**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN**

**SEMESTER GANJIL TAHUN AKADEMIK 2023/2024**

****

**Disusun oleh :**

**Nama** : Sabastian Rafariza Wahyuono

**NIM** : 2318043

**Prodi** : TEKNIK INFO4RMATIKA S1

**Kelompok** : 12

**PROGRAM STUDI TEKNIK INFORMATIKA S-1**

**FAKULTAS TEKNOLOGI INDUSTRI**

**INSTITUT TEKNOLOGI NASIONAL MALANG**

**2023**

**LEMBAR PERSETUJUAN**

**ALGORITMA DAN PEMROGRAMAN**

**SEMESTER GANJIL TAHUN AKADEMIK 2023/2024**

****

*Disusun Oleh*

NAMA : Sabastian Rafariza Wahyuono

NIM : 2318043

PRODI : TEKNIK INFORMATIKA S-1

|  |  |
| --- | --- |
| Mengetahui,  Kepala Laboratorium  Pemrograman Komputer  **(Ahmad Faisol, ST, MT)**  **NIP.P.1031000431** | Menyetujui,  Dosen Pembimbing  **(Joseph Dedy Irawan, ST, MT)**  **NIP.197404162005011002** |

**PROGRAM STUDI TEKNIK INFORMATIKA S-1**

**FAKULTAS TEKNOLOGI INDUSTRI**

**INSTITUT TEKNOLOGI NASIONAL MALANG**

**2023**

# KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Tuhan Yang Maha Esa, karena atas berkah rahmat dan karunia-Nya sehingga kami dapat menyelesaikan Laporan Praktikum Mata Kuliah, guna persyaratan dalam menempuh mata kuliah.

Laporan ini disusun berdasarkan percobaan dan teori dasar yang ada dalam buku panduan praktikum, teori yang diperoleh praktikan dari perkuliahan, dan tidak lupa yaitu Internet sehingga praktikan dapat menambah tidak hanya menguasai teori saja namun juga memahami serta mengaplikasikannya.

Terwujudnya laporan ini, tentunya tidak lepas dari bantuan-bantuan yang telah kami terima. Pada kesempatan ini, kami menyampaikan terima kasih yang sebesar-besarnya kepada yang terhormat:

1. Bapak Joseph Dedy Iriawan, ST. MT. selaku dosen pembimbing Praktikum Algoritma dan Pemrograman.
2. Bapak Joseph Dedy Irawan, ST, MT. dan Bapak Renaldi Primaswana Prasetya, S.Kom, M.Kom. selaku dosen mata kuliah Algoritma dan Pemrograman.
3. Bapak Ahmad Faisol, ST, MT. selaku Ketua Pelaksana Praktikum Algoritma dan Pemrograman Program Studi Teknik Informatika ITN Malang.
4. Instruktur Lab. Pemrograman Komputer Teknik Informatika yang telah memberi petunjuk kepada penulis selama pelaksanaan praktikum.
5. Rekan-rekan yang telah membantu dalam pelaksanaan dan penyelesaian laporan ini.

Dalam menyusun laporan ini kami menyadari bahwa laporan ini masih memiliki kekurangan, karena itu segala kritik dan saran yang membangun akan kami nanti demi perbaikan penyusunan laporan selanjutnya.

Harapan kami laporan praktikum ini bermanfaat bagi penulis sendiri maupun pembaca sekalian.

|  |
| --- |
| Malang, Desember 2023 |
|  |
|  |
|  |
| Penulis |

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc153276243)

[DAFTAR ISI ii](#_Toc153276244)

[DAFTAR GAMBAR v](#_Toc153276245)

[DAFTAR TABEL vii](#_Toc153276246)

[BAB I PENDAHULUAN 1](#_Toc153276247)

[I.1 Latar Belakang 1](#_Toc153276248)

[I.2 Rumusan Masalah 2](#_Toc153276249)

[I.3 Tujuan 2](#_Toc153276250)

[I.4 Manfaat 2](#_Toc153276251)

[BAB II Pengenalan C++ 3](#_Toc153276252)

[II.1 Landasan Teori 3](#_Toc153276253)

[II.2 Langkah – Langkah Praktikum 9](#_Toc153276254)

[II.3 Latihan Praktikum ke-1: Program Mencetak NIM 9](#_Toc153276255)

[II.4 Tugas Praktikum ke-1: Program Tahun Ajaran 10](#_Toc153276256)

[II.5 Tugas Praktikum ke-2: Program Mencetak Data Mahasiswa 10](#_Toc153276257)

[II.6 Tugas praktikum ke-3: Program Mencetak Identitas Mahasiswa 11](#_Toc153276258)

[II.7 Kesimpulan 13](#_Toc153276259)

[BAB III Operator dan Statement 14](#_Toc153276260)

[III.1 Landasan Teori 14](#_Toc153276261)

[A. Operator 14](#_Toc153276262)

[B. Operator I/O 21](#_Toc153276263)

[III.2 Langkah – Langkah Praktikum 21](#_Toc153276264)

[III.3 Latihan Praktikum ke-1: Program Operator Aritmatika 21](#_Toc153276265)

[III.4 Latihan Praktikum ke-2:Program Perkalian x dan y 22](#_Toc153276266)

[III.5 Latihan Praktikum ke-3: Program Operasi Bitewise 23](#_Toc153276267)

[III.6 Tugas Praktikum ke-1: Program Menghitung Volume Bola 24](#_Toc153276268)

[III.7 Tugas Praktikum ke-2: 25](#_Toc153276269)

[III.8 Tugas praktikum ke-3: Program Menghitung Luas Segitiga 25](#_Toc153276270)

[III.9 Kesimpulan 26](#_Toc153276271)

[BAB IV Percabangan 27](#_Toc153276272)

[IV.1 Landasan Teori 27](#_Toc153276273)

[IV.2 Langkah – Langkah Praktikum 30](#_Toc153276274)

[IV.3 Latihan Praktikum ke-1: (If Else) 30](#_Toc153276275)

[IV.4 Latihan Praktikum ke-2: (If Else) 31](#_Toc153276276)

[IV.5 Latihan Praktikum ke-3: (If Else) 32](#_Toc153276277)

[IV.6 Latihan Praktikum ke-4: (Switch case) 33](#_Toc153276278)

[IV.7 Latihan Praktikum ke-5: (Switch case) 35](#_Toc153276279)

[IV.8 Tugas Praktikum ke-1: (If Else) 38](#_Toc153276280)

[IV.9 Tugas Praktikum ke-2: (If Else) 39](#_Toc153276281)

[IV.10 Tugas praktikum ke-3: (If Else) 40](#_Toc153276282)

[IV.11 Tugas Praktikum ke-4: (Switch case) 41](#_Toc153276283)

[IV.12 Tugas Praktikum ke-5: (Switch case) 44](#_Toc153276284)

[IV.13 Tugas praktikum ke-6 (Switch case) 45](#_Toc153276285)

[IV.14 Kesimpulan 46](#_Toc153276286)

[BAB V Perulangan 47](#_Toc153276287)

[V.1 Landasan Teori 47](#_Toc153276288)

[V.2 Langkah – Langkah Praktikum 50](#_Toc153276289)

[V.3 Latihan Praktikum ke-1: (For) 51](#_Toc153276290)

[V.4 Latihan Praktikum ke-2: (For) 51](#_Toc153276291)

[V.5 Latihan Praktikum ke-3 (For) 53](#_Toc153276292)

[V.6 Latihan Praktikum ke-4 (While) 54](#_Toc153276293)

[V.7 Latihan Praktikum ke-5 (Do While): 55](#_Toc153276294)

[V.8 Latihan Praktikum ke-6 (Nested Loop): 57](#_Toc153276295)

[V.9 Latihan Praktikum ke-7 (Nested Loop) 59](#_Toc153276296)

[V.10 Tugas Praktikum ke-1 ( For ): 60](#_Toc153276297)

[V.11 Tugas Praktikum ke-2 (For): 61](#_Toc153276298)

[V.12 Tugas praktikum ke-3 (For) 63](#_Toc153276299)

[V.13 Tugas Praktikum ke-4 (While) 63](#_Toc153276300)

[V.14 Tugas Praktikum ke-5 ( Do While): 64](#_Toc153276301)

[V.15 Tugas praktikum ke-6 (While) 65](#_Toc153276302)

[V.16 Tugas Praktikum ke-7 (Nested Loop): 66](#_Toc153276303)

[V.17 Tugas Praktikum ke-8 (Nested Loop) 67](#_Toc153276304)

[V.18 Tugas praktikum ke-9 (Nested Loop) 69](#_Toc153276305)

[V.19 Kesimpulan 70](#_Toc153276306)

[BAB VI Function (Non-Void) dan Prosedure (Void) 71](#_Toc153276307)

[VI.1 Landasan Teori 71](#_Toc153276308)

[VI.2 Langkah – Langkah Praktikum 75](#_Toc153276309)

[VI.3 Latihan Praktikum ke-1: Program menghitung Volume Bola 75](#_Toc153276310)

[VI.4 Latihan Praktikum ke-2: Program Mencetak Biodata Mahasiswa 76](#_Toc153276311)

[VI.5 Latihan Praktikum ke-3: Program Sub-rutin Menentukan nilai fungsi 77](#_Toc153276312)

[VI.6 Tugas Praktikum ke-1: 78](#_Toc153276313)

[VI.7 Tugas Praktikum ke-2: 80](#_Toc153276314)

[VI.8 Tugas praktikum ke-3 81](#_Toc153276315)

[VI.9 Kesimpulan 82](#_Toc153276316)

[BAB VII Array 83](#_Toc153276317)

[VII.1 Landasan Teori 83](#_Toc153276318)

[VII.2 Langkah – Langkah Praktikum 85](#_Toc153276319)

[VII.3 Latihan Praktikum ke-1: 85](#_Toc153276320)

[VII.4 Latihan Praktikum ke-2: 86](#_Toc153276321)

[VII.5 Latihan Praktikum ke-3 87](#_Toc153276322)

[VII.6 Tugas Praktikum ke-1: 88](#_Toc153276323)

[VII.7 Tugas Praktikum ke-2: 90](#_Toc153276324)

[VII.8 Tugas praktikum ke-3 92](#_Toc153276325)

[VII.9 Kesimpulan 94](#_Toc153276326)

[KESIMPULAN 95](#_Toc153276327)

[DAFTAR PUSTAKA 96](#_Toc153276328)

# DAFTAR GAMBAR

[Gambar 2.1 Hasil Runing Program NIM 9](#_Toc153276329)

[Gambar 2.2 Hasil Running Program Tahun Ajaran 10](#_Toc153276330)

[Gambar 2.3 Hasil *Running* Program Cetak Data 11](#_Toc153276331)

[Gambar 2.4 Hasil *Running* Program Identitas Mahasiswa 12](#_Toc153276332)

[Gambar 3.1 Hasil *Running* Program Operator Aritmatika 22](#_Toc153276333)

[Gambar 3.2 Hasil *Running* Program Perkalian x dan y 22](#_Toc153276334)

[Gambar 3.3 Hasil *Running* Program Operasi Bitewise 23](#_Toc153276335)

[Gambar 3.4 Hasil *Running* Program Volume Bola 24](#_Toc153276336)

[Gambar 3.5 Hasil *Running* Program Konversi Suhu 25](#_Toc153276337)

[Gambar 3.6 Hasil *Running* Program Menghitung Luas Segitiga 26](#_Toc153276338)

[Gambar 4.1 Contoh *Running* Program Login 30](#_Toc153276339)

[Gambar 4.2 Hasil *Running* Program Menentukan Bulan 32](#_Toc153276340)

[Gambar 4.3 Hasil *Running* Menghitung Segitiga 33](#_Toc153276341)

[Gambar 4.4 Contoh *Running* Latihan 4 34](#_Toc153276342)

[Gambar 4.5 Hasil *Running* Program Tiket Bioskop 37](#_Toc153276343)

[Gambar 4.6 Hasil *Running* Program Menghitung BB 38](#_Toc153276344)

[Gambar 4.7 Hasil *Running* Program Grading Nilai 39](#_Toc153276345)

[Gambar 4.8 Hasil *Running* Program Menentukan Ganjil Genap 41](#_Toc153276346)

[Gambar 4.9 Hasil *Running* Program Pemesanan Kamar 43](#_Toc153276347)

[Gambar 4.10 Contoh *Running* Progam Pembelian Bahan Bakar 44](#_Toc153276348)

[Gambar 4.11 Hasil *Running* Program Memilih Buku 45](#_Toc153276349)

[Gambar 5.1 Hasil *Running* Program NIM Ganjil 51](#_Toc153276350)

[Gambar 5.2 Hasil *Running* Program Angka Ganjil 52](#_Toc153276351)

[Gambar 5.3 Hasil *Running* Perulangan For Angka dan Huruf 53](#_Toc153276352)

[Gambar 5.4 Hasil *Running* Program Fibonaci 54](#_Toc153276353)

[Gambar 5.5 Hasil *Running* Program Perulangan Angka dan Huruf 56](#_Toc153276354)

[Gambar 5.6 Hasil *Running* Program Mencetak Pola 58](#_Toc153276355)

[Gambar 5.7 Hasil *Running*  Piramida Bintang dan Urutan Angka 59](#_Toc153276356)

[Gambar 5.8 Hasil *Running* Program Perpangkatan Tiga 61](#_Toc153276357)

[Gambar 5.9 Hasil *Running* Program Penjualan Kendaraan 62](#_Toc153276358)

[Gambar 5.10 Hasil *Running* Progam Input Data 63](#_Toc153276359)

[Gambar 5.11 Hasil *Running* Program angka kelipatan 64](#_Toc153276360)

[Gambar 5.12 Hasil *Running* Program Konversi Jam 65](#_Toc153276361)

[Gambar 5.13 Hasil *Running* Program Wajib Membayar Hutang 66](#_Toc153276362)

[Gambar 5.14 Hasil *Running* Pola Piramida Terbalik 67](#_Toc153276363)

[Gambar 5.15 Program *Running* bangun datar NIM Ganjil 68](#_Toc153276364)

[Gambar 5.16 Hasil *Running* Pencetak Pola 69](#_Toc153276365)

[Gambar 6.1 Hasil *Running* Program Menghitung Volume 75](#_Toc153276366)

[Gambar 6.2 Hasil Running Program Biodata Mahasiswa 76](#_Toc153276367)

[Gambar6.3 Hasil Running Program menentukan nilai fungsi 77](#_Toc153276368)

[Gambar 6.4 Hasil Running Program volume bangun ruang 79](#_Toc153276369)

[Gambar 6.5 Hasil Running Program Membuat Persegi 81](#_Toc153276370)

[Gambar 6.6 Program Perulangan *Input* Data 82](#_Toc153276371)

[Gambar 9.1 Hasil Running Program Menambahkan Array 86](#_Toc153276372)

[Gambar 9.2 Hasil Running Program Data Tinggi Badan 87](#_Toc153276373)

[Gambar 9.3 Hasil Running Program Grafik Poin 88](#_Toc153276374)

[Gambar 9.4 Hasil Running Program Pendataan Nilai Mahasiswa 90](#_Toc153276375)

[Gambar 9.5 Hasil Running Program Pendataan Tekanan Darah 92](#_Toc153276376)

[Gambar 8.6 Running Program Nilai Rata - Rata Array 93](#_Toc153276377)

# DAFTAR TABEL

[Tabel 2.1 Tabel Penaamaan Variabel 7](#_Toc153276378)

[Tabel 3.1 Tabel operator aritmatika : 15](#_Toc153276379)

[Tabel 3.2 Operator relasional 16](#_Toc153276380)

[Tabel 3.3 Operator majemuk 17](#_Toc153276381)

[Tabel 3.4 Operator AND 18](#_Toc153276382)

[Tabel 3.5 Operator OR 19](#_Toc153276383)

[Tabel 3.6 Operator XOR 19](#_Toc153276384)

[Tabel 3.7 Operator NOT 20](#_Toc153276385)

# BAB I PENDAHULUAN

1. **Latar Belakang**

C++ adalah bahasa pemrograman yang dibuat oleh Bjarne Stroustrup yang merupakan perkembangan dari bahasa C pada awal tahun 1970-an. Bahasa itu diturunkan dari bahasa sebelumnya, yaitu B. Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem UNIX. Pada perkembanganya, versi ANSI (American National Standart Institute). Bahasa pemrograman C menjadi versi dominan, meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk sistem embedded. Bjarne Stroustrup pada Bel labs pertama kali mengembangkan C++ pada awal 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (low level coding). Pada C++ ditambahkan konsep-konsep baru seperti class dengan sifat-sifatnya seperti inheritance dan overloading. Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (Object Oriented Programming).

