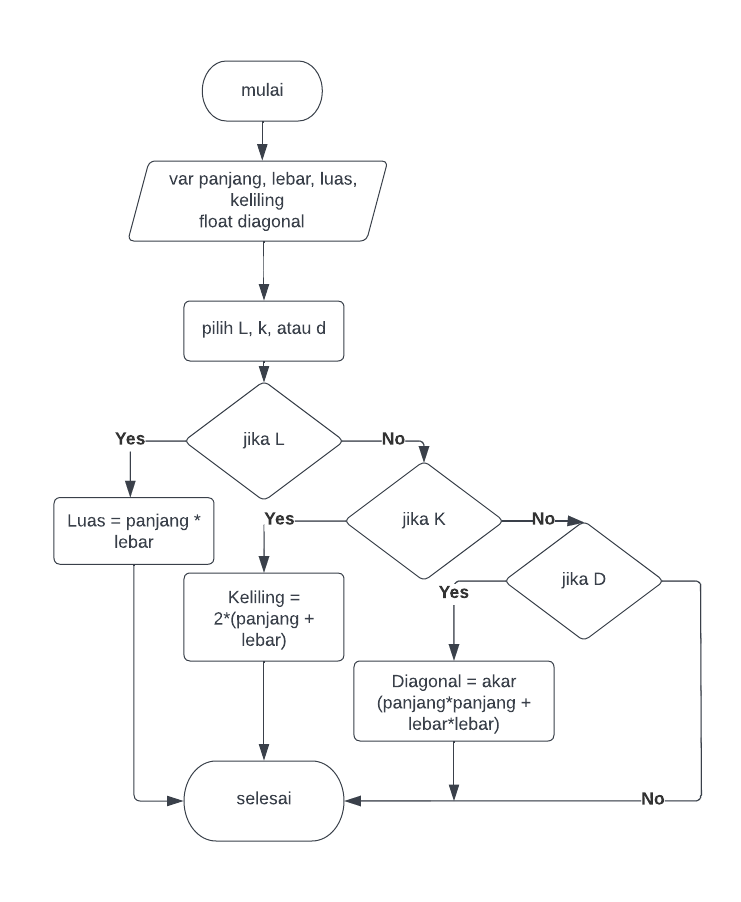


Gambar 1.17 Output Tugas 3 Keliling



Gambar 1.18 Output Tugas 3 Diagonal

1. Flowchart



Gambar 1.19 Flowchart Tugas 3

1. Penjelasan

Program ini memiliki fungsi yang sama seperti pada program nomor 2, perbedaannya adalah program ini ditulis dengan Bahasa C. Jika di Bahasa C++ kita menggunakan header cmath.h untuk dapat menggunakan fungsi sqrt, disini kita menggunakan header math.h. Disini kita juga menggunakan header ctype untuk dapat menggunakan fungsi tolower.

1. Buatlah program agar hasil outputnya menjadi persegi seperti gambar di bawah ini dalam bahasa C, dengan ketentuan panjang sisinya di inputkan.



Jawaban:

1. Kode Program

#include <stdio.h>

int main()

{

printf("Nama: Royhan Abdurrohim\n");

printf("NIM: 32602200018\n\n");

int panjang;

printf("Masukkan panjang sisi: ");

scanf("%d", &panjang);

for (int i = 0; i < panjang; i++)

{

for (int j = 0; j < panjang; j++)

{

if (i == 0 || i == panjang - 1 || j == 0 || j == panjang - 1)

{

printf("\* ");

}

else

{

printf(" ");

}

}

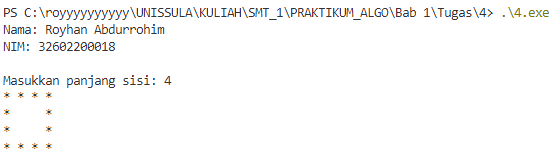
printf("\n");

}

return 0;

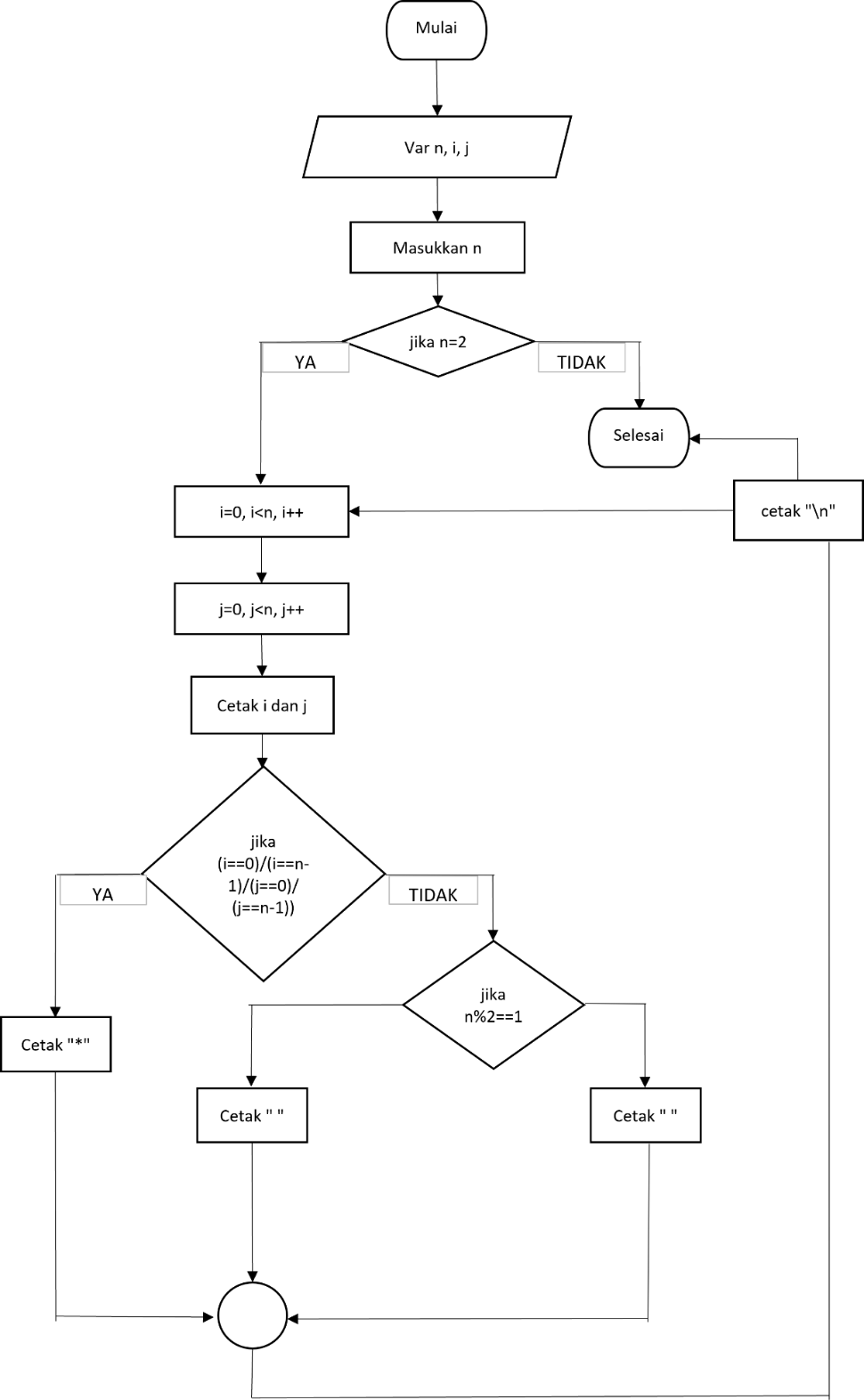
}

1. Output



Gambar 1.20 Output Tugas 4

1. Flowchart



Gambar 1.21 Flowchart Tugas 4

1. Penjelasan

Program ini menggunakan struktur seleksi dan percabangan di dalamnya. Disini saya menggunakan dua perulangan for, perulangan pertama digunakan sebagai indikator pada sisi vertikal, dan perulangan kedua digunakan sebagai indikator pada sisi horizontal.