Dev-C++ merupakan sebuah IDE (Integrated Development Environment) C / C++ yang sudah dilengkapi dengan TDM-GCC Compiler (bagian dari GNU Compiler Collection / GCC). Dev-C++ merupakan IDE gratis dan full fiture yang didistribusikan dibawah lisensi GNU General Public License untuk pemrograman C dan C++. Seperti yang sudah disebutkan diatas IDE merupakan lembar kerja terpadu untuk pengembangan program.

Program C++ menghasilkan program yang sederhana dan mudah digunakan, sehingga bisa digunakan untuk memecahkan suatu masalah pada kehidupan sehari-hari.

1. **Rumusan Masalah**
2. Bagaimana cara melakukan pemrograman dengan menggunakan Bahasa C++?
3. Bagaimana cara membuat program dengan menggunakan aplikasi Dev C++?
4. Bagaimana mengimplementasikan pemrograman dengan bahasa C++ dalam kehidupan sehari-hari?
5. **Tujuan**
6. Praktikan mampu memahami pemograman dengan menggunakan bahasa C++
7. Praktikan mampu membuat program dengan menggunakan aplikasi Dev C++
8. Praktikan mampu mengimplementasikan pemrograman dengan Bahasa C++ dalam kehidupan sehari-hari.
9. **Manfaat**
10. Praktikan dapat mengerti dan memahami bahasa pemrograman C++.
11. Praktikan dapat membuat program dengan bahasa pemrograman C++.
12. Praktikan dapat membuat program dan mengaplikasikannya dalam kehidupan sehari-hari.

**BAB II** **Pengenalan C++**

|  |  |  |
| --- | --- | --- |
| **Durasi Pertemuan** | **:** | 120 Menit |
| **Tujuan Praktikum** | **:** | 1. Mampu memahami pengertian Algoritma dan Pemrograman dengan menggunakan C++ 2. Mampu memahami konsep ‘Include’ pada C++ 3. Mampu memahami bentuk umum dari pemrograman C++ 4. Mampu memahami variabel dan tipe data pada C++ 5. Mampu membuat program sederhana dari pemahaman sebelumnya |
| **Alat / bahan** | **:** | 1. Seperangkat *computer*. 2. Perangkat lunak: Dev C++. 3. Modul Praktikum Algoritma dan Pemrogramman 2023. |

1. **Landasan Teori**
2. Pengertian C++

Bahasa Pemrograman C++ adalah bahasa Pemrograman Komputer Tingkat Tinggi (*High Level Language*), tapi C++ juga dimungkinkan untuk menulis Bahasa Pemrograman Tinggkat Rendah (*Low Level Language*) di dalam pengkodingan

C++ merupakan penyempurnaan dari bahasa pemrograman sebelumnya yaitu bahasa C, oleh Bjarne Stroustrup pada tahun 1980. Awal C++ mempunyai nama yaitu “*C with Classes*” dan berganti nama menjadi C++ pada tahun 1983. Bjarne Stroustrup membuat bahasa pemrograman C++ dengan tambahan fasilitas, yang sangat berguna pada tahun itu sampai sekarang, yaitu bahasa pemrograman yang mendukung OOP (*Object Oriented Programming*).

1. Header dan Fungsi Main

Header adalah tempat kita memasukan *library* yang ada di c++ ke dalam program, *library* tersebut berisi deklarasi berbagai fungsi yang dibutuhkan oleh program baik itu sebagai standar Input/Output maupun sebagai syntax.

Apabila kita lihat pada program baris pertama terdapat #include <iostream>. Tanda ‘#’ adalah ***Preprocessor directive***. *Preprocessor directive* sendiri adalah intruksi untuk *compiler* memerintahkan untuk memproses suatu arahan atau informasi sebelum memulai melakukan kompilasi pada program yang telah kita tulis, misal untuk memasukan *file* *library*, dll. Sebagai contoh, jika dilihat dari program diatas kita akan memasukan (include) *library* iostream ke dalam program kita, *library* iostream sendiri adalah header yang di butuhkan untuk melakukan input dan output, dll. Jika di perhatikan terdapat kode using namespace std, *Using* adalah sebuah kata kunci yang digunakan untuk memperkenalkan suatu *Namespace* kepada suatu ruang lingkup, dengan ini kita tidak diwajibkan lagi untuk menyertakan nama dari Namespace dan Scope Operator :: setiap kali mencoba untuk mengakses entitas dari Namespace tersebut. Selanjutnya fungsi main(), main() adalah blok program utama kita yang di tulis diantara { hingga }. Fungsi inilah yang akan dipanggil pertama kali pada saat eksekusi program. Apabila ada fungsi lain yang dibuat, maka fungsi tersebut akan dijankan ketika dipanggil di fungsi utama. Hal ini karena bahasa C/C++ merupakan bahasa procedural yang menerapkan konsep runtutan (program dieksekusi baris-perbaris dari atas ke bawah secara berurutan).

1. Return 0;

Intruksi return menyebabkan fungsi main() berakhir dan mengembalikan kode yang mengikuti instruksi tersebut, dalam kasus ini 0. Ini merupakan cara yang paling sering digunakan untuk mengakhiri program.

1. Komentar Program

Komentar adalah bagian dari kode program yang tidak ikut dibaca pada proses kompilasi. Dengan kata lain, komentar tidak akan mempengaruhi jalannya program. Dalam C++ terdapat 2 jenis komentar, yaitu :

1. Menggunakan tanda //

Tanda ini digunakan untuk menuliskan komentar yang banyaknya hanya satu baris.

|  |
| --- |
| // ini adalah komentar untuk satu baris |

1. Menggunakan tanda /\*\*/

Berbeda dengan sebelumnya, tanda ini dapat digunakan untuk menuliskan komentar yang banyaknya satu baris atau lebih. Komentar ini dimulai dari tanda **/\*** sampai ditemukan tanda **\*/**.

|  |
| --- |
| /\*ini adalah komentar panjang  yang banyaknya  lebih dari satu baris\*/ |

1. Tanda Semicolon

Tanda semicolon “ ; ” digunakan untuk mengakhiri sebuah pernyataan. Setiap pernyataan harus diakhiri dengan sebuah tanda semicolon.

1. Input dan Output

Statemen Input adalah Statemen / fungsi yang digunakan untuk menerima data dari input keyboard, contoh: cin (character in).

|  |
| --- |
| cin>>a; |

Selain cin ada juga getline, getline digunakan jika ingin mengambil input lebih dari 1 kata. getline memiliki struktur :

|  |
| --- |
| getline(cin,nama\_identifier); |

Statemen Output adalah Statemen yang digunakan untuk menampilkan data ke layar monitor., contoh: cout (character out).

|  |
| --- |
| cout<<”RPL”; |

1. Jenis Identifier

Identifier atau pengenal adalah suatu nama yang biasa dipakai dalam pemrograman untuk menyatakan variabel, konstanta, tipe data, dan fungsi. Identifier yang berperan sebagai variabel dan konstanta berfungsi untuk menampung sebuah nilai yang digunakan dalam program.

1. Konstanta

Konstanta adalah jenis identifier yang bersifat konstan atau tetap, artinya nilai dari konstanta di dalam program tidak berubah. Terdapat dua cara untuk membuat sebuah konstanta, yaitu dengan menggunakan preposesor directive #define dan menggunakan kata kunci const.

1. Menggunakan kata kunci *const*

Mendefinisikan sebuah konstanta dapat dilakukan dengan menggukan kata kunci const. Berikut ini adalah bentuk umumnya.

|  |
| --- |
| const tipe\_data nama\_konstanta = nilai; |

1. Menggunakan *Preprocessor Directive* #*define*

Keuntungan menggunakan #define apabila dibandingkan dengan const adalah kecepatan kompilasi, karena sebelum kompilasi dilaksanakan, kompiler pertama kali mencari symbol #define (oleh sebab itu mengapa # dikatakan preprocessor directive). #define adalah preprocessor directive, oleh sebab itu tidak diakhiri dengan tanda semicolon (;).

1. Variabel

Variabel adalah sebuah identifier yang mempunyai nilai dinamis. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variabel bisa diubah-ubah sesuai kebutuhan.

1. Deklarasi variabel

Untuk menggunakan variabel pada C++, kita harus mendeklarasikan tipe data yang akan digunakan. Sintaks penulisan deklarasi variabel adalah dengan menuliskan tipe data yang akan digunakan diikuti dengan identifier yang benar. Pengertian deklarasi disini berarti mengenalkan sebuah pengenal ke program dan menentukan jenis data yang disimpam didalamnya.

Bentuk pendeklarasian variabel :

|  |
| --- |
| tipe\_data nama\_variabel; |

Contoh :

|  |
| --- |
| int A;  float a;  char kata; |

Jika akan menggunakan tipe data yang sama untuk beberapa identifier maka dapata dituliskan dengan menggunakan tanda koma.

|  |
| --- |
| int a,b,c;  float d,e,f; |

Nama dari suatu variabel dapat ditentukan sendiri oleh program dengan aturan sebagai berikut :

1. Tidak boleh dimulai dengan karakter non huruf
2. Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Bahasa C++ bersifat case-sensitive, yang artinya huruf besar dan kecil dianggap berbeda. Jadi antara nim, NIM, dan Nim dianggap berbeda.
3. Tidak boleh mengandung spasi.
4. Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah (underscore). Yang termasuk simbol khusus yang tidak diperbolehkan antara lain : **$, ?, %, #, !, &, \*, (, ), -, +, =, dsb**.
5. Panjang bebas, tetapi hanya 32 karakter pertama yang terpakai.

#### Tabel 2.1 Tabel Penaamaan Variabel

|  |  |
| --- | --- |
| Penamaan yang Benar | Penamaan yang Salah |
| namasiswa | nama siswa (salah karena menggunakan spasi) |
| XY12 | 12X (salah karena dimulai dengan angka) |
| harga\_total | harga.total (salah karena menggunakan karakter .) |
| JenisMotor | Jenis Motor (salah karena menggunakan spasi) |
| alamatrumah | for (salah karena menggunakan kata kunci bahasa pemrograman) |

1. Inisialisasi variabel

Inisialisasi dapat didefinisikan sebagai proses pengisian nilai awal (nilai default) ke dalam suatu variabel. Dalam C++, pengisian nilai dilakukan dengan menggunakan operator sama dengan (=).

Bentuk pernyataan yang digunakan untuk memberikan nilai ke variabel yang telah dideklarasikan atau didefinisikan :

|  |
| --- |
| nama\_variabel = nilai\_awal; |

Adakalanya dalam penulisan program, variabel langsung diberi nilai setelah didefinisikan. Sebagai contoh :

|  |
| --- |
| int A;  A = 10; |

Dua pernyataan seperti diatas sebenarnya dapat disingkat melalui pendefinisian yang disertai penugasan nilai, sebagai berikut :

|  |
| --- |
| int A = 10; |

1. Tipe Data
2. Tipe Data Dasar

contoh dibawah ini, tipe data dapat dirubah (*type cast)* dengan cara dibawah ini :

|  |
| --- |
| float x = 3.345;  int p = int(x); |

Maka nilai P adalah 3 (terjadi *truncating*).

1. Karakter dan String Literal

String adalah tipe data text yang memungkinkan kita menyimpan nilai dengan bentuk text, kumpulan dari karakter. Berbeda dengan char gabungan dari karakter. Perbedaannya dengan tipe data *char* adalah mengenai penyimpanannya. Char menyimpan hanya beberapa karakter saja, sedangkan String dapat menyimpan banyak karakter, kata, dan kalimat sekaligus.

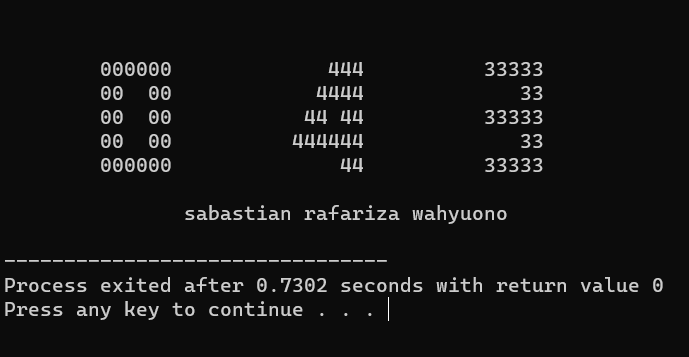
1. Escape Sequence Characters

Escape sequence characters adalah urutan sebuah table, penulisan kode yang mewakili sebuah karakter yang tidak dapat diwakili secara langsung seperti newline yang diwakili dengan tanda \n dan petik dua dengan menggunakan tanda \".

1. **Langkah – Langkah Praktikum**
2. Buka Aplikasi Dev C++.
3. Buka Modul Praktikum Algoritma dan Pemrogramman.
4. Menjalankan script dari modul
5. **Latihan Praktikum ke-1: Program Mencetak NIM**
6. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int nilai;  int main(){  cout<<"\v";  coklut<<"\v";  cout<<"\t000000"<<"\t\t 444"<<"\t\t33333\n";  cout<<"\t00 00"<<"\t\t 4444"<<"\t\t 33\n";  cout<<"\t00 00"<<"\t\t 44 44"<<"\t\t33333\n";  cout<<"\t00 00"<<"\t\t444444"<<"\t\t 33\n";  cout<<"\t000000"<<"\t\t 44"<<"\t\t33333\n";  cout<<"\v";  cout<<"\t sabastian rafariza wahyuono\n";  return 0;  } |

1. Hasil *Running*



### Gambar 2.1 Hasil Runing Program NIM

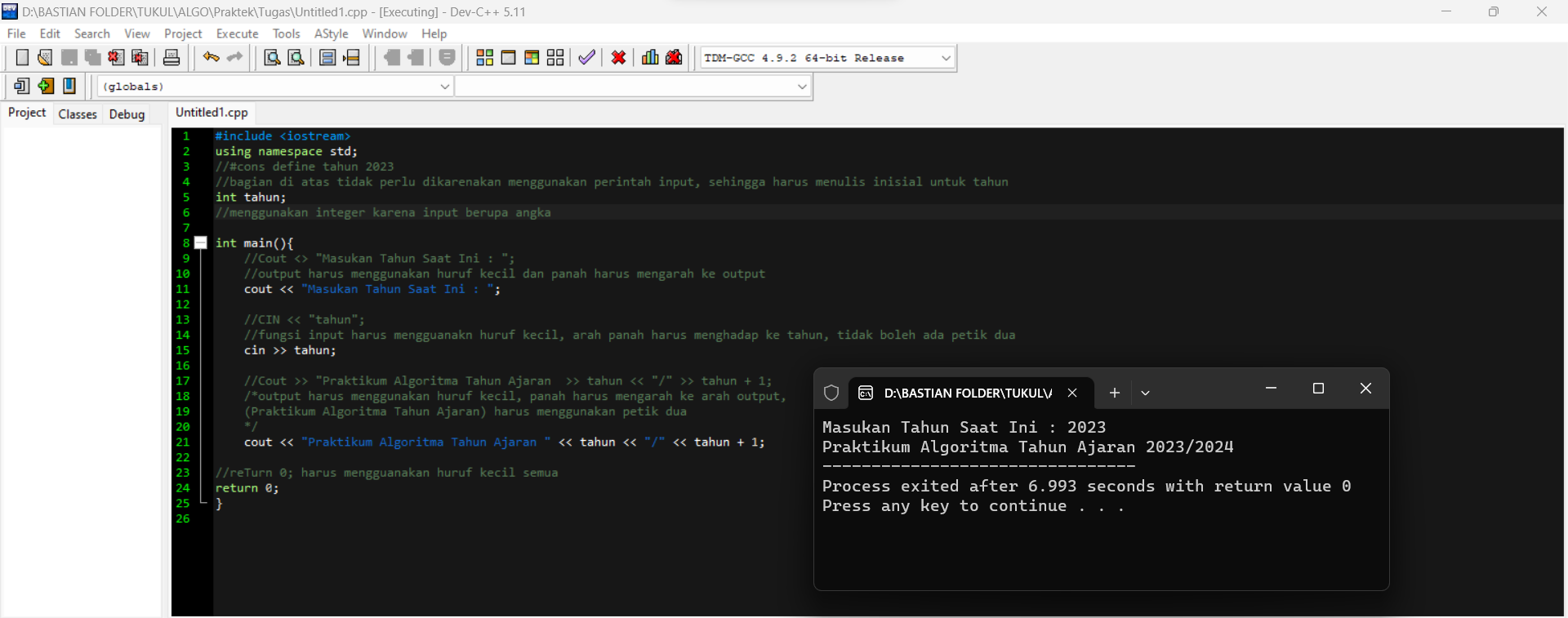
1. Analisa

Menggunakan perintah cout untuk menampilkan keluaran atau output. Kemudian menggunakan *escape sequence characters* \v untuk tab vertikal, \t untuk tab horizontal, \n untuk baris baru (*new line)*.

1. **Tugas Praktikum ke-1: Program Tahun Ajaran**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int tahun;  int main(){    cout << "Masukan Tahun Saat Ini : ";    cin >> tahun;    cout << "Praktikum Algoritma Tahun Ajaran " << tahun << "/" << tahun + 1;    return 0;  } |

1. Hasil *Running*



### Gambar 2.2 Hasil Running Program Tahun Ajaran

1. Analisa

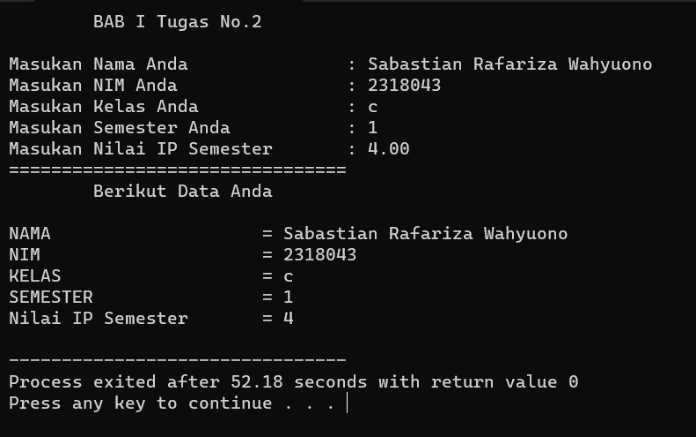
Menggunakan tipe data *integer* dengan variabel tahun. Menggunakan perintah cin untuk input variabel tahun, kemudian mengeluarkan variabel tahun dan mengeluarkan hasil tahun ditambah dengan satu

.

1. **Tugas Praktikum ke-2: Program Mencetak Data Mahasiswa**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){    string nama;  int nim;  char kelas;  int semester;  double ip;  cout<<"\tBAB I Tugas No.2\n\n";  cout << "Masukan Nama Anda\t\t: ";  getline (cin, nama);  cout << "Masukan NIM Anda\t\t: ";  cin >> nim;  cout << "Masukan Kelas Anda\t\t: ";  cin >> kelas;  cout << "Masukan Semester Anda\t\t: ";  cin >> semester;  cout << "Masukan Nilai IP Semester\t: ";  cin >> ip;  cout<<"================================\n\tBerikut Data Anda\n\n";  cout << "NAMA\t\t\t= "<<nama<<endl;  cout << "NIM\t\t\t= "<<nim<<endl;  cout << "KELAS\t\t\t= "<<kelas<<endl;  cout << "SEMESTER\t\t= "<<semester<<endl;  cout << "Nilai IP Semester\t= "<<ip<<endl;  } |

1. Hasil Running



### Gambar 2.3 Hasil *Running* Program Cetak Data

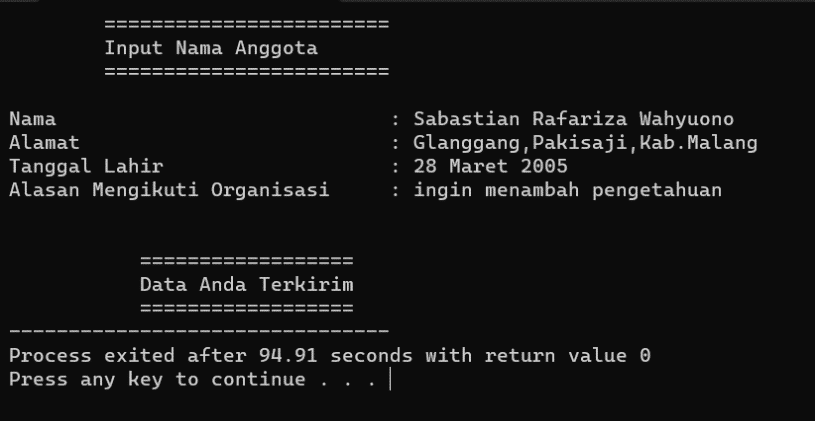
1. Analisa

Menggunakan tipe data *string* dengan variabel nama, *char* dengan variabel kelas, *integer* dengan variabel nim dan semester, *double* dengan variabel ip. Kemudian menggunakan cout untuk menampilkan keterangan dan menggunakan cin untuk memasukan variabel nama, nim, kelas, semester, dan ip

1. **Tugas praktikum ke-3: Program Mencetak Identitas Mahasiswa**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  string na,al,tgl,als;  int main(){  cout<<"\t========================\n\tInput Nama Anggota\n\t========================\n"<<endl;  cout<<"Nama\t\t\t\t: ";  getline(cin,na);  cout<<"Alamat\t\t\t\t: ";  getline(cin,al);  cout<<"Tanggal Lahir\t\t\t: ";  getline(cin,tgl);  cout<<"Alasan Mengikuti Organisasi\t: ";  getline(cin,als);    cout<<"\n\t\n\t ==================\n\t Data Anda Terkirim\n\t ==================";  return 0;  } |

1. Hasil Running



### Gambar 2.4 Hasil *Running* Program Identitas Mahasiswa

1. Analisa

Menggunakan tipe data *string* dengan variabel na, al, tgl, dan als. Menggunakan perintah cout untuk mengeluarkan keterangan dan menggunakan perintah *getline* (cin, nama\_variabel) karena *input* menggunakan spasi, Jika tidak menggunakan perintah *getline* (cin, nama\_variabel) maka akan *error* jika diberi spasi.

1. **Kesimpulan**
2. C++ merupakan bahasa pemrograman yang tinggi dan sensitif terhadap sebuah kalimat,kata,tanda baca,jadi kita haru lebih berkonsentrasi dalam mengerjakan sesuatu dialam bahasa pemrograman C++.
3. C++ merupakan penyempurnaan dari bahasa pemrograman sebelumnya yaitu bahasa C.
4. Header adalah tempat kita memasukan library yang ada di c++ ke dalam program, *library* tersebut berisi deklarasi berbagai fungsi yang dibutuhkan oleh program baik itu sebagai standar Input/Output maupun sebagai syntax.

**BAB III  
Operator dan Statement**

|  |  |  |
| --- | --- | --- |
| **Durasi Pertemeuan** | **:** | 120 Menit |
| **Tujuan Praktikum** | **:** | 1. Mampu memahami konsep Operator dan I/O pada C++ 2. Mampu membuat program sederhana dari pemahaman sebelumnya |
| **Alat / bahan** | **:** | 1. Seperangkat *computer*. 2. Perangkat lunak: Dev C++. 3. Modul Praktikum Algoritma dan Pemrogramman 2023. |

1. **Landasan Teori**
2. Operator

Operator merupakan simbol yang biasa dilibatkan dalam program untuk melakukan suatu operasi atau manipulasi. Dengan operator, kita dapat melakukan operasi perhitungan, perbandingan, manipulasi bit, dan lain-lain.

Sebagai pengenalan, terdapat istilah yang harus diketahui dalam bekerja dengan operator. Sebagai contoh :

|  |
| --- |
| A = 4 + 12; |

Keterangan :

1. A disebut variable
2. = disebut operator assignment
3. 4 dan 12 disebut operand
4. 4 + 12 disebut ekspresi
5. + disebut operator aritmatika (penambahan)
6. A = 4 + 12 disebut statemen aritmatika

Pada dasarnya, ada tiga jenis operator dalam pemrograman C++

1. *Unary*

Sifat *unary* pada operator adalah melibatkan sebuah *operand* pada suatu operasi aritmatika. Contoh: -5

1. *Binary*

Sifat *binary* pada operator adalah melibatkan 2 buah *operand* pada suatu operasi artimatika. Contoh: 4 + 8

1. *Ternary*

Sifat *ternary* pada operator adalah melibatkan 3 buah *operand* pada suatu operasi aritmatika. Contoh: 10 % 3 + 2; a > b ? 1 : 0;

Namun operator dibagi menjadi beberapa jenis, sebagai berikut :

* + - 1. Operator Assingment

Operator assignment (penugasan) adalah operator yang digunakan untuk memasukkan atau mengisikan nilai ke dalam suatu variabel. Dalam bahasa C++ digunakan operator berupa tanda sama dengan ( “ = ” ).

* + - 1. Operator Aritmatika

Operator ini digunakan untuk melakukan sebuah opearasi penghitungan atau aritmatika. Tugasnya untuk memecahkan masalah matematika.

#### Tabel 3.1 Tabel operator aritmatika :

|  |  |  |
| --- | --- | --- |
| Operator | Deskripsi | Contoh |
| + | Penjumlahan | X + Y |
| - | Pengurangan | X - Y |
| \* | Perkalian | X \* Y |
| / | Pembagian | X / Y |
| % | Sisa pembagian Integer (Modulus) | X % Y |

Pada operator penjumlahan, pengurangan, perkalian, dan pembagian dilakukan operasi seperti biasa. Sedangkan, yang di maksud dengan sisa pembagian adalah sisa dari hasil pembagian bukan hasil dari pembagian.

* + - 1. Operator Relasional

Operator relasional atau hubungan digunakan untuk membandingkan hubungan antara dua buah operand. Pada operator ini menghasilkan kondisi benar atau salah.

#### Tabel 3.2 Operator relasional

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Deskripsi | Contoh | |
| == | Sama dengan  (bukan assigment) | 5==5 | Benar, karena 5 sama dengan 5 |
| != | Tidak sama dengan | 5!=2 | Benar, karena 5 tidak sama dengan 2 |
| > | Lebih besar | 5 > 2 | Benar, karena 5 lebih besar dari 2 |
| < | Lebih Kecil | 5 < 2 | Tidak, karena 5 lebih besar dari 2 |
| >= | Lebih besar atau sama dengan | 5>=5 | Benar, karena 5 sama dengan 5 |
| <= | Lebih kecil atau sama dengan | 5<=2 | Tidak, karena 5 lebih besar dari 2 |

* + - 1. Operator Naik dan Turun (*Increment dan Decrement*)

Operator *increment* merupakan operator yang dapat menambahkan (menaikkan) suatu nilai. Operator i*ncrement* ini ditandai dengan tanda **“++”.** Sedangkan, operator *decrement* merupakan operator yang mengurangkan (menurunkan) suatu nilai. Operator *decrement* ini ditandai dengan tanda **“ -- ”.**

Terdapat dua jenis *increment/decrement*, yaitu *pre-increment* adalah melakukan penambahan nilai sebelum suatu variabel itu diproses, dan *post-increment* yaitu melakukan proses terlebih dahulu sebelum dilakukan penambahan nilai.

* + - 1. Operator Majemuk (Compound Assignment)

Operator majemuk digunakan untuk memendekkan penulisan operasi penugasan semacam :

|  |
| --- |
| x = x + 2 ;  y = y \* 4 ; |

Menjadi :

|  |
| --- |
| x += 2;  y \*= 4; |

#### Tabel 3.3 Operator majemuk

|  |  |  |
| --- | --- | --- |
| Operator | Keterangan | Contoh Penggunaanya |
| += | Penjumlahan | z += x sama  dengan, z = z + x |
| -= | Pengurangan | z -= x sama  dengan, z = z - x |
| \*= | Perkalian | z = x sama  dengan, z = z x |
| /= | Pembagian | z /= x sama  dengan, z = z / x |

* + - 1. Operator Bitwise (Logika)

Operator logika adalah operasi matematika yang mengoprasikan suatu nilai dalam bilangan biner. Operator bitwise akan bekerja dengan mengubah bilangan desimal menjadi bilangan biner, setelah itu akan melakukan operasi berdasarkan operator bitwise yang digunakan dan pada akhirnya nilai tersebut akan diubah menjadi bilangan desimal kembali. Operator ini hanya bisa digunakan pada operand bertipe data int atau char.

##### Shift Left (<<)

Operator Bitwise Shift Left (<<) adalah operator yang akan menggeser nilai dalam bentuk bilangan biner ke kiri

##### Shift Right (>>)

Operator Bitwise Shift Right (>>) adalah operator yang akan menggeser nilai dalam bentuk bilangan biner ke kanan.

##### AND (&)

Membandingkan berdasarkan sifat dari AND yaitu untuk menghasilkan nilai 1 (true) kedua operand harus bernilai 1(true) jika tidak akan menghasilkan nilai 0 (false).

Pertama, ubah bilangan desimal menjadi biner, dan setelah perbandingan selesai dilakukan, hasil dari perbandingan akan diubah ke bentuk bilangan desimal lagi.

contoh : 12 & 10 ?

#### Tabel 3.4 Operator AND

|  |  |  |
| --- | --- | --- |
| Operand 1 (12) | Operand 2 (10) | Hasil |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

Hasil : 1000 = 8

##### OR ( | )

Membandingkan berdasarkan sifat dari OR yaitu untuk mendapatkan nilai 1 (true) maka salah satu atau semua operand harus bernilai 1 (true), jika semua operand bernilai 0 (false) maka akan mendapatkan nilai 0 (false).

Pertama, ubah bilangan desimal menjadi biner, dan setelah perbandingan selesai dilakukan, hasil dari perbandingan akan diubah ke bentuk bilangan desimal lagi.

contoh : 12 **|** 10 ?

#### Tabel 3.5 Operator OR

|  |  |  |
| --- | --- | --- |
| Operand 1 (12) | Operand 2 (10) | Hasil |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

Hasil : 1110 = 14

##### XOR ( ^ )

Membandingkan berdasarkan sifat dari XOR yaitu untuk mendapatkan nilai 1 (true) maka kedua operand harus memiliki nilai yang berbeda, jika kedua operand memiliki nilai yang sama maka akan mendapatkan nilai 0 ( false ).