1. Buatlah simulasi menu program dengan tampilan di bawah ini menggunakan WHILE.

MENU PILIHAN

* 1. Dangdut
  2. Pop
  3. Rock
  4. Exit

Pilihan Anda (1/2/3/4) ? ...

Apabila dipilih menu no 1, maka akan tampil teks “Anda memilih menu 1”. Demikian pula untuk menu 2 dan 3. Kemudian setelah itu muncul teks “Tekan ENTER untuk kembali ke menu utama”. Artinya begitu kita tekan ENTER menu pilihan akan muncul kembali, dst. Akan tetapi bila yang dipilih menu 4 (EXIT), program langsung berhenti.

Jawaban:

1. Kode Program

#include <iostream>

using namespace std;

main()

{

cout << "Nama: Royhan Abdurrohim\n";

cout << "NIM: 32602200018\n\n";

int pilihan;

do

{

cout << "MENU PILIHAN" << endl;

cout << "1. Dangdut" << endl;

cout << "2. Pop" << endl;

cout << "3. Rock" << endl;

cout << "4. Exit" << endl;

cout << "Pilihan Anda (1/2/3/4) ? ";

cin >> pilihan;

cin.ignore();

if (pilihan != 4)

{

cout << "Anda memilih menu " << pilihan << endl;

cout << "Tekan ENTER untuk kembali ke menu utama" << endl;

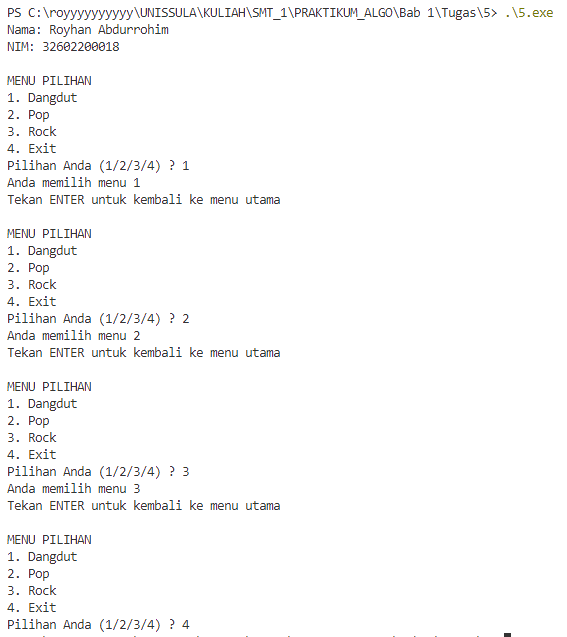
cin.ignore();

}

} while (pilihan != 4);

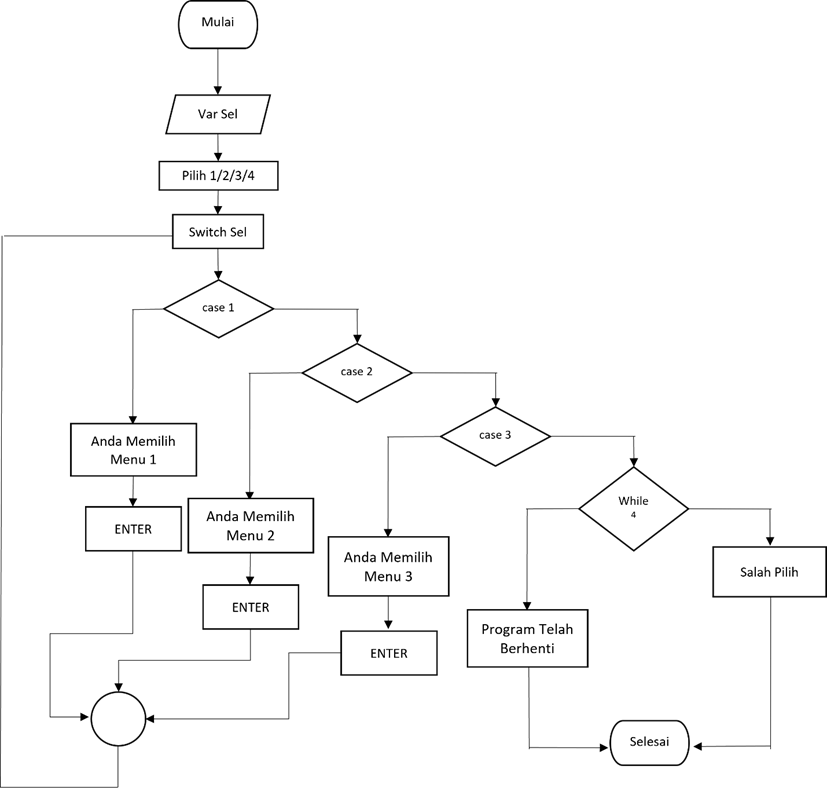
}

1. Output



Gambar 1.22 Output Tugas 5

1. Flowchart



Gambar 1.23 Flowchart Tugas 5

1. Penjelasan

Pada program ini, saya menggunakan perulangan do-while. Pada perulangan do-while, kode yang berada di dalam blok do akan dijalankan setidaknya satu kali walaupun kondisi di dalam while tidak terpenuhi. Itulah sebabnya menu pada program ini bisa muncul pada saat pertama kali dijalankan, walaupun nilai pada variabel pilihan tidak memenuhi kondisi while.

1. Kesimpulan

Kesimpulan pada bab kali ini adalah, dalam membuat program, kita harus familiar dengan sintaks pada bahasa pemrograman yang akan kita gunakan terlebih dahulu. Ini akan membantu kita dalam membuat kode yang efektif dan meminimalisir kemungkinan adanya bug pada program yang kita buat.

Walaupun setiap bahasa pemrograman memiliki sintaks dan ciri khasnya masing-masing, pada dasarnya mereka semua sama dan tidak akan jauh berbeda satu sama lainnya.

Disini kita juga telah mempelajari dan mengaplikasikan beberapa macam struktur dalam pemrograman , yaitu: sekuensial, seleksi dan perulangan.

# BAB II

# ARRAY 2D DAN 3D

## Tujuan

Adapun tujuan dari praktikum ini:

* + 1. Mengenal *array* 2 dimensi dan *array* 3 dimensi
    2. Mengerti dan memahami penggunaan array 2 dimensi dan 3 dimensi
  1. Dasar Teori

### Pengertian Array

*Array* adalah sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe yang sama. Setiap data tersebut menempati lokasi atau alamat memori yang berbeda- beda dan selanjutnya disebut dengan elemen *array.* Elemen *array* itu kemudian dapat kita akses melalui indeks yang terdapat di dalamnya.

Untuk mendeklarasikan sebuah *array* dalam C++, kita harus menggunakan tanda [] (*bracket*).

### Array Multidimensi

*Array* Multidimensi yaitu *array* yang terdiri dari beberapa subskrip *array*. Sebagai contoh, *array* 2 dimensi adalah *array* yang mempunyai 2 subskrip *array*, 3 dimensi mempunyai 3 subskrip *array* dan seterusnya. *Array* seperti ini sering digunakan untuk pemrosesan matriks.