Pertama, ubah bilangan desimal menjadi biner, dan setelah perbandingan selesai dilakukan, hasil dari perbandingan akan diubah ke bentuk bilangan desimal lagi.

contoh :12 ^ 10

#### Tabel 3.6 Operator XOR

|  |  |  |
| --- | --- | --- |
| Operand 1 (12) | Operand 2 (10) | Hasil |
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

Hasil : 0110 = 6

##### NOT ( ~ )

Operator Bitwise NOT dilambangkan dengan menggunakan simbol "~", Operator ini digunakan untuk membalik nilai bit dari suatu operand. Berikut ilustrasi penggunaan Operator Bitwise NOT.

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int x = 15;  cout<<"~x = " << ~x << endl;  return 0;  } |

Contoh : ~12?

#### Tabel 3.7 Operator NOT

|  |  |
| --- | --- |
| Operand | Hasil |
| 1 | 0 |
| 0 | 1 |

Hasil : (12 + 1) \* -1 = -13*(second complimentary)*

##### Logika AND ( && )

Operator AND akan menghasilkan nilai 1 (true) jika semua operand-nya bernilai true. Namun, jika salah satu operand-nya bernilai 0 (false) maka akan menghasilkan nilai 0 (false).

##### Logika OR ( || )

Operator OR akan menghasilkan nilai 1 (true), jika salah satu operand-nya bernilai true, dan hanya akan menghasilkan nilai 0 (false) jika seluruh operand-nya bernilai 0 (false).

##### Logika NOT ( ! )

Operator NOT akan menghasilkan nilai kebalikan dari operand-nya, misal, jika operand-nya bernilai 1 (true) maka akan menghasilkan nilai 0 (false).

##### Operator Ternary

Operator ternary adalah operator yang digunakan dalam operasi yang melibatkan tiga buah operand. Adapun operator yang digunakan untuk menyatakannya adalah operator ?. berikut bentuk umum penggunaan operator ternary.

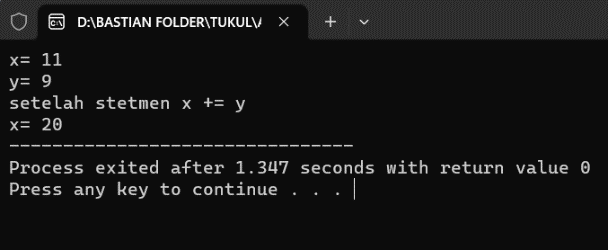
1. Operator I/O

Pada C++ terdapat 2 jenis I/O dasar, yaitu :

1. Statemen Input adalah Statemen / fungsi yang digunakan untuk menerima data dari *keyboard.* Operator yang digunakan “>>”, contoh: cin>> (character in).
2. Statemen Output adalah Statemen yang digunakan untuk menampilkan data ke layar monitor. Operator yang digunakan “<<”, contoh: cout<< (character out).
3. **Langkah – Langkah Praktikum**
4. Buka Aplikasi Dev C++.
5. Buka Modul Praktikum Algoritma dan Pemrogramman.
6. Menjalankan script dari modul
7. **Latihan Praktikum ke-1: Program Operator Aritmatika**
8. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  int x=11, y=9;  cout<<"x= "<<x<<endl;  cout<<"y= "<<y<<endl;  x+=y; //"x+=y" sama dengan "x=x+y"  cout<<"setelah stetmen x += y\n";  cout<<"x= "<<x;  return 0;  } |

1. Hasil Running



### Gambar 3.1 Hasil *Running* Program Operator Aritmatika

1. Analisa

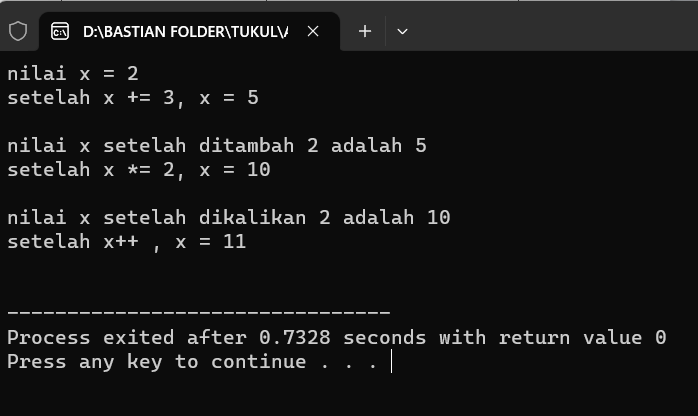
Variabel x' dan 'y' dideklarasikan dan diinisialisasi dengan nilai 11 dan 9 Kemudian Nilai dari 'x' dan 'y' dicetak ke layar. Variabel 'x' diperbarui dengan hasil penjumlahan 'x' dan 'y'. Nilai yang telah diubah dari variabel 'x' setelah operasi penjumlahan dicetak ke layar.

1. **Latihan Praktikum ke-2:Program Perkalian x dan y**

1. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  int x = 2;  cout<<"nilai x = "<<x<<endl;  x += 3;  cout<<"setelah x += 3, x = "<<x<<endl<<endl;  cout<<"nilai x setelah ditambah 2 adalah "<<x<<endl;  x\*=2;  cout<<"setelah x \*= 2, x = "<<x<<endl<<endl;  cout<<"nilai x setelah dikalikan 2 adalah "<<x<<endl;  x++;  cout<<"setelah x++ , x = "<<x<<endl<<endl;  return 0;  } |

1. Hasil Running



### Gambar 3.2 Hasil *Running* Program Perkalian x dan y

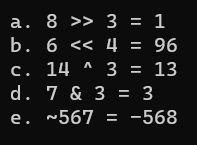
1. Analisa

Program ini mendeklarasikan variabel x dan menginisialisasinya dengan nilai 2. Kemudian nilai x ditambahkan dengan x+ atau 3 dan menegeluarkan hasil nilai x yang baru yaitu 5. Setelah itu nilai x yang sudah diperbarui dikalikan dengan x \*= 2 yang mengeluarkan hasil nilai x yaitu 10. Kemudian nilai x yang diperbarui Kembali di tambahkan dengan x++ dan mengeluarkan hasil nilai x yang baru yaitu 11.

1. **Latihan Praktikum ke-3: Program Operasi Bitewise**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  cout<<"a. 8 >> 3 = ";cout<<(8>>3)<<endl;  cout<<"b. 6 << 4 = ";cout<<(6<<4)<<endl;  cout<<"c. 14 ^ 3 = ";cout<<(14^3)<<endl;  cout<<"d. 7 & 3 = ";cout<<(7&3)<<endl;  cout<<"e. ~567 = ";cout<<(~567)<<endl;  return 0;  } |

1. Hasil Running



### Gambar 3.3 Hasil *Running* Program Operasi Bitewise

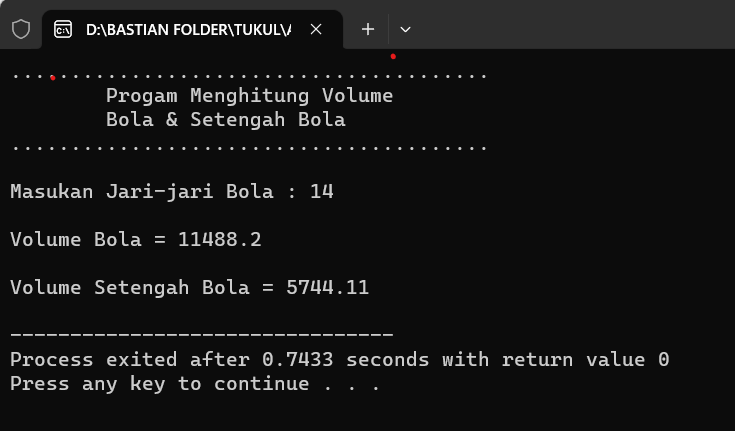
1. Analisa

Program mendeklarasikan variabel hasil\_a, hasil\_b, hasil\_c, hasil\_d, dan hasil\_e untuk menyimpan hasil operasi. Kemudian, program menghitung right shift dari 8 sebanyak 3 posisi dan hasilnya adalah 1. Kemudian left shift dari 6 sebanyak 4 posisi menghasilkan 96. Operasi XOR antara 14 dan 3 menghasilkan 13. Operasi AND antara 7 dan 3 menghasilkan 3. Operasi NOT pada 567 menghasilkan -568. Hasil-hasil ini kemudian dicetak ke layar dengan pesan yang sesuai.menampilkan operasi bitwise NOT dan dilanjutkan dengan hasilnya..

1. **Tugas Praktikum ke-1: Program Menghitung Volume Bola**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){    float phi = 3.14;  float r = 14;  float v,vs;    cout<<"........................................\n\tProgam Menghitung Volume\n\tBola & Setengah Bola\n........................................"<<endl<<endl;    cout<<"Masukan Jari-jari Bola : "<<r<<endl<<endl;    v = phi \* r \* r \* r \* 4/3;    cout<<"Volume Bola = "<<v<<endl<<endl;    vs = v/2;    cout<<"Volume Setengah Bola = "<<vs<<endl;    return 0;  } |

1. Hasil Running



### Gambar 3.4 Hasil *Running* Program Volume Bola

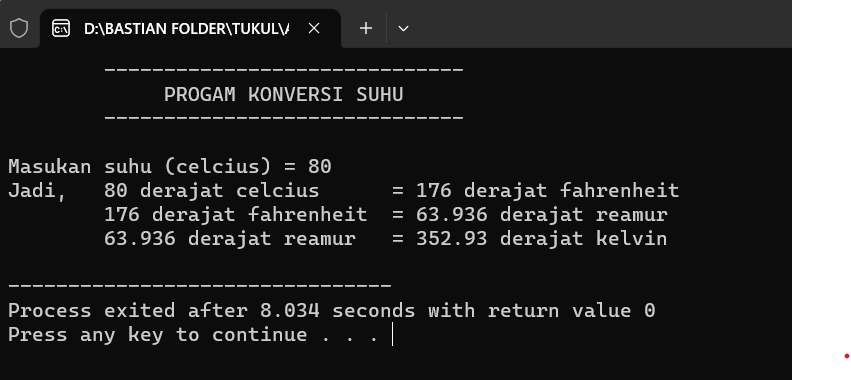
1. Analisa

Menggunakan tipe data float dan menggunakan variable v untuk mencari volume bola dengan hasil yang dikeluarkan adalah 11488.2 . Sedangkan variable yanag digunakan dalamn mencari volume setengah bola adalah vs dengan hasil yang dikeluarkan 5744.11.

1. **Tugas Praktikum ke-2:**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  double c;  cout<<"\t------------------------------\n\t PROGAM KONVERSI SUHU\n\t------------------------------"<<endl<<endl;  cout<<"Masukan suhu (celcius) = ";cin>>c;  double f = (c \* 9/5) + 32;  cout<<"Jadi,\t"<<c<<" derajat celcius\t= "<<f<<" derajat fahrenheit"<<endl;  double r = c \* 4/5 - 0.064;  cout<<"\t"<<f<<" derajat fahrenheit\t= "<<r<<" derajat reamur"<<endl;  double k = c + 273 - 0.07;  cout<<"\t"<<r<<" derajat reamur\t= "<<k<<" derajat kelvin"<<endl;  return 0;  } |

1. Hasil Running



### Gambar 3.5 Hasil *Running* Program Konversi Suhu

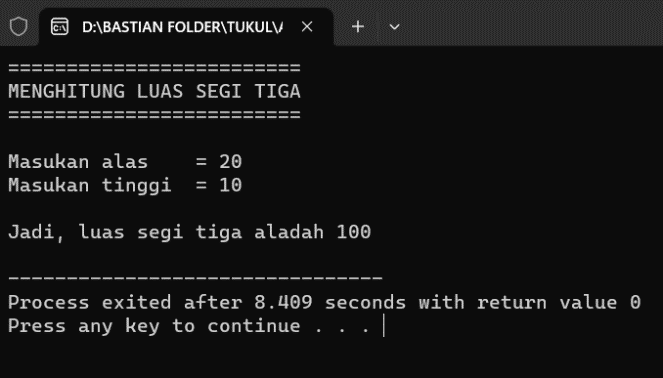
1. Analisa

Suhu yang sedang di konversikan adalah 80 derajat celcius. Menggunakan tipe data double, variabel yang digunakan untuk konversi ke Fahrenheit adalah f dengan hasil 176, untuk konversi ke reamur menggunakan variabel r dengan hasil 16.936, dan untuk konversi ke kelvin menggunakan variabel k dengan hasil 352.93.

1. **Tugas praktikum ke-3: Program Menghitung Luas Segitiga**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main (){  float l , a , t;  cout << "=========================\nMENGHITUNG LUAS SEGI TIGA\n========================="<<endl<<endl;  cout<<"Masukan alas\t= ";cin>>a;  cout<<"Masukan tinggi\t= ";cin>>t;  l = 0.5\*a\*t;  cout<<"\nJadi, luas segi tiga aladah "<<l<<endl;  } |

1. Hasil Running



### Gambar 3.6 Hasil *Running* Program Menghitung Luas Segitiga

1. Analisa

Menggunakan tipe data float untuk menampilkan luas segitiga mengggunakan variabel l, alas menggunakan variabel a, dan untuk tinggi menggunakan variabel t. keluaran dari variabel l adalah 0.5 \* a \* t.

1. **Kesimpulan**
2. Operator merupakan simbol yang biasa dilibatkan dalam program untuk melakukan suatu operasi atau manipulasi. Dengan operator, kita dapat melakukan operasi perhitungan, perbandingan, manipulasi bit, dan lain-lain,
3. Operator aritmatika adalah operator yang digunakan untuk melakukan sebuah opearasi penghitungan atau aritmatika. Tugasnya untuk memecahkan masalah matematika.
4. Terdapat dua jenis increment/decrement, yaitu pre-increment adalah melakukan penambahan nilai sebelum suatu variabel itu diproses, dan post-increment yaitu melakukan proses terlebih dahulu sebelum dilakukan penambahan nilai

**BAB IV  
Percabangan**

|  |  |  |
| --- | --- | --- |
| **Durasi Pertemuan** | **:** | 100 Menit |
| **Tujuan Praktikum** | **:** | 1. Mampu memahami konsep tentang Kondisi 2. Mampu memahami konsep IF dan Switch Case 3. Mampu membuat program dengan menambahkan konsep Kondisi. |
| **Alat / bahan** | **:** | 1. Seperangkat *computer*. 2. Perangkat lunak: Dev C++. 3. Modul Praktikum Algoritma dan Pemrogramman 2023. |

1. **Landasan Teori**
2. *IF Else*

Salah satu permasalahan yang pasti akan dijumpai dalam pembuatan program adalah suatu percabangan. Percabangan yang dimaksud di sini adalah suatu pemilihan statement yang akan dieksekusi di mana pemilihan tersebut didasarkan atas kondisi tertentu.

Dalam kehidupan sehari-hari terkadang kita dihadapkan pada kondisi dimana kita harus memilih suatu tindakan. Sebagai contoh ketika anda ingin menonton film di bioskop. Konsepnya seperti ini: Jika anda memiliki tiket, maka anda boleh masuk untuk menonton film.

Nah, disana ada sebuah kondisi yang harus terpenuhi yaitu memiliki tiket. Artinya jika anda tidak memiliki tiket (kondisi = FALSE) maka anda tidak diperbolehkan untuk menonton film di bioskop. Kurang lebih itulah contoh nyata konsep percabangan.

Intinya selama kondisi terpenuhi maka program akan dijalankan. Begitu juga dalam membuat program, kita akan sedikit banyak berurusan dengan pilihan. Disini kita akan mengenal dua fungsi percabangan yaitu if else dan switch case.

* + - 1. Struktur if Satu Kondisi

Struktur ini merupakan yang paling sederhana karena hanya melibatkan satu kondisi yang akan diperiksa.Bentuk umum dari struktur percabangan yang memiliki satu kondisi adalah sebagai berikut :

|  |
| --- |
| if (kondisi) {  pernyataan1;  pernyataan2;  // kode yang di jalankan jika pernyataan benar  ......  } |

Pernyataan IF diatas mempunyai pengertian, “ Jika kondisi bernilai benar, maka perintah/pernyataan akan dikerjakan dan jika tidak memenuhi syarat maka akan diabaikan”.

Jika ’pernyataan’ yang dijalankan hanya sebaris, maka tanda {} boleh ditiadakan. Seperti dibawah ini :

|  |
| --- |
| if (kondisi) pernyataan; |

* + - 1. Struktur if Dua Kondisi

Pada struktur ini terdapat sebuah statemen khusus yang berguna untuk mengatasi kejadian apabila kondisi yang didefinisikan tersebut tidak terpenuhi (bernilai salah).

Bentuk umum dari sturuktur if adalah sebagai berikut :

|  |
| --- |
| if (kondisi) {  pernyataan\_jika\_kondisi\_terpenuhi;  .....  }  Else {  pernyataan\_jika\_kondisi\_tidak\_terpenuhi;  .....  } |

* + - 1. Struktur if Tiga Kondisi atau Lebih

Percabangan ini merupakan perluasan dari struktur yang memiliki dua kondisi, yaitu dengan menyisipkan satu atau lebih kondisi ke dalamnya. Untuk struktur yang memiliki tiga kondisi atau lebih, bentuk umum dari struktur ini adalah adalah seperti yang tampak di bawah ini.

|  |
| --- |
| if (kondisi1) {  statemen\_jika\_kondisi\_terpenuhi;  } else if (kondisi2){  statemen\_jika\_kondisi2\_terpenuhi;  } else if (kondisi3){  statemen\_jika\_kondisi3\_terpenuhi;  }  .....  else {  statemen\_jika\_semua\_kondisi\_diatas\_tidak\_terpenuhi;  } |

1. Struktur Switch

Selain menggunakan statemen *if*, C++ juga menawarkan untuk dapat melakukan percabangan (pemilihan) dengan menggunakan statemen *switch*. Pernyataan ***switch*** adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan pernyataan, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksian. Pernyataan ***if****…****else if*** jamak dapat dibangun dengan pernyataan ***switch.***

Bentuk umum dari Switch :

|  |
| --- |
| switch (ekspresi){  case konstanta1:  pernyataan1;  break;  case konstanta2:  pernyataan2;  break;  case konstanta3:  pernyataan3;  break;  default:  pernyataan\_lain;  } |

Hal-hal yang perlu diperhatikan :

1. Dibelakang keyword case harus diikuti oleh sebuah konstanta, tidak boleh diikuti oleh ekspresi ataupun variabel.
2. Konstanta yang digunakan bertipe int atau char.
3. Jika bentuknya seperti diatas maka apabila ekspresisesuai dengan konstanta2 maka pernyataan2, pernyataan3 sampai dengan pernyataan lain dieksekusi. Untuk mencegah hal tersebut, gunakan *keyword* ***break****.* Jika *keyword* ***break*** digunakan maka setelah pernyataan2 dieksekusi program langsung keluar dari pernyataan ***switch****.* Selain digunakan dalam ***switch****,* keyword *break* banyak digunakan untuk keluar dari pernyataan yang berulang ( *looping* ).
4. Pernyataan lain (default) dieksekusi jika konstanta1 sampai konstanta N tidak ada yang memenuhi *ekspresi.*
5. Menggunakan Range di switch case statement

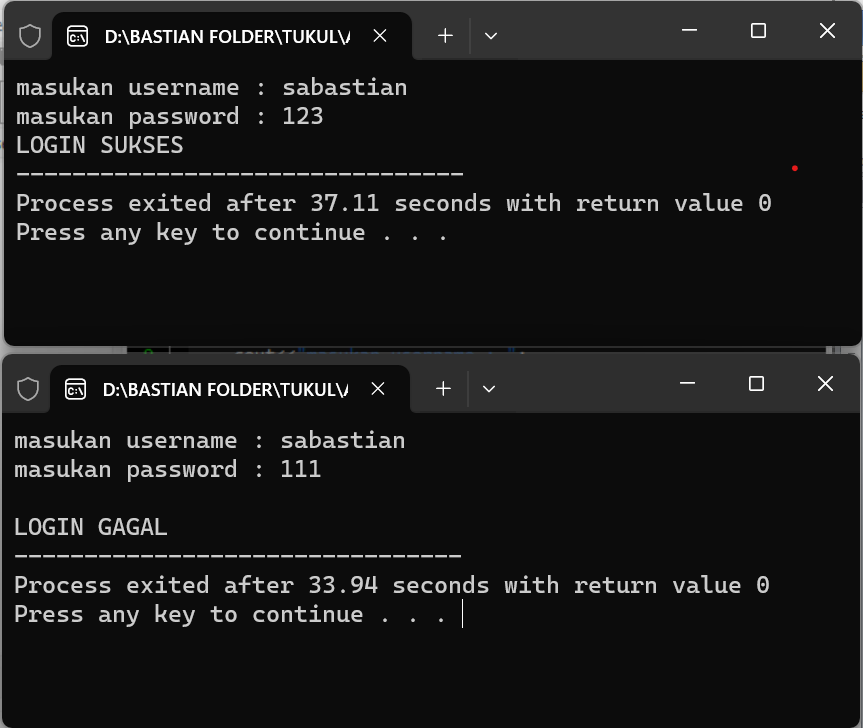
Kita juga bisa menggunakan range di switch case. Misal kita mengiginkan range 1 sampai 10, maka dapat ditulis dengan 1 ... 10, Sintaks menggunakan range di Case adalah seperti di bawah ini.

|  |
| --- |
| case low … high |

1. **Langkah – Langkah Praktikum**
2. Buka Aplikasi Dev C++.
3. Buka Modul Praktikum Algoritma dan Pemrogramman.
4. Menjalankan script dari modul.
5. **Latihan Praktikum ke-1: (If Else)**
6. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main (){  string u;  int p;  cout<<"masukan username : ";  cin>>u;  cout<<"masukan password : ";  cin>>p;  if ((u == "sabastian") && (p == 123)){  cout<<"LOGIN SUKSES";}  else {  cout<<"\nLOGIN GAGAL";}  } |

1. Hasil Running



### Gambar 4.1 Contoh *Running* Program Login

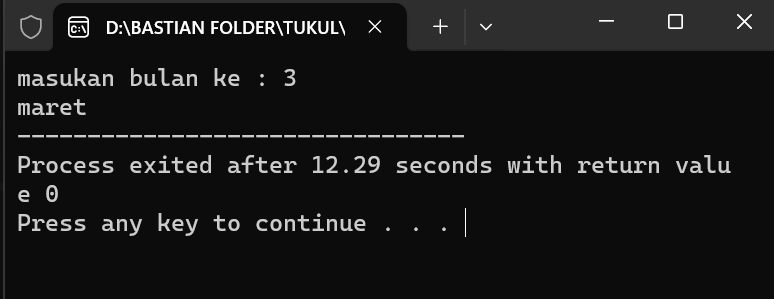
1. Analisa

Menggunakan tipe data string dan integer dengan variabel u untuk user dan p untuk password. Menggunakan 2 kondisi if-else dengan kondisi pertama menggunakan pembanding (atau sama dengan) dan menggunaak biswite && (atau AND), jika u atau username sama dengan sabastian dan p atau password sama engan 123 makan kondisi akan terpenuhu atau true dan jika tidak terpenuhui akan false.

1. **Latihan Praktikum ke-2: (If Else)**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int main(){  int n;  cout<<"masukan bulan ke : ";cin>>n;  if(n == 1){  cout<<"Januari";}  else if(n == 2){  cout<<"Febuary";}  else if(n == 3){  cout<<"maret";}  else if(n==4){  cout<<"april";}  else if(n==5){  cout<<"mei";}  else if(n==6){  cout<<"juni";}  else if(n==7){  cout<<"juli";}  else if(n==8){  cout<<"agustus";}  else if(n==9){  cout<<"september";}  else if(n==10){  cout<<"oktober";}  else if(n==11){  cout<<"november";}  else if(n==12){  cout<<"desember";}  return 0;  }    cout << "Bulan ke " << nomorBulan << " Adalah Bulan: " << namaBulan <<endl;  return 0;  } |

1. Hasil Running



### Gambar 4.2 Hasil *Running* Program Menentukan Bulan

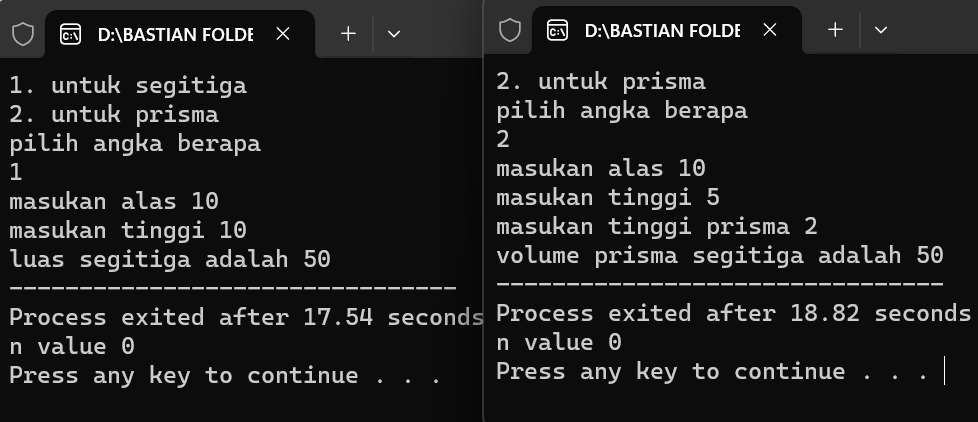
1. Analisa

Menggunakan tipe data integer dengan variabel n sebagai angka bulan. Menggunakan 13 kondisi dengan menggunakan pembanding == (atau sama dengan). jika n sama dengan 1 makan akan keluar bulan januari, jika n sama dengan 2 makan akan keluar bulan febuari, jika n sama dengan 3 makan akan keluar bulan maret, jika n sama dengan 4 makan akan keluar bulan april, jika n sama dengan 5 makan akan keluar bulan mei, jika n sama dengan 6 makan akan keluar bulan juni, jika n sama dengan 7 makan akan keluar bulan juli, jika n sama dengan 8 makan akan keluar bulan agustus, jika n sama dengan 9 makan akan keluar bulan september, jika n sama dengan 10 makan akan keluar bulan oktober, jika n sama dengan 11 makan akan keluar bulan november, jika n sama dengan 12 makan akan keluar bulan desember.

1. **Latihan Praktikum ke-3: (If Else)**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  char rumus;  cout<<"1. untuk segitiga\n2. untuk prisma"<<endl;  cout<<"pilih angka berapa\n";  cin>>rumus;  if (rumus=='1'){  int a,t,l;  cout<<"masukan alas ";cin>>a;  cout<<"masukan tinggi ";cin>>t;  l=1.0/2.0\*a\*t;  cout<<"luas segitiga adalah "<<l;}  else if (rumus=='2'){  int a,t,v,tp;  cout<<"masukan alas ";cin>>a;  cout<<"masukan tinggi ";cin>>t;  cout<<"masukan tinggi prisma ";cin>>tp;  v=((a\*t)/2)\*tp;  cout<<"volume prisma segitiga adalah "<<v;}  else {  cout<<"NO TIDAK ADA";}  } |

1. Hasil Running



### Gambar 4.3 Hasil *Running* Menghitung Segitiga

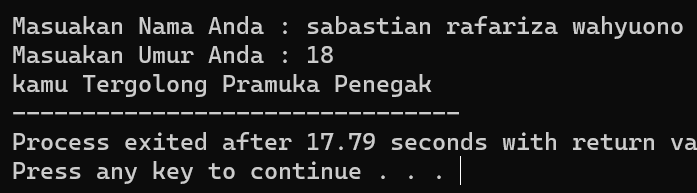
1. Analisa

Menggunakan tipe data char dan integer dengan variabel rumus, a, t, v, tp, l. Menggunakan 3 kondisi dengan menggunakan pembanding (atau sama dengan). Pada kondisi pertama jika memilih angka 1 maka kondisi true kemudian akan mengeluarkan inputan untuk variabel a dan t lalu akan di hitung di variabel l. Pada kondisi kedua akan true jika memilih angka 2 lalu akan mengeluarkan perintah untuk input variabel a, t, dan tp, kemudian akan dihitung menggunakan variabel v. Pada kondisi ke tiga yaitu else akan true jika kondisi 1 dan 2 tidak terpenuhu atau false.

1. **Latihan Praktikum ke-4: (Switch case)**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){    int umur;  string nama;    cout<<"Masuakan Nama Anda : ";getline(cin,nama);  cout<<"Masuakan Umur Anda : ";cin>>umur;    switch(umur){  case 1 ... 6 :  cout<<"Umur Kamu Belum Tergolong Tingakatan di Pramuka";  break;  case 7 ... 10:  cout<<"kamu Tergolong Pramuka Siaga";  break;  case 11 ... 15:  cout<<"kamu Tergolong Pramuka Penggalang";  break;  case 16 ... 20:  cout<<"kamu Tergolong Pramuka Penegak";  break;  case 21 ... 25:  cout<<"kamu Tergolong Pramuka Pendega";  break;  case 26 ... 100:  cout<<"kamu Tergolong Majelis Pembimbing";  break;  default:  cout<<"PILIHAN ANDA TIDAK ADA";  }  } |

1. Hasil Running



### Gambar 4.4 Contoh *Running* Latihan 4

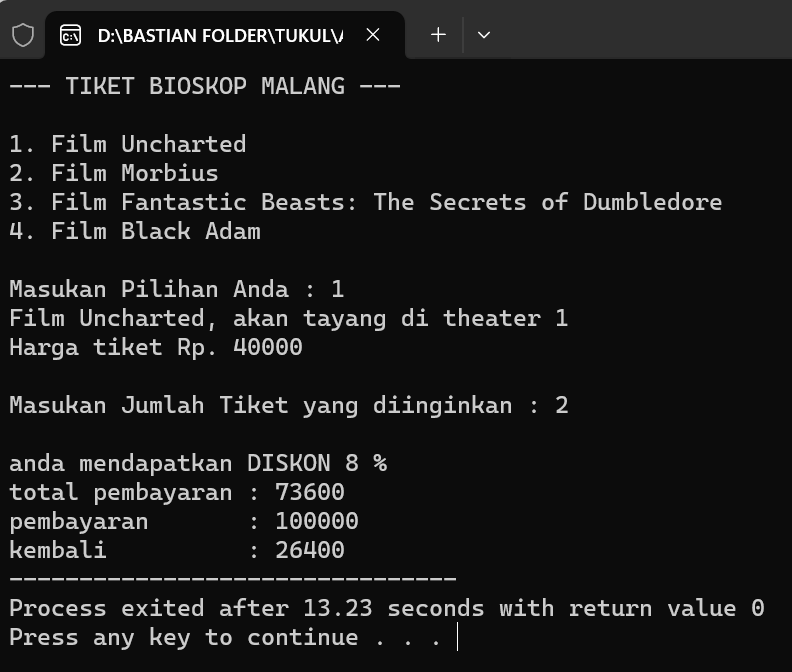
1. Analisa

Menggunakan tipe data integer dengan variabel umur dan tipe data string dengan variabel nama. Mengeluarkan mengeluarkan inputan untuk variabel nama dan inputan untuk variabel umur. Menggunakan switch variabel nama dengan case 1 – 6 yang akan mengeluarkan “Umur Kamu Belum Tergolong Tingakatan di Pramuka”, case 7 – 10 mengeluarkan "kamu Tergolong Pramuka Siaga", case 11 – 15 mengeluarkan "kamu Tergolong Pramuka Penggalang", case 16 – 20 mengeluarkan "kamu Tergolong Pramuka Penegak", case 21 – 25 mengeluarkan "kamu Tergolong Pramuka Pendega", case 26 – 100 mengeluarkan kamu "Tergolong Majelis Pembimbing". Jika ttidak ada yang benar makan mengeluarkan "PILIHAN ANDA TIDAK ADA".