* + - 1. *Array* Dua Dimensi

*Array* dua dimensi adalah *array* yang memilik dua buah elemen bertipe *array*. Dengan kata lain, *array* dua dimensi memiliki dua buah subskrip, yang biasanya dipresentasikan dengan baris dan kolom.

Contoh sederhana adalah data yang tertulis dalam tabel berikut ini :

Tabel 2. 1 Array 2 Dimensi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Merk HP | 1992 | 1993 | 1994 | 1995 |
| 1. | Nokia | 35 | 45 | 80 | 120 |
| 2. | Samsung | 100 | 110 | 70 | 101 |
| 3. | Sony | 10 | 15 | 20 | 17 |

Jika dibuat programnya adalah sebagai berikut :

Int data\_hp [3] [4];

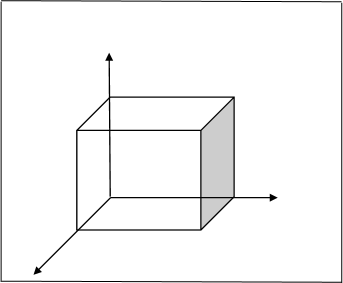
Bentuk umum pendeklarasian sebuah array dua dimensi adalah sebagai berikut :

tipe\_data nama\_array[jumlah\_elemen\_baris][jumlah\_elemen\_kolom] ;

* + - 1. *Array Tiga Dimensi*

Array tiga dimensi merupakan array yang memiliki tiga buah subskrip dan lebih kompleks apabila dibandingkan dengan array dua dimensi. Di sini, subskrip yang ada akan dipresentasikan dengan sumbu x, y, z atau panjang, lebar, dan tinggi seperti yang ditunjukkan oleh gambar berikut.

Bentuk umum pendeklarasian sebuah *array* 3 dimensi adalah sebagai berikut :



Gambar 2. 1 Array 3 Dimensi

tipe\_data nama\_array [jumlah\_elemen\_x] [jumlah\_elemen\_y] [jumlah\_elemen\_z] ;

### Array yang Bersifat Konstant

Nilai dalam elemen *array* dapat dibuat tetap, yaitu dengan mendefinisikannya sebagai konstanta. Caranya sama seperti pada saat kita membuat konstanta dari *identifier*, yaitu dengan menggunakan kata kunci const didepan nama *array* yang didefinisikan.

### Array sebagai tipe data bentukan

Dalam C++, *array* juga dapat digunakan sebagai tipe data bentukan seperti halnya struktur dan enumerasi*.* Untuk melakukan hal ini kita harus menggunakan kata kunci typedef, dimana berfungsi untuk memberikan nama lain dari *array* yang dideklarasikan. Berikut ini bentuk umum untuk membuat *array* sebagai tipe data bentukan.

typedef tipe\_data nama\_array[jumlah\_elemen];

### Array dari Karakter

Dalam C++, kumpulan karakter disebut dengan *string* (teks). Dan untuk mendeklarasikan *array* dari tipe karakter kita tentu akan menuliskannya ke dalam bentuk umum seperti dibawah ini :

char nama\_array[jumlah\_elemen];

Dengan demikian, apabila kita ingin melakukan deklarasi variabel *string* (misalnya dengan nama TEKS) yang terdiri dari 5 buah karakter, maka sintaksnya adalah sebagai berikut :

char TEKS[5] = {“B‟, “U‟, “D‟, “I‟, “\0‟};

Karakter terakhir, “\0” disebut dengan karakter *null*, yaitu karakter yang digunakan sebagai terminator dari sebuah *string* didalam C++. Namun, karena bentuk tersebut susah untuk ditulis dan riskan terhadap terjadinya sebuah kesalahan, maka C++ memperbolehkan kita untuk dapat menuliskan

karakter-karakter tersebut dalam sebuah string yaitu dengan cara seperti dibawah ini :

char TEKS[5] = “BUDI”;

* 1. Latihan
     1. Latihan 1
        1. Kode Program

#include <iostream>

using namespace std;

main()

{

cout << "Nama: Royhan Abdurrohim\n";

cout << "NIM: 32602200018\n\n";

int i, j;

int data[2][2];

cout << "Input Data" << endl;

for (i = 0; i <= 1; i++)

{

for (j = 0; j <= 1; j++)

{

cout << "data[" << i + 1 << "][" << j + 1 << "]=";

cin >> data[i][j];

}

}

cout << "Matrik yang diinputkan : " << endl;

for (i = 0; i <= 1; i++)

{

for (j = 0; j <= 1; j++)

{

cout << data[i][j] << " ";

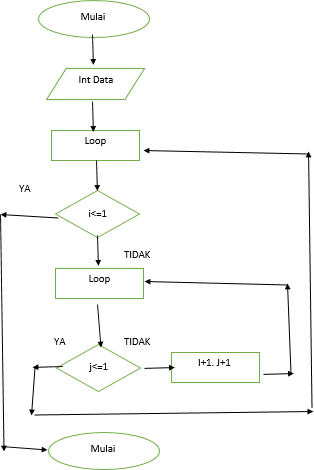
}

cout << endl;

}

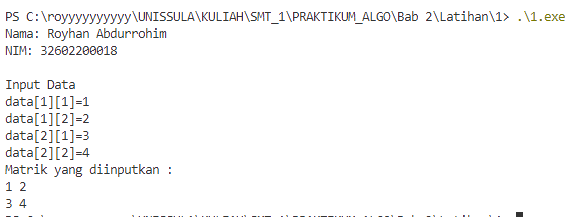
}

* + - 1. *Flowchart*



Gambar 2. 2 *Flowchart* Latihan 1

* + - 1. Output



Gambar 2. 3 Output Latihan 1

* + - 1. Penjelasan

Program pada latihan 1 ini menampilkan program *array* yang berupa matrik 2x2 (2 baris dan 2 kolom). *Header* yang digunakan adalah *iostream* untuk menampilkan *cin, cout, endl*, dan *endl*. Lalu menggunakan tipe data *integer* karena menunjukkan sebuah variabel untuk menyimpan nilai yang dimasukkan, dalam pemasukan nilai harus menggunakan tanda [] (*bracket)* sebagai pendeklarasian bentuk *array.* Pertama kita menginput nilai data yang diinginkan, menggunakan fungsi perulangan *(for*) untuk mengulang suatu proses yang telah diketahui banyaknya. Lalu membuat *output* data matriks yang telah diinputkan.

* + 1. Latihan 2
       1. Kode Program

#include <iostream>

using namespace std;

int main()

{

cout << "Nama: Royhan Abdurrohim\n";

cout << "NIM: 32602200018\n\n";

float nilai[2][3];

int baris, kolom;

for (baris = 0; baris < 2; baris++)

{

for (kolom = 0; kolom < 3; kolom++)

{

cout << "Nilai [baris][kolom] ke-[" << baris + 1 << "][" << kolom + 1 << "] = ";

cin >> nilai[baris][kolom];

}

}

cout << endl;

cout << "Matriks Nilai yang dimasukkan : " << endl

<< endl;

for (baris = 0; baris < 2; baris++)

{

for (kolom = 0; kolom < 3; kolom++)

{

cout << nilai[baris][kolom] << " ";

}

cout << endl;

}

cout << endl;

float jml\_nilai1, jml\_nilai2;

float rerata1, rerata2;

jml\_nilai1 = 0;

jml\_nilai2 = 0;

float jml\_nilaicol1, jml\_nilaicol2, jml\_nilaicol3;

float reratacol1, reratacol2, reratacol3;

jml\_nilaicol1 = 0;

jml\_nilaicol2 = 0;

jml\_nilaicol3 = 0;

for (baris = 0; baris < 2; baris++)

{

if (baris == 0)

{

for (kolom = 0; kolom < 3; kolom++)

jml\_nilai1 = jml\_nilai1 + nilai[baris][kolom];

}

else

{

for (kolom = 0; kolom < 3; kolom++)

jml\_nilai2 = jml\_nilai2 + nilai[baris][kolom];

}

}

rerata1 = jml\_nilai1 / 3;

rerata2 = jml\_nilai2 / 3;

cout << "Rerata baris ke 1 = " << rerata1 << endl;

cout << "Rerata baris ke 2 = " << rerata2 << endl;

// per kolom

cout << endl;

for (kolom = 0; kolom < 3; kolom++)

{

if (kolom == 0)

{

for (baris = 0; baris < 2; baris++)

jml\_nilaicol1 = jml\_nilaicol1 + nilai[baris][kolom];

}

else if (kolom == 1)

{

for (baris = 0; baris < 2; baris++)

jml\_nilaicol2 = jml\_nilaicol2 + nilai[baris][kolom];

}

else

{

for (baris = 0; baris < 2; baris++)

jml\_nilaicol3 = jml\_nilaicol3 + nilai[baris][kolom];

}

}

reratacol1 = jml\_nilaicol1 / 2;

reratacol2 = jml\_nilaicol2 / 2;

reratacol3 = jml\_nilaicol3 / 2;

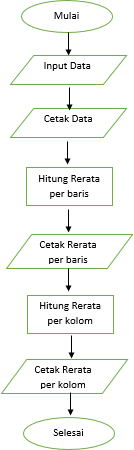
cout << "Rerata kolom ke 1 = " << reratacol1 << endl;

cout << "Rerata kolom ke 2 = " << reratacol2 << endl;

cout << "Rerata kolom ke 3 = " << reratacol3 << endl;

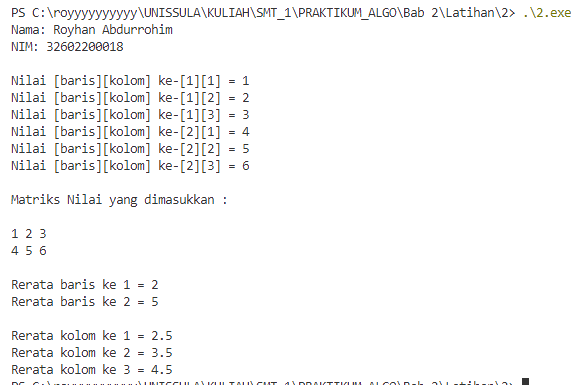
}

* + - 1. Flowchart



Gambar 2. 4 Flowchart Latihan 2

* + - 1. Output



Gambar 2. 5 Output Latihan 2

* + - 1. Penjelasan

Program ini menampilkan array matriks 2x3 (2 baris dan 3 kolom) beserta rerata baris dan kolomnya. *Header* yang digunakan adalah *iostream* yang merupakan *header* pada program C++ untuk menampilkan *cin, cout, endl*, dan *ends.* Tipe data yang digunakan adalah *float* sebagai deklarasi nilai desimal. Menggunakan fungsi perulangan (*for*) untuk mengulang suatu proses yang telah diketahui banyaknya. Menggunakan fungsi (*if*) untuk menyeleksi kondisi, yaitu jika pernyataan pertama tidak sesuai maka diseleksi dipernyataan kedua dan seterusnya. Maka disertakan pula fungsi (*else*) yang merupakan lanjutan dari fungsi (*if*) dimana jika kondisi benar maka statement pertama yang akan dilaksanakan, tetapi jika kondisi salah maka statement yang ada pada blok *else* yang dilaksanakan. Lalu memasukkan inputan baris dan kolom serta nilai matriks yang diinginkan. Memasukkan input rerata baris dan kolom. Jika telah memasukkan input dengan benar maka akan menghasilkan output matriks baris dan kolom yaitu 2 baris dan 3 kolom serta menghasilkan output rerata matriks sesuai dengan baris dan kolomnya.

* + 1. Latihan 3
       1. Kode Program

#include <iostream>

using namespace std;

main()

{

cout << "Nama: Royhan Abdurrohim\n";

cout << "NIM: 32602200018\n\n";

int i, j, k, x = 1;

int matrik[3][4][4];

cout << " 3 dimensi array " << endl

<< endl;

for (i = 0; i < 3; i++)

{

for (j = 0; j < 4; j++)

{

for (k = 0; k < 5; k++)

{

matrik[i][j][k] = x;

cout << " " << matrik[i][j][k];

x++;

}

cout << endl;

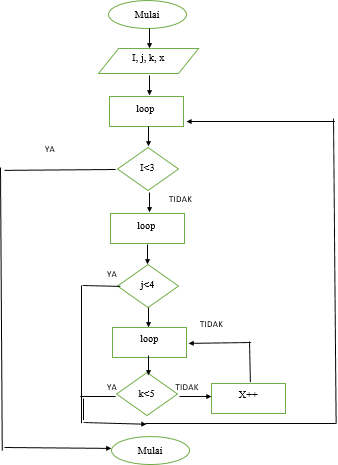
}

cout << endl;

}

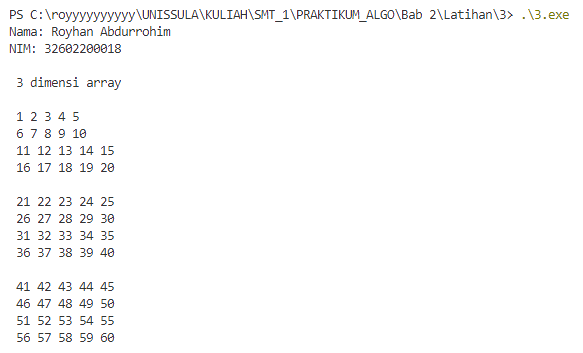
}

* + - 1. Flowchart



Gambar 2. 6 Flowchart Latihan 3

* + - 1. Output



Gambar 2. 7 Output Latihan 3

* + - 1. Penjelasan

Program ini menampilkan *array* 3 dimensi. *Header* yang digunakan adalah *iostream* karena menggunakan bahasa pemrograman C++ yaitu untuk menampilkan *cin, cout, endl,* dan *ends*. Tipe data yang digunakan adalah *integer* yang menampilkan variabel untuk menyimpan nilai yang dimasukkan. Matriks *array* 3 dimensi dapat digambarkan sebagai benda ruang. Program diatas di-*input*-kan 3 dimensi dimana angka-angkanya selalu berselisihan. Dan program tersebut menggunakan fungsi (*for*) untuk mengulang suatu proses yang telah diketahui banyaknya, lalu ada juga fungsi matriks yang berfungsi sebagai kumpulan bilangan, simbol, atau ekspresi.

* 1. Tugas
     1. Ubahlah Latihan 2 kedalam Bahasa C!