1. **Latihan Praktikum ke-5: (Switch case)**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){    int nb,jumlah,total,diskon,bayar;    cout<<"--- TIKET BIOSKOP MALANG ---"<<endl<<endl;  cout<<"1. Film Uncharted\n2. Film Morbius\n3. Film Fantastic Beasts: The Secrets of Dumbledore\n4. Film Black Adam"<<endl<<endl;  cout<<"Masukan Pilihan Anda : ";cin>>nb;  switch(nb){  case 1 :  cout<<"Film Uncharted, akan tayang di theater 1\nHarga tiket Rp. 40000 \n"<<endl;  cout<<"Masukan Jumlah Tiket yang diinginkan : ";cin>>jumlah;  total = 40000 \* jumlah;  switch (total){  case 0 ... 40000 :  cout<<"anda tidak mendapat diskon";  break;  default :  cout<<"\nanda mendapatkan DISKON 8 %"<<endl;  diskon = total-(0.08\*total);  cout<<"total pembayaran : "<<diskon<<endl;  cout<<"pembayaran\t : ";cin>>bayar;  if (bayar >= diskon){  cout << "kembali\t\t : "<<bayar - diskon;  }  else {  cout<<"\npembayaran tidak sesuai";  }  }  break;    case 2 :    cout<<"Film Morbius, akan tayang di theater 2\nHarga tiket Rp. 35000 ";  cout<<"Masukan Jumlah Tiket yang diinginkan : ";cin>>jumlah;  total = 35000 \* jumlah;  switch (total){  case 0 ... 40000 :  cout<<"anda tidak mendapat diskon";  break;  default :  cout<<"\nanda mendapatkan DISKON 8 %"<<endl;  diskon = total-(0.08\*total);  cout<<"total pembayaran : "<<diskon<<endl;  cout<<"pembayaran\t : ";cin>>bayar;  if (bayar >= diskon){  cout << "kembali\t\t : "<<bayar - diskon;  }  else {  cout<<"\npembayaran tidak sesuai";  }  }  break;    case 3 :  cout<<"Film Fantastic Beasts: The Secrets of Dumbledore, akan tayang di theater 3\nHarga tiket Rp. 30000 ";  cout<<"Masukan Jumlah Tiket yang diinginkan : ";cin>>jumlah;  total = 30000 \* jumlah;  switch (total){  case 0 ... 40000 :  cout<<"anda tidak mendapat diskon";  break;  default :  cout<<"\nanda mendapatkan DISKON 8 %"<<endl;  diskon = total-(0.08\*total);  cout<<"total pembayaran : "<<diskon<<endl;  cout<<"pembayaran\t : ";cin>>bayar;  if (bayar >= diskon){  cout << "kembali\t\t : "<<bayar - diskon;  }  else {  cout<<"\npembayaran tidak sesuai";  }  }  break;    case 4 :  cout<<"Film Black Adam, akan tayang di theater 4\nHarga tiket Rp. 50000 ";  cout<<"Masukan Jumlah Tiket yang diinginkan : ";cin>>jumlah;  total = 50000 \* jumlah;  switch (total){  case 0 ... 40000 :  cout<<"anda tidak mendapat diskon";  break;  default :  cout<<"\nanda mendapatkan DISKON 8 %"<<endl;  diskon = total-(0.08\*total);  cout<<"total pembayaran : "<<diskon<<endl;  cout<<"pembayaran\t : ";cin>>bayar;  if (bayar >= diskon){  cout << "kembali\t\t : "<<bayar - diskon;  }  else {  cout<<"\npembayaran tidak sesuai";  }  }  break;  }  } |

1. Hasil Running



### Gambar 4.5 Hasil *Running* Program Tiket Bioskop

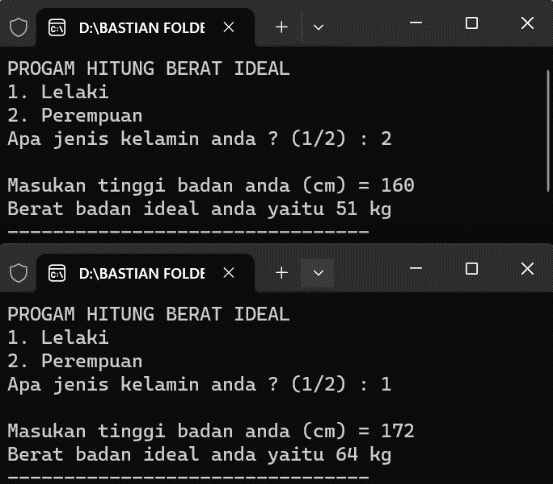
1. Analisa

Menggunakan tipe data integer dengan variabel nb, jumlah, total, diskon, bayar. Menggunakan switch pada variabel nb dengan case 1 mengeluarkan perintah mamasukan variabel jumlah. Kemudian variabel total adalah perkalian variabel jumlah dengan harga tiket, selanjutnya menggunakan switch variabel total jika pembayaran lebih dari 40000 maka akan menuju default maka mendapatkan diskon diskon jika tidak maka tidak lebih dari 40000 maka tidak akan mendapat diskon. kemudian akan memunculkan total pembayaran dan inputan untuk memasukan pembayaran. Jika pembayaran kurang dari total haraga maka maka akan memunculkan pembayaran tidak sesuai den jika pembayaran lebih dari total harga makan akan mengeluarkan kembalian yaitu pembayaran dikurangi total pembayaran. Untuk total pembayaran saya menggunakan if else. Begitu juga dengan case 2, 3, dan 4.

1. **Tugas Praktikum ke-1: (If Else)**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int main(){  int lp,tinggi;  cout << "PROGAM HITUNG BERAT IDEAL\n1. Lelaki\n2. Perempuan\nApa jenis kelamin anda ? (1/2) : ";cin>>lp;  if (lp==1){  cout << "\nMasukan tinggi badan anda (cm) = ";cin>>tinggi;  int berat = (tinggi-100)-((tinggi-100)\*0.10);  cout << "Berat badan ideal anda yaitu "<<berat<<" kg";}  else if (lp==2){  cout << "\nMasukan tinggi badan anda (cm) = ";cin>>tinggi;  int berat = (tinggi-100)-((tinggi-100)\*0.15);  cout << "Berat badan ideal anda yaitu "<<berat<<" kg";}  else{  cout << "Error salah memilih jenis kelamin";}  return 0;  } |

1. Hasil *Running*



### Gambar 4.6 Hasil *Running* Program Menghitung BB

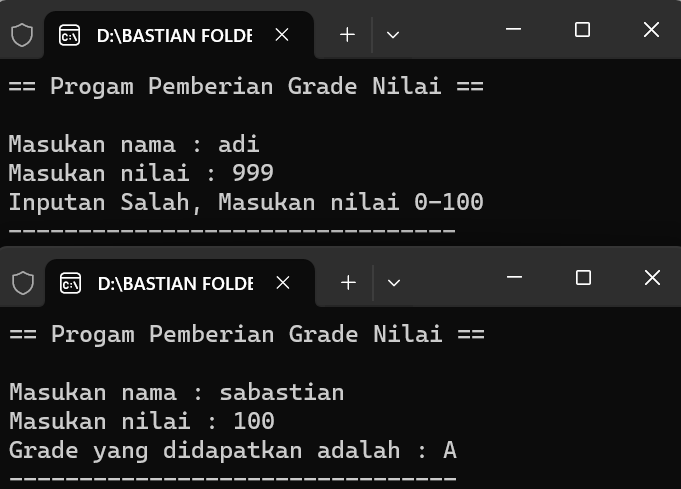
1. Analisa

Menggunakan tipe data integer dengan variabel lp, tinggi, dan berat. Menggunakan 3 kondisi dengan menggunakan pembanding == (atau sama dengan). Padan kondisi pertama jika nilai lp == (sama dengan) 1 maka akan meminta untuk menginputkan tinggi, kemudian akan menampilkan berat dengan rumus berat adalah (tinggi-100)-((tinggi-100)\*0.15). Padan kondisi kedua jika nilai lp == (sama dengan) 2 maka akan meminta untuk menginputkan tinggi, kemudian akan menampilkan berat dengan rumus berat adalah (tinggi-100)-((tinggi-100)\*0.15). jika kedua kondisi tidak terpenuhi makan akan menampilkan kondisi else dengan memunculkan tulisan eror.

1. **Tugas Praktikum ke-2: (If Else)**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int main(){  string nama;  int nilai;  cout << "== Progam Pemberian Grade Nilai =="<<endl<<endl;  cout << "Masukan nama : ";  getline (cin,nama);  cout << "Masukan nilai : ";cin>>nilai;    if (nilai<0||nilai>100){  cout<<"Inputan Salah, Masukan nilai 0-100";  }  else if (nilai >= 80){  cout<<"Grade yang didapatkan adalah : A ";  }  else if (nilai >= 60){  cout<<"Grade yang didapatkan adalah : B ";  }  else if (nilai >= 40){  cout<<"Grade yang didapatkan adalah : C ";  }  else if (nilai >= 20){  cout<<"Grade yang didapatkan adalah : D ";  }  else if (nilai >= 10){  cout<<"Grade yang didapatkan adalah : E ";  }  else if (nilai >= 0){  cout<<"Grade yang didapatkan adalah : F ";  }  return 0;  } |

1. Hasil *Running*



### Gambar 4.7 Hasil *Running* Program Grading Nilai

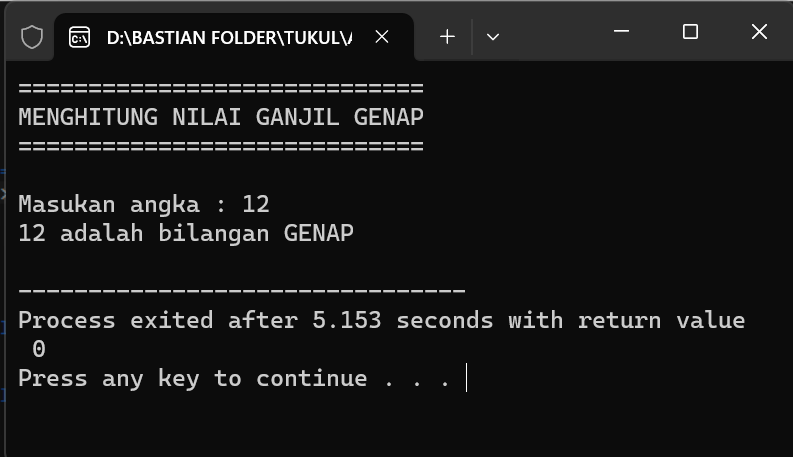
1. Analisa

Menggunakan tipe data string untuk variabel nilai dan integer untuk variabel nilai. Menggunakan 7 kondisi dengan menggunakan pembanding > , <, >= dan menggunakan biswite || OR. Kondisi pertama jiks nilsi kursng dari 0 atau lebuh dari seratus makan akan menampilkan “input salah, masukan nilai 0-100”.selanjutnya jika nilai lebih besar samadengan 80 maka akan mengeluarkan A, jika nilai lebih besar samadengan 60 maka akan mengeluarkan B, jika nilai lebih besar samadengan 40 maka akan mengeluarkan C, jika nilai lebih besar samadengan 20 maka akan mengeluarkan D, jika nilai lebih besar samadengan 10 maka akan mengeluarkan E, jika nilai lebih besar samadengan 0 maka akan mengeluarkan F.

1. **Tugas praktikum ke-3: (If Else)**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){    int angka;    cout<<"============================="<<endl<<"MENGHITUNG NILAI GANJIL GENAP"<<endl<<"============================="<<endl;  cout<<"\nMasukan angka : ";cin>>angka;    int h = angka % 2;      if (h==0){  cout <<angka<< " adalah bilangan GENAP\n";  }  else if (h==1){  cout <<angka<< " adalah bilangan GANJIL\n";  }    return 0;  } |

1. Hasil Running



### Gambar 4.8 Hasil *Running* Program Menentukan Ganjil Genap

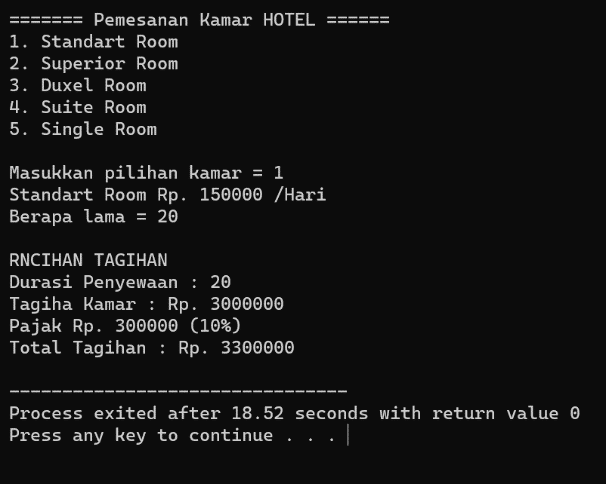
1. Analisa

Menggunakan tipe data integer dengan variabel angka dan h. menggunakan 2 kondisi dengan simbol pembanding ==. Menggunakan 2 kondisi . pertama menginputkan variabel angka kemudian akan mencari variabel h dengan mencari sisah bagi variabel angka dengan pembagian 2. Pada kondisi 1 akan true jika sisah bagi atau h == 0 dan akan menampilkan “bilangan tersebut GENAP”. Pada kondis 2 akan true jika sisah bagi atau h == 1 dan akan menampilkan “bilangan anda GANJIL”.

1. **Tugas Praktikum ke-4: (Switch case)**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  int kamar,lama,tagihan,total,pajak;    cout <<"======= Pemesanan Kamar HOTEL ======"<<endl;  cout <<"1. Standart Room"<<endl;  cout <<"2. Superior Room"<<endl;  cout <<"3. Duxel Room"<<endl;  cout <<"4. Suite Room"<<endl;  cout <<"5. Single Room"<<endl<<endl;  cout<<"Masukkan pilihan kamar = ";cin>>kamar;    switch(kamar){  case 1 :  cout<<"Standart Room Rp. 150000 /Hari"<<endl;  cout<<"Berapa lama = ";cin>>lama;    cout<<"\nRNCIHAN TAGIHAN"<<endl;  cout<<"Durasi Penyewaan : "<<lama<<endl;  tagihan = lama\*150000;  cout<<"Tagiha Kamar : Rp. "<<tagihan<<endl;  pajak = 0.1\*tagihan;  cout<<"Pajak Rp. "<<pajak<<" (10%)"<<endl;    cout<<"Total Tagihan : Rp. "<<pajak+tagihan<<endl;    break;  case 2 :  cout<<"Superior Room Rp. 490000 /Hari"<<endl;  cout<<"Berapa lama = ";cin>>lama;    cout<<"\nRNCIHAN TAGIHAN"<<endl;  cout<<"Durasi Penyewaan : "<<lama<<endl;  tagihan = lama\*490000;  cout<<"Tagiha Kamar : Rp. "<<tagihan<<endl;  pajak = 0.1\*tagihan;  cout<<"Pajak Rp. "<<pajak<<" (10%)"<<endl;    cout<<"Total Tagihan : Rp. "<<pajak+tagihan<<endl;    break;  case 3 :  cout<<"Deluxe Room Rp. 750000 /Hari"<<endl;  cout<<"Berapa lama = ";cin>>lama;    cout<<"\nRNCIHAN TAGIHAN"<<endl;  cout<<"Durasi Penyewaan : "<<lama<<endl;  tagihan = lama\*750000;  cout<<"Tagiha Kamar : Rp. "<<tagihan<<endl;  pajak = 0.1\*tagihan;  cout<<"Pajak Rp. "<<pajak<<" (10%)"<<endl;    cout<<"Total Tagihan : Rp. "<<pajak+tagihan<<endl;    break;  case 4:  cout<<"Suite Room Rp. 1550000 /Hari"<<endl;  cout<<"Berapa lama = ";cin>>lama;    cout<<"\nRNCIHAN TAGIHAN"<<endl;  cout<<"Durasi Penyewaan : "<<lama<<endl;  tagihan = lama\*1550000;  cout<<"Tagiha Kamar : Rp. "<<tagihan<<endl;  pajak = 0.1\*tagihan;  cout<<"Pajak Rp. "<<pajak<<" (10%)"<<endl;    cout<<"Total Tagihan : Rp. "<<pajak+tagihan<<endl;    break;  case 5 :  cout<<"Single Room Rp. 700000 /Hari"<<endl;  cout<<"Berapa lama = ";cin>>lama;    cout<<"\nRNCIHAN TAGIHAN"<<endl;  cout<<"Durasi Penyewaan : "<<lama<<endl;  tagihan = lama\*700000;  cout<<"Tagiha Kamar : Rp. "<<tagihan<<endl;  pajak = 0.1\*tagihan;  cout<<"Pajak Rp. "<<pajak<<" (10%)"<<endl;    cout<<"Total Tagihan : Rp. "<<pajak+tagihan<<endl;    break;  default :  cout << "Pilihan Anda Tidak Ada";  break;  }  return 0;  } |

1. Hasil Running



### Gambar 4.9 Hasil *Running* Program Pemesanan Kamar

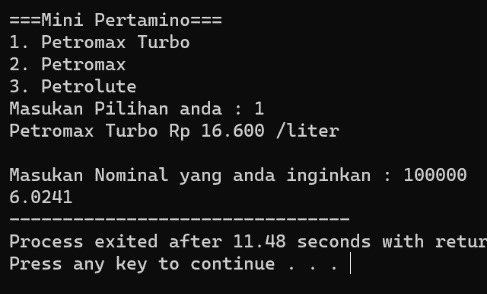
1. Analisa

Pemrogaman di atas digunakan untuk memesan kamar hotel. Menggunakan tipe data integer dengan variabel kamar,lama,tagihan,total,pajak. Pertama akan memunculkan inputan untuk variabel kamar. Kemudian menggunakan switch kamar pada case 1, 2, 3, 4 akan menggeluarkan harga kamar masing masing dan akann mengeluarkan inputan untuk variabel lama. Kemudian akan memunculkan rincihan tagihan, durasi penyewaan yaitu variabel lama, variabel tagihan yaitu variabel tagihan dengan mengalikan variabel lama dengan harga kamar, kemudian mengeluarkan variabel pajak yaitu variabael tagihan dikali 0,1 atau 10%.

1. **Tugas Praktikum ke-5: (Switch case)**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int main(){  int pilihan;  float total,nominal;    cout<<"===Mini Pertamino==="<<endl;  cout<<"1. Petromax Turbo"<<endl;  cout<<"2. Petromax"<<endl;  cout<<"3. Petrolute"<<endl;  cout<<"Masukan Pilihan anda : ";cin>>pilihan;    switch(pilihan){  case 1:  cout<<"Petromax Turbo Rp 16.600 /liter"<<endl<<endl;  cout<<"Masukan Nominal yang anda inginkan : ";cin>>nominal;  total = nominal/16600,00;  cout<<total;  break;  case 2:  cout<<"Turbo Rp 14.000 /liter";  cout<<"Masukan Nominal yang anda inginkan : ";cin>>nominal;  total = nominal/14000,00;  cout<<total;  break;  break;  case 3:  cout<<"Petrolite Rp 10.000 /liter";  cout<<"Masukan Nominal yang anda inginkan : ";cin>>nominal;  total = nominal/10000,00;  cout<<total;  break;  break;  }  } |

1. Hasil Running



### Gambar 4.10 Contoh *Running* Progam Pembelian Bahan Bakar

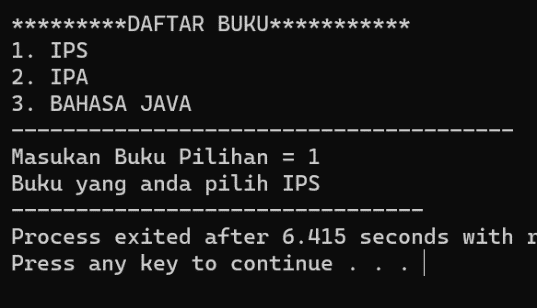
1. Analisa

Menggunakan tipe data integer dengan variabel pilihan dan float dengan variabel total dan nominal. Menggunakan switch variabel pilihan, case 1, 2, dan 3 akan memunculkan inputan untuk variabel nominal. Kemudian vaariabbel total aka memproses nominal di bagi harga bahan bakar yng sudah di tentukan.

1. **Tugas praktikum ke-6 (Switch case)**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int main (){  int buku;  cout<<"\*\*\*\*\*\*\*\*\*DAFTAR BUKU\*\*\*\*\*\*\*\*\*\*\*"<<endl;  cout<<"1. IPS"<<endl;  cout<<"2. IPA"<<endl;  cout<<"3. BAHASA JAVA"<<endl;  cout<<"---------------------------------------"<<endl;  cout<<"Masukan Buku Pilihan = ";  cin>>buku;  switch(buku) {  case 1 : cout<<"Buku yang anda pilih IPS"; break;  case 2 : cout<<"Buku yang anda pilih IPA"; break;  case 3 : cout<<"Buku yang anda pilih BAHASA JAVA"; break;  default : cout<<"Buku tidak tersedia";  }  } |

1. Hasil Running



### Gambar 4.11 Hasil *Running* Program Memilih Buku

1. Analisa

Menggunakan tipe data integer dengan variabel buku. Kemudian akan memunculkan pilihan untuk daftar buku dan masukan pilihan dengan variabel buku. Selanjutnya akan memunculkan switch buku dengan case 1, 2, 3. Yang masing- masing akan memunculkan outputan yang berbeda.

1. **Kesimpulan**
2. If else adalah suatu pemilihan statement yang akan dieksekusi di mana pemilihan tersebut didasarkan atas kondisi tertentu.
3. Struktur if satu kondisi merupakan yang paling sederhana karena hanya melibatkan satu kondisi yang akan diperiksa.
4. Struktur If dua kondisi terdapat sebuah statemen khusus yang berguna untuk mengatasi kejadian apabila kondisi yang didefinisikan tersebut tidak terpenuhi (bernilai salah).
5. Struktur if tiga kondisi atau lebih merupakan perluasan dari struktur yang memiliki dua kondisi, yaitu dengan menyisipkan satu atau lebih kondisi ke dalamnya. Untuk struktur yang memiliki tiga kondisi atau lebih.
6. Pernyataan *switch* adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan pernyataan, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksian.

**BAB V  
Perulangan**

|  |  |  |
| --- | --- | --- |
| **Durasi Pertemuan** | **:** | 120 Menit |
| **Tujuan Praktikum** | **:** | 1. Mampu memahami konsep perulangan 2. Mampu melakukan perulangan dengan bantuan “For” 3. Mampu melakukan perulangan dengan bantuan “While” 4. Mampu melakukan perulangan dengan bantuan “Do-While” 5. Mampu membuat program berdasarkan perulangan. |
| **Alat / bahan** | **:** | 1. Seperangkat *computer*. 2. Perangkat lunak: Dev C++. 3. Modul Praktikum Algoritma dan Pemrogramman 2023. |

1. **Landasan Teori**

Perulangan data atau yang biasa disebut dengan “*looping*” adalah proses yang dilakukan secara berulang-ulang sampai batas yang ditentukan. Perulangan digunakan untuk mengulang sekumpulan perintah sesuai dengan kondisi yang diberikan. Dengan *loop*, suatu proses yang berulang dapat diimplementasikan menggunakan *statement-statement* yang pendek, tanpa harus menuliskan statement berulang-ulang.

Dalam bahasa C++ terdapat empat buah jenis struktur pengulangan, yaitu :

1. Struktur For
2. For Satu Variabel

Merupakan salah satu bentuk fungsi *looping*, digunakan untuk melaksanakan pernyataan berulang kali terhadap sejumlah nilai yang telah ditetapkan. *Looping* yang dilakukan oleh *for* telah diketahui batas awal, batas kondisi dan perubahannya. Tipe data dari variabel yang akan digunakan sebagai indeks haruslah tipe data yang mempunyai urutan yang teratur, misalnya tipe data int (0,1,2,…) atau char(‘a’,’b’,’c’,...).

Bentuk umum:

|  |
| --- |
| for (inisialisasi ; kondisi ; perubahan) {  Statement;  } |

1. For Dengan Banyak Variabel

Struktur for di dalam C++ dapat juga melibatkan lebih dari satu variabel namun yang jelas satu diantaranya akan digunakan sebagai indeks pengulangan. Untuk memahami konsepnya, berikut ini disajikan sebuah contoh program dimana di dalamnya terdapat strukur for yang melibatkan dua buah variabel, yaitu A, B dan C.

|  |
| --- |
| for (inisialisasi1, inisialisasi2 ; kondisi1, kondisi2 ; perubahan1, perubahan2) {  Statement;  } |

1. Struktur While

Struktur perulangan *WHILE* adalah pengulangan yang melakukan pemeriksaan kondisi di awal blok struktur. Pengulangan hanya akan dilakukan jika kondisi yang didefinisikan di dalamnya terpenuhi (bernilai benar). Hal ini berarti jika kondisi yang didefinisikan tidak terpenuhi (bernilai salah) maka statemen-statemen yang terdapat dalam blok pengulangan pun tidak akan pernah dieksekusi oleh program.

Cek Kondisi?

while (true)

Blok kode

yang diulang

Bentuk umumnya :

|  |
| --- |
| while (kondisi) {  statemen\_statemen\_yang\_akan\_diulang;  perubahan;  } |

Algoritma Program :

1. Sebuah **while** loop mengecek **kondisi**
2. Selama **kondisi** yang dicek bernilai **true**, kode di dalam **while** loop di jalankan
3. **Kondisi** di cek lagi
4. Proses ini berlanjut hingga saat **kondisi** bernilai **false**
5. Saat **kondisi** yang di cek bernilai **false**, perulangan berakhir
6. Struktur Do-While

Berbeda dengan struktur while yang melakukan pemeriksaan kondisi di awal blok perulangan, pada struktur do-while kondisi justru ditempatkan di bagian akhir. Hal ini tentu menyebabkan struktur pengulangan ini minimal akan **melakukan satu kali** proses walaupun kondisi yang didefinisikan tidak terpenuhi (bernilai salah).

Blok kode

yang diulang

Cek Kondisi?

while (true)

Bentuk umum :

|  |
| --- |
| do {  statemen\_yang\_akan\_diulang;  perubahan;  } while (kondisi); |

Algoritma Program :

1. Kode didalam perulangan di jalankan
2. kemudian **kondisi** di cek
3. Selama **kondisi** yang dicek bernilai **true**, kode di dalam **do** di jalankan
4. **Kondisi** di cek lagi
5. Proses ini berlanjut hingga saat **kondisi** bernilai **false**
6. Saat **kondisi** yang di cek bernilai **false**, perulangan berakhir
7. Struktur Nested Loop

Pernyataan nested for adalah suatu perulangan for di dalam perulangan for yang lain. Di dalam penggunaan nested for, perulangan yang di dalam terlebih dahulu dihitung hingga selesai, kemudian perulangan yang di luar diselesaikan.

Biasanya, nested loops digunakan untuk membuat aplikasi matematika yang menggunakan baris dan kolom. Loop luar, biasanya digunakan untuk mendefinisikan kolom. Sedangkan loop dalam, digunakan untuk mendefinisikan baris. Bentuk umum :

1. Struktur Nested For

|  |
| --- |
| for (inisialisasi1 ; kondisi1 ; perubahan1) {  for (inisialisasi2 ; kondisi2 ; perubahan2) {  Statement;  }  } |

1. Struktur Nested While

|  |
| --- |
| Inisialisasi1;  while(kondisi){  inisialisasi2;  while(kondisi){  statemen;  perubahan2  }  Perubahan1;  } |

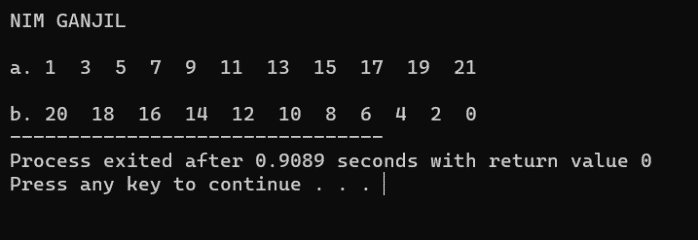
1. Struktur Nested Do-While

|  |
| --- |
| Inisialisasi1;  do{  inisialisasi2;  do{  Statement;  Perubahan2;  }while(kondisi2);  Perubahan1;  }while(Kondisi1); |

1. **Langkah – Langkah Praktikum**
2. Buka Aplikasi Dev C++.
3. Buka Modul Praktikum Algoritma dan Pemrogramman.
4. Menjalankan script dari modul
5. **Latihan Praktikum ke-1: (For)**
6. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int main(){  int p,q;  cout<<"NIM GANJIL"<<endl<<endl;  cout<<"a. ";  for (p = 1 ; p <= 11 ; p++) {  cout<<p\*2-1<<" ";  }    cout<<"\n\nb. ";  for (q = 10 ; q >= 0 ; q--) {  cout<<q\*2<<" ";  }  return 0;  } |

1. Hasil Running



### Gambar 5.1 Hasil *Running* Program NIM Ganjil

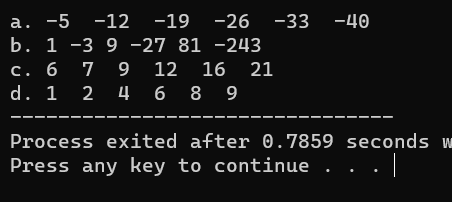
1. Analisa

Pada program diatas, terdapat variabel p dan q yang bertipe data *integer*. Kemudian variabel p akan di ulang dengan pernyataan for sesuai dengan kondisinya, yaitu p = 1 sampai p <=11, p++ diikuti dengan output p\*2-1 yang akan di ulang sampai kondisi tersebut tidak terpenuhi lagi. Untuk variabel q di juga ulang dengan pernyataan for dengan kondisi, yaitu q = 10 sampai q >=0, q-- diikuti dengan output q\*2 yang akan di ulang sampai kondisi tersebut tidak terpenuhi lagi.