Jawaban:

* + - 1. Kode Program

#include <stdio.h>

int main()

{

printf("Nama: Royhan Abdurrohim\n");

printf("NIM: 32602200018\n\n");

float nilai[2][3];

int baris, kolom;

for (baris = 0; baris < 2; baris++)

{

for (kolom = 0; kolom < 3; kolom++)

{

printf("Nilai [baris][kolom] ke-[%d][%d] = ", baris + 1, kolom + 1);

scanf("%f", &nilai[baris][kolom]);

}

}

printf("\nMatriks Nilai yang dimasukkan : \n\n");

for (baris = 0; baris < 2; baris++)

{

for (kolom = 0; kolom < 3; kolom++)

{

printf("%g ", nilai[baris][kolom]);

}

printf("\n");

}

printf("\n");

float jml\_nilai1, jml\_nilai2;

float rerata1, rerata2;

jml\_nilai1 = 0;

jml\_nilai2 = 0;

float jml\_nilaicol1, jml\_nilaicol2, jml\_nilaicol3;

float reratacol1, reratacol2, reratacol3;

jml\_nilaicol1 = 0;

jml\_nilaicol2 = 0;

jml\_nilaicol3 = 0;

for (baris = 0; baris < 2; baris++)

{

if (baris == 0)

{

for (kolom = 0; kolom < 3; kolom++)

{

jml\_nilai1 = jml\_nilai1 + nilai[baris][kolom];

}

}

else

{

for (kolom = 0; kolom < 3; kolom++)

{

jml\_nilai2 = jml\_nilai2 + nilai[baris][kolom];

}

}

}

rerata1 = jml\_nilai1 / 3;

rerata2 = jml\_nilai2 / 3;

printf("Rerata baris ke 1 = %g\n", rerata1);

printf("Rerata baris ke 2 = %g\n", rerata2);

printf("\n");

for (kolom = 0; kolom < 3; kolom++)

{

if (kolom == 0)

{

for (baris = 0; baris < 2; baris++)

{

jml\_nilaicol1 = jml\_nilaicol1 + nilai[baris][kolom];

}

}

else if (kolom == 1)

{

for (baris = 0; baris < 2; baris++)

{

jml\_nilaicol2 = jml\_nilaicol2 + nilai[baris][kolom];

}

}

else

{

for (baris = 0; baris < 2; baris++)

{

jml\_nilaicol3 = jml\_nilaicol3 + nilai[baris][kolom];

}

}

}

reratacol1 = jml\_nilaicol1 / 2;

reratacol2 = jml\_nilaicol2 / 2;

reratacol3 = jml\_nilaicol3 / 2;

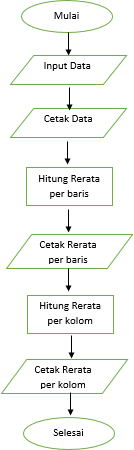
printf("Rerata kolom ke 1 = %g\n", reratacol1);

printf("Rerata kolom ke 2 = %g\n", reratacol2);

printf("Rerata kolom ke 3 = %g\n", reratacol3);

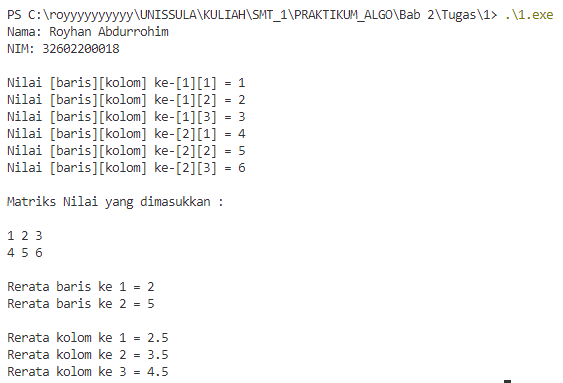
}

* + - 1. Flowchart



Gambar 2. 8 Flowchart Tugas 1

* + - 1. Output



Gambar 2. 9 Output Tugas 1

* + - 1. Penjelasan

Program ini menampilkan *array* matriks 2x3 (2 baris dan 3 kolom) beserta rerata baris dan kolomnya. *Header* yang digunakan adalah *stdio*.*h* yang merupakan *header* bahasa C yaitu *header* yang berisi definisi *makro*, *konstanta*, dan deklarasi fungsi serta jenis yang digunakan untuk berbagai standard *input output* operasi. Tipe data yang digunakan adalah *integer* untuk menunjukkan sebuah variabel. Kita menginputkan data matriks. Menggunakan fungsi *printf* untuk menampilkan keluaran data serta hasil rerata baris kolom yang telah diinputkan. Menggunakan fungsi perulangan (*for*) untuk mengulang suatu proses yang telah diketahui banyaknya

* + 1. Jalankan Program Berikut
       1. Kode Program

#include <iostream>

using namespace std;

int main()

{

cout << "Nama: Royhan Abdurrohim\n";

cout << "NIM: 32602200018\n\n";

typedef int matrik[3][2];

matrik elemen1, elemen2, hasil\_jumlah;

int baris, kolom;

for (baris = 0; baris < 3; baris++)

{

for (kolom = 0; kolom < 2; kolom++)

{

cout << "Elemen A[" << baris << "][" << kolom << "] = ";

cin >> elemen1[baris][kolom];

}

}

cout << endl;

for (baris = 0; baris < 3; baris++)

{

for (kolom = 0; kolom < 2; kolom++)

{

cout << "Elemen B[" << baris << "][" << kolom << "] = ";

cin >> elemen2[baris][kolom];

}

}

cout << endl;

{

for (baris = 0; baris < 3; baris++)

{

for (kolom = 0; kolom < 2; kolom++)

{

hasil\_jumlah[baris][kolom] = elemen1[baris][kolom] + elemen2[baris][kolom];

}

}

system("cls");

cout << "Hasil Penjumlahan Elemen A dan Elemen B adalah : " << endl;

for (baris = 0; baris < 3; baris++)

{

for (kolom = 0; kolom < 2; kolom++)

{

cout << "Hasil Jumlah [" << baris << "][" << kolom << "] = " << hasil\_jumlah[baris][kolom] << endl;

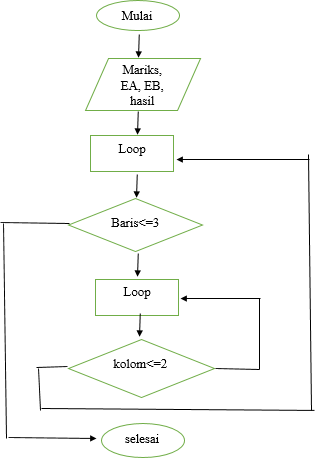
}

}

}

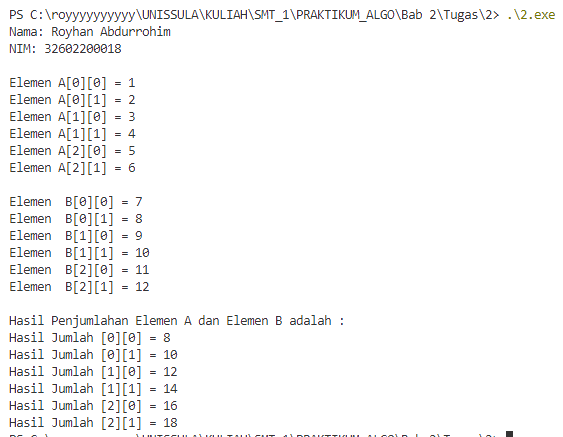
}

* + - 1. Flowchart



Gambar 2. 10 Flowchart Tugas 2

* + - 1. Output



Gambar 2. 11 Output Tugas 2

* + - 1. Penjelasan

Program ini menampilakan hasil penjumlahan dari elemen A dan B dengan menginputkan matrik baris dan kolom lalu menampilkan hasil dari penjumlahan dari kedua elemen tersebut. *Header* yang digunakan adalah *iostream* karena menggunakan bahasa pemrograman C++ berfungsi untuk menampilkan *cin, cout, endl,* dan *ends*. Tipe data yang digunakan adalah *typedef* yang digunakan untuk mendeklarasikan tipe data dengan nama lain. Serta menggunakan fungsi perulangan (*for*) untuk mengulang suatu proses yang telah diketahui banyaknya.

* + 1. Modifikasi tugas 2 dengan menginputkan jumlah baris dan kolom. Kemudian tambahkan operasi pengurangan dan perkalian didalamnya!
       1. Kode program

#include <iostream>

using namespace std;

int main()

{

cout << "Nama: Royhan Abdurrohim\n";

cout << "NIM: 32602200018\n\n";

typedef int \*\*matrik;

matrik elemen1, elemen2, hasil\_jumlah, hasil\_kurang, hasil\_kali;

int baris, kolom;

cout << "Jumlah baris = ";

cin >> baris;

cout << "Jumlah kolom = ";

cin >> kolom;

elemen1 = new int \*[kolom];

elemen2 = new int \*[kolom];

hasil\_jumlah = new int \*[kolom];

hasil\_kurang = new int \*[kolom];

hasil\_kali = new int \*[kolom];

for (int i = 0; i < kolom; i++)

{

elemen1[i] = new int(baris);

elemen2[i] = new int(baris);

hasil\_jumlah[i] = new int(baris);

hasil\_kurang[i] = new int(baris);

hasil\_kali[i] = new int(baris);

}

for (int i = 0; i < baris; i++)

{

for (int j = 0; j < kolom; j++)

{

cout << "Elemen A[" << i << "][" << j << "] = ";

cin >> elemen1[i][j];

}

}

cout << endl;

for (int i = 0; i < baris; i++)

{

for (int j = 0; j < kolom; j++)

{

cout << "Elemen B[" << i << "][" << j << "] = ";

cin >> elemen2[i][j];

}

}

for (int i = 0; i < baris; i++)

{

for (int j = 0; j < kolom; j++)

{

hasil\_jumlah[i][j] = elemen1[i][j] + elemen2[i][j];

}

}

for (int i = 0; i < baris; i++)

{

for (int j = 0; j < kolom; j++)

{

hasil\_kurang[i][j] = elemen1[i][j] - elemen2[i][j];

}

}

for (int i = 0; i < baris; i++)

{

for (int j = 0; j < kolom; j++)

{

hasil\_kali[i][j] = elemen1[i][j] \* elemen2[i][j];

}

}

system("cls");

cout << endl;

cout << "Hasil Penjumlahan Elemen A dan Elemen B adalah : " << endl;

for (int i = 0; i < baris; i++)

{

for (int j = 0; j < kolom; j++)

{

cout << "Hasil Jumlah [" << i << "][" << j << "] = " << hasil\_jumlah[i][j] << endl;

}

}

cout << endl;

cout << "Hasil Pengurangan Elemen A dan Elemen B adalah : " << endl;

for (int i = 0; i < baris; i++)

{

for (int j = 0; j < kolom; j++)

{

cout << "Hasil Kurang [" << i << "][" << j << "] = " << hasil\_kurang[i][j] << endl;

}

}

cout << endl;

cout << "Hasil Perkalian Elemen A dan Elemen B adalah : " << endl;

for (int i = 0; i < baris; i++)

{

for (int j = 0; j < kolom; j++)

{

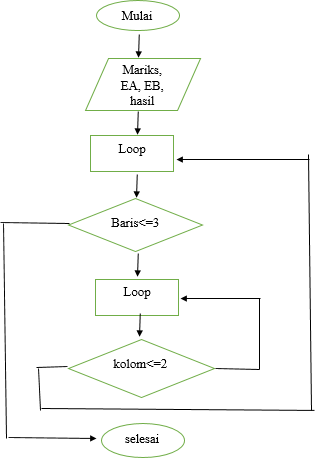
cout << "Hasil Kali [" << i << "][" << j << "] = " << hasil\_kali[i][j] << endl;

}

}

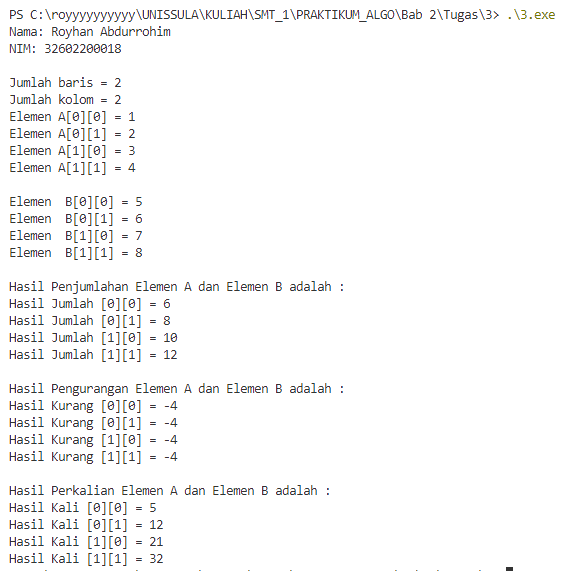
}

* + - 1. Flowchart



Gambar 2. 12 Flowchart Tugas 3

* + - 1. Output



Gambar 2. 13 Output Tugas 3

* + - 1. Penjelasan

Untuk tugas 3 hampir sama dengan tugas 2, Yaitu dengan menginputkan elemen A dan elemen B. Lalu menyertakan *output* hasil dari penjumlahan, pengurangan, dan perkalian kedua elemen tersebut. *Header* yang digunakan adalah *iostream* karena menggunakan bahasa pemrograman C++ berfungsi untuk menampilkan *cin, cout, endl,* dan *ends*. Tipe data yang digunakan adalah *typedef* yang digunakan untuk mendeklarasikan tipe data dengan nama lain. Serta menggunakan fungsi perulangan (*for*) untuk mengulang suatu proses yang telah diketahui banyaknya.

* + 1. Jalankan program berikut dan kedalam bahasa C++ !
       1. Kode program

#include <iostream>

using namespace std;

main()

{

cout << "Nama: Royhan Abdurrohim\n";

cout << "NIM: 32602200018\n\n";

char h = 64, nama[4][3][13] = {

"Angsa",

"Anjing",

"Anoa",

"Banteng",

"Badak",

"Beruang",

"Camar",

"Capung",

"Cendrawasih",

"Dara",

"Domba",

"Duyung",

};

printf("Daftar Hewan sesuai abjad : \n\n");

for (int i = 0; i < 4; i++)

{

++h;

printf("Kelompok %c \n", h);

for (int s = 0; s < 3; s++)

{

printf("%d. %s \n", s + 1, nama[i][s]);

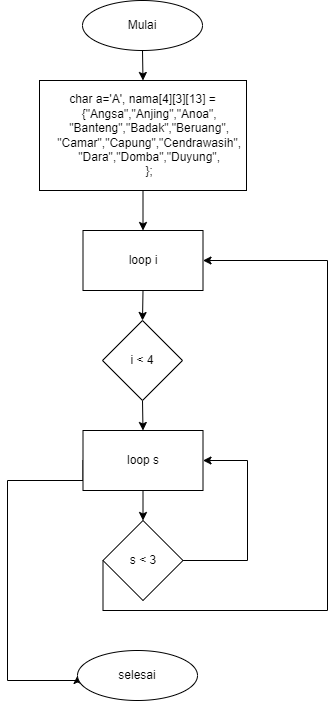
}

printf("\n");

}

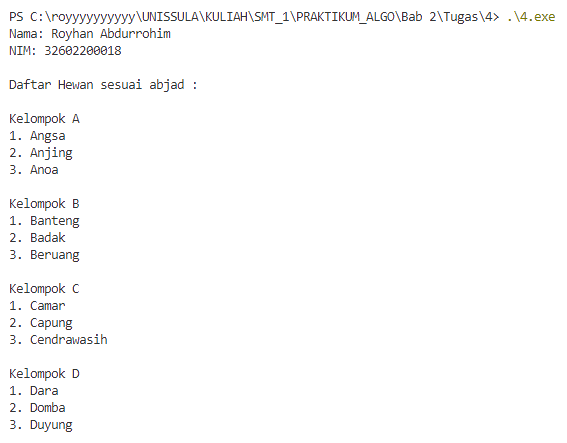
}

* + - 1. Flowchart



Gambar 2. 14 FLowchart Tugas 4

* + - 1. Output



Gambar 2. 15 Output Tugas 4

* + - 1. Penjelasan

Program dimulai dengan menginisialisasi variabel char h yang berisi nilai desimal untuk karakter didalam tabel ASCII dan char nama yang berisi nama hewan sesuai abjad. Kemudian kita melakukan looping untuk menampilkan nama hewan yang dikelompokkan berdasarkan sesuai abjad.

* + 1. Buatlah program untuk membentuk sebuah kelompok dimana jumlah kelompok dan anggota diinputkan. Kemudian tampilkan kembali kelompok dan anggota yang diinputkan.
       1. Kode program

#include <iostream>

using namespace std;

int main()

{

cout << "Nama: Royhan Abdurrohim\n";

cout << "NIM: 32602200018\n\n";

int jumah\_kelompok, jumlah\_anggota;

cout << "Masukkan jumlah kelompok = ";

cin >> jumah\_kelompok;

cout << "Masukkan jumlah anggota = ";

cin >> jumlah\_anggota;

cout << endl;

string \*\*kelompok = new string \*[jumlah\_anggota];

for (int i = 0; i < jumah\_kelompok; i++)

{

kelompok[i] = new string[jumlah\_anggota];

}

for (int i = 0; i < jumah\_kelompok; i++)

{

cout << "Anggota kelompok " << i + 1 << ":" << endl;

for (int j = 0; j < jumlah\_anggota; j++)

{

cout << "Masukkan nama anggota nomor " << j + 1 << " = ";

cin >> kelompok[i][j];

}

cout << endl;

}

for (int i = 0; i < jumah\_kelompok; i++)

{

cout << "Kelompok " << i + 1 << endl;

for (int j = 0; j < jumlah\_anggota; j++)

{

cout << j + 1 << ". " << kelompok[i][j] << endl;

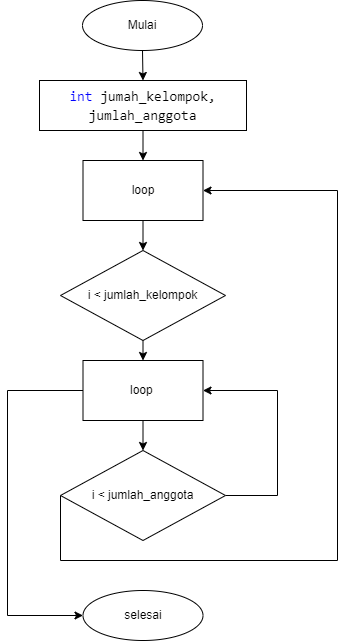
}

cout << endl;

}

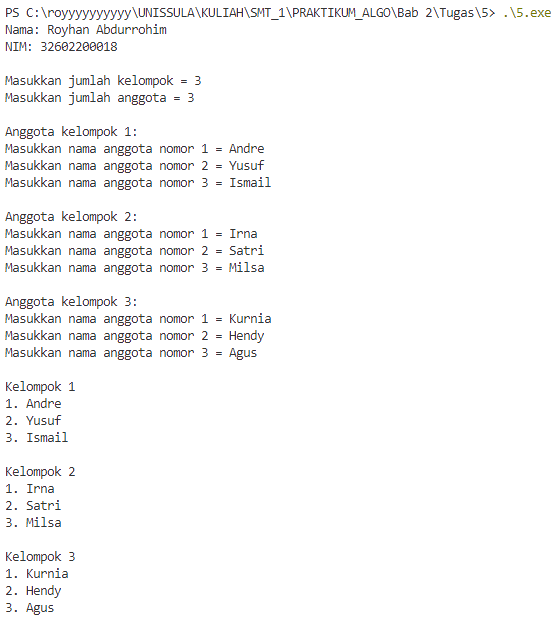
}

* + - 1. Flowchart



Gambar 2. 16 Flowchart Tugas 5

* + - 1. Output



Gambar 2. 17 Output Tugas 5

* + - 1. Penjelasan

Program dimulai dengan menginputkan jumlah kelompok dan jumlah anggota yang diinginkan. Kemudian kita membuat array dinamis yang akan digunakan untuk menampung data nama-nama anggota kelompok. Setelah itu kita melakukan looping sebanyak jumlah kelompok dan anggota untuk menerima input nama anggota kelompok. Setelah itu kita melakukan looping terhadap array dinamis yang telah dibuat untuk menampilkan kembali data kelompok dan nama-nama anggotanya.

* 1. Kesimpulan

Pada bab kali ini telah mempelajari array 2D dan 3D. Array merupakan

struktur data yang dapat digunakan untuk menyimpan beberapa data dengan tipe data yang sama. Data-data tersebut menempati alamat memori yang berbeda. Di bahasa pemrograman C++, kita dapat mendeklarasikan array dengan menggunakan simbol [] (bracket). Untuk mengakses data pada array, kita harus menggunakan indeks.

Array memiliki beberapa macam bentuk, diantaranya yaitu: array 1D, array 2D dan array 3D. Array 1D adalah array yang hanya berisi satu subscript array saja. Array 2D merupakan array yang terdiri dari 2 subscript array, dapat juga direpresentasikan sebagai baris dan kolom. Array 3D merupakan array yang terdiri dari 3 subscript array, dapat juga direpresentasikan sebagai panjang, lebar dan tinggi.

Array sering digunakan pada penerapan matriks. Dengan mempelajari bab ini, kita telah memahami tentang array serta penerapannya dalam bahasa pemrograman.

# BAB III

# METODE SORTING DAN SEARCHING

## Tujuan

Adapun tujuan dari praktikum ini:

* 1. Mengerti dan Memahami Metode Pencarian Data
  2. Dapat menggunakan Metode Pencarian Data dalam Program
  3. Memahami penggunaan metode Sorting

## Dasar Teori

### Pencarian Data

Pencarian (searching) merupakan tindakan untuk mendapatkan suatu data dalam kumpulan data. Dalam kehidupan sehari-hari ,seringkali kita berurusan dengan pencarian, misalnya untuk menemukan nomor telepon seseorang pada buku telepon atau mencari suatu istilah dalam kamus.

Terdapat beragam algoritma pencarian untuk keperluan mencari data. Perlu diketahui, yang di maksud dengan algoritma pencarian adalah “algoritma yang menerima sebuah argument a dan mencoba untuk menemukan sebuah rekaman yang memiliki kunci a”.

Pencarian data dilakukan terhadap data dapat dilakukan terhadap data yang secara keseluruhan berada dalam memory computer ataupun terhadap data yang berada dalam penyimpan eksternal.

### Bubble Sort

Sorting merupakan suatu proses untuk menyusun kembali humpunan obyek menggunakan aturan tertentu. Sorting disebut juga sebagai suatu algoritma untuk meletakkan kumpulan elemen data kedalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen. Pada dasarnya ada dua macam urutan yang biasa digunakan dalam suatu proses sorting:

1. Urut naik (ascending)

Mengurutkan dari data yang mempunyai nilai paling kecil sampai paling besar

1. Urut turun (descending)

Mengurutkan dari data yang mempunyai nilai paling besar sampai paling kecil.

Metode-metode sorting yang akan dibahas kali ini meliputi:

1. Bubble Sort (Metode Gelembung)
2. Insertion Sort (Metode Penyisipan) Desertion Sort.
3. Selection Sort (Metode Seleksi)
4. Quick Sort (Metode Quick)

Metode gelembung (bubble sort) sering juga disebut dengan metode penukaran (exchange sort) adalah metode yang mengurutkan data dengan cara membandingkan masing-masing elemen, kemudian melakukan penukaran bila perlu. Metode ini mudah dipahami dan diprogram, tetapi bila dibandingkan dengan metode lain yang kita pelajari, metode ini merupakan metode yang paling tidak efisien.

**Algoritma Bubble Sort**

* 1. Membandingkan data ke-i dengan data ke-(i+1) (tepat bersebelahan). Jika tidak sesuai maka tukar (data ke-i = data ke(i+1) dan data ke-(i+1) = data ke-i). Apa maksudnya tidak sesuai? Jika kita menginginkan algoritme menghasilkan data dengan urutan ascending (A-Z) kondisi tidak sesuai adalah data ke-i > data ke-i+1, dan sebaliknya untuk urutan descending (A-Z).
  2. Membandingkan data ke-(i+1) dengan data ke-(i+2). Kita melakukan pembandingan ini sampai data terakhir. Contoh: 1 dgn 2; 2 dgn 3; 3 dgn 4; 4 dgn 5 … ; n-1 dgn n.
  3. Selesai satu iterasi, adalah jika kita sudah selesai membandingkan antara (n-1) dgn n. Setelah selesai satu iterasi kita lanjutkan lagi iterasi berikutnya sesuai dengan aturan ke-1. mulai dari data ke-1 dgn data ke-2, dst.
  4. Proses akan berhenti jika tidak ada pertukaran dalam satu iterasi.

## Latihan

1. Latihan 1
2. Kode Program

#include <iostream>

#include <conio.h>

using namespace std;

int cari(int data[], int n, int k)

{

int posisi, i, ketemu;

if (n <= 0)

posisi = -1;

else

{

ketemu = 0;

i = 1;

while ((i < n - 1) && !ketemu)

if (data[i] == k)

{

posisi = i;

ketemu = 1;

}

else

i++;

if (!ketemu)

posisi = -1;

}

return posisi;

}

int main()

{

int data[8];

int dicari;

cout << "Masukkan Data : ";

for (int a = 0; a < 8; a++)

{

cin >> data[a];

}

cout << "Masukkan Data yang dicari : ";

cin >> dicari;

cout << "Posisi " << dicari << " dalam larik data: "

<< (cari(data, 8, dicari)) << "\n";

getch();

}2. Latihan 2.

Contoh Program Bubble Sort.

#include <iostream> using namespace std; int main() { int i,j,n,data[10],simpan,k; cout<<"masukkan banyak data= ";cin>>n; for(i=1;i<=n;i++)

{ cout<<"data "<<i<<" = ";cin>>data[i];

} cout<<"awal = "; for(i=1;i<=n;i++) cout<<data[i]<<" "; cout<<endl;

for(i=1;i<n;i++)

{ for(j=1;j<n;j++)

{

if(data[j]>data[j+1])

{ simpan=data[j]; data[j]=data[j+1]; data[j+1]=simpan;

}

} } cout<<"hasil= "; for(i=1;i<=n;i++) cout<<data[i]<<" "; return 0;

}

Tugas.

1. Tugas 1.

Ubahlah script di bawah ini menggunkan bahasa C.

#include <iostream> using namespace std; int data[10],data2[10]; int n;

void tukar(int a,int b)

{ int t; t = data[b]; data[b] = data[a]; data[a] = t; } void Input() { cout<<"Masukkan jumlah data = ";cin>>n; cout<<"--------------------------------------"<<endl; for(int i=0;i<n;i++)

{ cout<<"Masukkan data ke-"<<(i+1)<<" = ";cin>>data[i]; data2[i] = data[i];

} cout<<endl; } void Tampil() { for(int i=0;i<n;i++)

{ cout<<data[i]<<" ";

} cout<<endl;

}void bubble\_sort()

{ for(int i=1;i<n;i++)

{ for(int j=n-1;j>=i;j--) { if(data[j]<data[j-1]) tukar(j,j-1);

}

Tampil(); } cout<<endl; } main() {

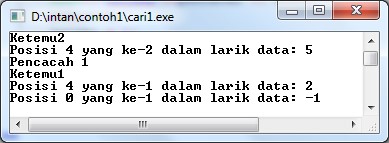
cout<<"\* Bubble Sort \*"<<endl; Input(); cout<<"Proses Bubble Sort,,,,,,,"<<endl;

cout<<"-------------------------------------"<<endl; Tampil(); bubble\_sort();

cout<<"-------------------------------------"<<endl;}

2. Tugas 2.

Jalankan program agar menghasilkan tampilan seperti gambar berikut:



#include <iostream> using namespace std; int cari (int data[],int n,int k,int m)

{ int posisi,i,ketemu,pencacah=0; if (n <= 0) posisi=-1; else{ ketemu = 0; i=1; while ((i<n+1)&&! ketemu) if (data[i] == k)

{ pencacah++; cout<<"Pencacah " <<pencacah <<endl; if (pencacah != m)

{ cout<<"Ketemu"<<pencacah << endl; posisi=i; ketemu = 1;

} else

I--;

} else i++; if (!ketemu) } return posisi;

} int main() { int data [9]={10,9,4,6,3,4,2,5,0}; int dicari; int ke; dicari=4; ke =2; cout<<"Posisi " <<dicari<<" yang ke-"<<ke<<" dalam larik data: " <<(cari(data,8,dicari,ke))<<"\n"; ke =1; cout<<"Posisi " <<dicari<<" yang ke-"<<ke<<" dalam larik data: " <<(cari(data,8,dicari,ke))<<"\n"; dicari = 0; ke=1;

cout<<"Posisi " <<dicari<<" yang ke-"<<ke<<" dalam larik data: " <<(cari(data,9,dicari,ke))<<"\n"; return 0;

}

Tugas 3.

Ubahlah Script pada latihan 1 dengan menggunakan metode Insertion Sort secara descending.

1. Tugas 4.

Buatlah program pengurutan data dengan menggunakan semua metode pengurutan untuk mengurutkan angka secara ascending dan descending.

1. Tugas 5.

Ubahlah script di bawah ini dengan menggunakan bahasa java.

#include <stdio.h> void tampilkan\_larik (int data[],int n)

{ int i; for(i=0;i< n ;i++) printf("%d ",data[i]); printf("\n");

} int partisi(int data[],int p,int r)

{ int pivot= data[r]; int i = p-1; int j; for (j=p;j<r;j++)

{ if (data[j]<= pivot)

{ i++; int tmp = data[i]; data[i]= data[j]; data[j]= tmp;

} }

int tmp = data[i+1]; data[i+1]= data[r]; data[r]= tmp; return i+1;

} void quick\_sort(int data[],int p,int r)

{

int q; if (p<r) { q = partisi(data,p,r); quick\_sort(data,p,q-1); quick\_sort(data,q+1,r);

} } int main() { int jum\_data = 9; int i; int data[]= {25,57,48,37,12,92,80,33,1};

quick\_sort(data,0,jum\_data-1); printf("hasil pengurutan : \n"); tampilkan\_larik(data,jum\_data); return 0;

}**DAFTAR PUSTAKA**