1. **Latihan Praktikum ke-2: (For)**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int main(){  int a, b, bb, c, cc, d;    cout<<"a. ";  for (a = -5 ; a >= -40 ;a-=7 ) {  cout<<a<<" ";  }    cout<<"\nb. ";  for (b=1,bb=-3;b<=81,bb>=-243;b\*=9,bb\*=9) {  cout<<b<<" "<<bb<<" ";  }    cout<<"\nc. ";  for(c=0,cc=6;c<=5,cc<=21;c++,cc=c+cc){  cout<<cc<<" ";  }    cout<<"\nd. ";  for (d = 1; d < 10; d++) {  if (d == 3 || d == 5 || d == 7) {  continue;  }  cout << d << " ";  }  } |

1. Hasil Running



### Gambar 5.2 Hasil *Running* Program Angka Ganjil

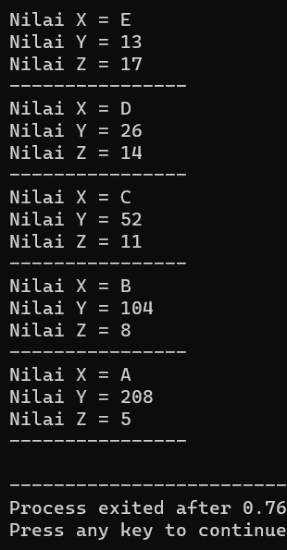
1. Analisa

Program pertama menggunakan variabel i dimulai dari -5 dan berkurang sebesar 7 setiap iterasi, sampai mencapai -40. *Loop* ini mencetak nilai i setiap iterasi.. *Loop* kedua dimulai dari 1 dan dikalikan dengan -3 setiap iterasi, hingga mencapai 81. *Loop* ini mencetak nilai i setiap iterasi.. *Loop* ketiga menggunakan dua variabel: i dan a. Loop ini mencetak nilai a setiap iterasi (dimulai dari a = 6) dan kemudian nilai a diubah dengan menambahkan nilai i ditambah 1 setiap iterasi. Loop terakhir mencetak nilai i dari 1 hingga 9. Jika nilai i sama dengan 1 atau 9, nilai i akan dicetak. Jika tidak, tetapi nilai i adalah bilangan genap, nilai i juga akan dicetak.

1. **Latihan Praktikum ke-3 (For)**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int main(){    int x,y,z;    for(x=1, y=13, z=17;x<=5, y<=208, z>=5;x++, y\*=2, z-=3){  switch(x){  case 1 :  cout << "Nilai X = E"<<endl;  break;  case 2 :  cout << "Nilai X = D"<<endl;  break;  case 3 :  cout << "Nilai X = C"<<endl;  break;  case 4 :  cout << "Nilai X = B"<<endl;  break;  case 5 :  cout << "Nilai X = A"<<endl;  break;  }    cout << "Nilai Y = "<<y<<endl;  cout << "Nilai Z = "<<z<<endl;  cout << "----------------"<<endl;  }  } |

1. Hasil Running



### Gambar 5.3 Hasil *Running* Perulangan For Angka dan Huruf

1. Analisa

Program di atas menggunakan perulangan *for* untuk mengiterasi nilai dari variabel a, y, dan z. Pertama, program menginisialisasi variabel a dengan karakter 'E', serta variabel y dan z dengan nilai 13 dan 17. Kemudian, perulangan dilakukan selama nilai karakter a masih lebih besar atau sama dengan 'A'. Dalam setiap iterasi, program mencetak nilai a, y, dan z ke layar, kemudian mengurangkan nilai a sebesar satu (a--), mengalikan nilai y dengan 2 (y\*=2), dan mengurangkan nilai z sebesar 3 (z-=3). *Output* dari program menampilkan perubahan nilai-nilai tersebut pada setiap iterasi, dan setelah nilai a kurang dari 'A', program mengembalikan nilai 0, menandakan bahwa program telah selesai.

1. **Latihan Praktikum ke-4 (While)**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  main()  {  int i, a=0, b=1, c, n;  cout << "masukan angka : "; cin>>n;  cout<<a<<" "<<b<<" ";  while(i<=n-3){  c=a+b;  a=b;  b=c;  cout<<c<<" ";  i++;  }  } |

1. Hasil *Running*



### Gambar 5.4 Hasil *Running* Program Fibonaci

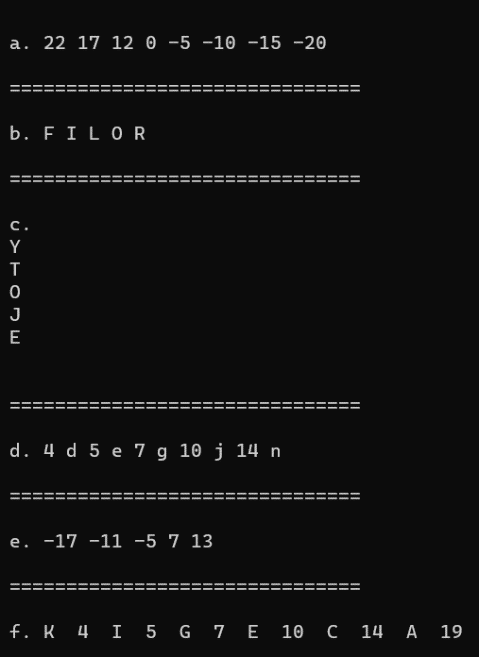
1. Analisa

Progam diatas merupakan progam untuk mencetak angka *fibionic* yang menggunakan tipe data *integer* dengan variabel i, a=0, b=1, c, n. kemudian mengeluarkan perintah input variabel n dan mengeluarkan variabel a dan b. setelah itu menggunakan perulanagan *while* dengan progam i<=n-3 yang didalamnya menggunakan progam c = a + b; a = b; b = c; cout << c << " "; dan i++;.

1. **Latihan Praktikum ke-5 (Do While):**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  cout << "\na. ";  int a = 22;  do{  cout << a << " ";  if(a == 12){  a -= 12;  continue;  }  a -= 5;  }  while(a >= -20);  cout <<"\n\n===============================\n\n";  cout << "b. ";  char hurufb = 'F';  do{  cout << hurufb << " ";  hurufb += 3;  }  while(hurufb <= 'R');  cout <<"\n\n===============================\n\n";  cout << "c. \n";  char hurufc = 'Y';  do {  cout<<hurufc<<endl;  hurufc -= 5;  }  while(hurufc >= 'A');  cout <<"\n\n===============================\n\n";  cout << "d. ";  int loop = 0, angka = 4;  char hurufd = 'd';  do {  cout << angka << " ";  cout << hurufd << " ";  loop++;  angka += loop;  hurufd += loop;  }  while(loop < 5);  cout <<"\n\n===============================\n\n";  cout << "e. ";  int keluaran = -17, looper = 0;  do {  cout << keluaran << " ";  if(looper == 2)  {  keluaran += 12;  looper++;  continue;  }  keluaran += 6;  looper++;  }  while(keluaran <= 13);  cout <<"\n\n===============================\n\n";  cout << "f. ";  int well = 4, loopnya = 0;  char done = 'K';  do {  cout << done << " ";  cout << well << " ";  loopnya++;  well += loopnya;  done -= 2;  }  while(loopnya<=5);  cout <<"\n";  } |

1. Hasil *Running*



### Gambar 5.5 Hasil *Running* Program Perulangan Angka dan Huruf

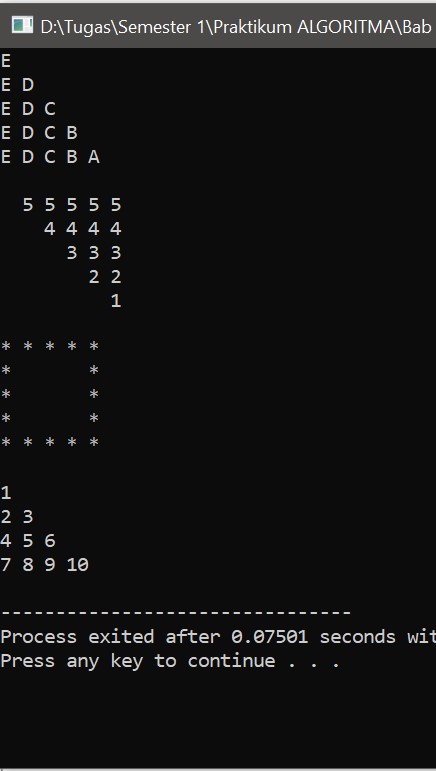
1. Analisa

Progam pertama akan mencetak serangkaian angka dari 22 hingga -20 dengan langkah pengurangan sebesar 5 yang menggunakan perulanagan Do-While. Progam kedua mencetak serangkaian karakter dimulai dari 'F' dan ditambahkan 3 setiap kali melalui *loop* do-while. Progam krtiga mencetak serangkaian karakter dimulai dari 'Y' dan dikurangi 5 setiap kali melalui *loop* do-while hingga huruf 'E'. progam keempat mencetak perulangan angka 4 di tambah nilai perulangan 1 sampai 3, kemudian dilanjutkan dangan serangkaian karakter dimulai dari 'd' dan ditambahkan perulangan 1 sampai 3 setiap kali melalui loop do-while. Kelima menggunakan *loop* do-while untuk mencetak serangkaian angka berdasarkan perubahan nilai variabel keluaran dan *looper*. Keenam hampir sama dengan progam nomer empat.

1. **Latihan Praktikum ke-6 (Nested Loop):**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  for (char a = 'E'; a >= 'A'; a--) {  for (char a1 = 'E'; a1 >= a; a1--) {  cout << a1 <<" ";  }  cout <<endl;  }    cout<<endl;  for ( int b = 5; b >= 1;b--) {  for (int b1 = 5 ; b1 >= b; b1--) {  cout << " ";  }  for (int b2 = 1; b2 <= b; b2++){  cout << b << " ";  }  cout <<endl;  }    cout<<endl;  for(int c = 1; c<=5; c++){  for(int c1 = 1; c1 <= 5; c1++){  if (c==1||c==5||c1==1||c1==5){  cout<<"\* ";  }  else{  cout<<" ";  }  }  cout<<endl;  }    cout<<endl;  for (int d=1;d<=4;d++){  for (int d1=1; d1<=d; d1++){  int d2;  d2++;  cout<<d2<<" ";  }  cout<<endl;  }    return 0;  } |

1. Hasil *Running*



### Gambar 5.6 Hasil *Running* Program Mencetak Pola

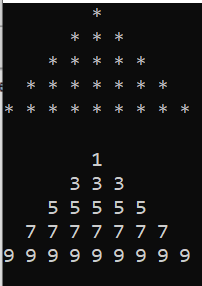
1. Analisa

Program ini menggunakan *Nested Loop* atau perulangan bersarang *for* dengan menggunakan tipe data *int* dan *char*. Didalam program tersebut terdapat 4 program dengan *output* yang berbeda-beda. Dalam program *nested loop* atau mperulangan bersarang, perulangan pertama adalah untuk baris dan perulangan kedua untuk kolom. Di program yang pertama menggunakan tipe data *char* dan perulangan *for*. Di perulangan pertama terdapat variabel a dengan konstanta karakter huruf E, dengan kondisi variabel a lebih kecil sama dengan karakter A dan perubahan - - a atau setiap perubahan dikurangi 1. Di perulangan kedua terdapat variabel b dengan konstanta karakter E, dengan kondisi variabel A lebih kecil sama dengan karakter A, dan perubahan b - - atau bisa disebut juga dengan setiap perubahan dikurangi 1 dan mencetak dengan perintah *cout* variabel b dan disertai “ “ atau spasi dalam setiap mencetak. Di program kedua menggunakan 3 perulangan yang nanti akan mencetak sebuah anak tangga yang terbalik dengan angka 5 dibaris paling atas dan paling banyak angkanya dan angka 1 dengan total hanya 1 angka dibawah sendiri. Di program ketiga akan mencetak sebuah bentuk persegi dengan pinggiran simbol \* dengan cara menggunakan 2 perulangan seperti biasa dan menggunakan percabangan *if else* jika variabel sama dengan 1, 5 maka akan mencetak \* jika tidak akan mencetak spasi. Di program ke 4 akan mencetak sebuah anak tangga dengan berisikan angka yang urut mulai dari angka 1 dari atas dan berakhir di angka 10. Program di selesaikan dengan perintah *return 0*.

1. **Latihan Praktikum ke-7 (Nested Loop)**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  for (int i = 0; i < 5; i++) {  for (int j = 0; j < 5 - i - 1; j++) {  cout << " ";  }  for (int k = 0; k < 2 \* i + 1; k++) {  cout << "\* ";  }  cout << endl;  }    cout<<endl;  int a = 1;  for (int i = 0; i < 5; i++) {  for (int j = 0; j < 5 - i - 1; j++) {  cout << " ";  }  for (int k = 0; k < 2 \* i + 1; k++) {  cout <<a<< " ";  }  cout << endl;  a+=2;  }  return 0;  } |

1. Hasil *Running*



### Gambar 5.7 Hasil *Running* Piramida Bintang dan Urutan Angka

1. Analisa

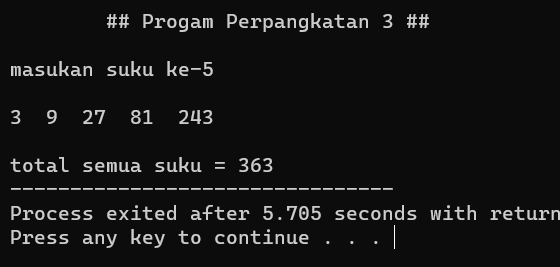
Pada program segitiga bintang menggunakan perulangan *for* dengan tipe data *int* yang dimana program ini akan mencetak sebuah piramida yang berisikan simbil \* dengan menggunakan 3 perulangan perulangan pertama yaitu untuk menentukan berapa baris yang akan digunakan, di perulangan kedua digunakan untuk mencetak spasi bukan simbol \* yang nantinya akan membentuk sebuah piramida, dan di perulangan yang ketiga yang akan mencetak simbol \*. Setelah itu menggunakan perintah *endl* agar *output* mencetak kebawah bukan kesamping. Program diakhiri dengan perintah *return 0.*

Pada program segitiga angka menggunakan 3 perulangan *for* dengan tipe data *int* dan sema seperti program sebelumnya perulangan pertama digunakan untuk mencetak berapa baris yang akan digunakan, pada perulangan ke 2 akan mencetak spasi agar *output* nantinya akan bisa membentuk sebuah piramida. Dan di perulangan ketiga atau terakhir akan menceta isi dari piramida tersebut. Program diakhiri dengan perintah *return 0.*

1. **Tugas Praktikum ke-1 ( For ):**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  cout<<"\t## Progam Perpangkatan 3 ##"<<endl<<endl;  int suku,a,b,c = 3,total;  cout<<"masukan suku ke-";  cin>>suku;  cout <<endl;  for(a=1; a<=suku; a++) {  cout<<c<<" ";  c = c \* 3;  total = total + c / 3;  }  cout<<"\n\ntotal semua suku = "<<total;  return 0;  } |

1. Hasil *Running*



### Gambar 5.8 Hasil *Running* Program Perpangkatan Tiga

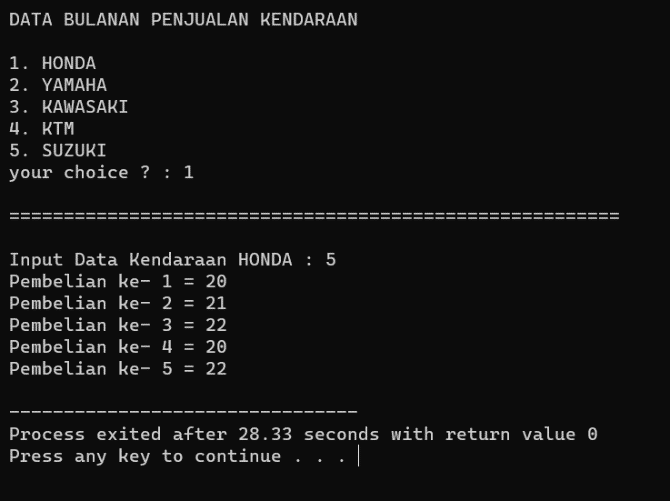
1. Analisa

Pemrogaman diatas manggunakan variabel suku, a, b, c = 3, total, dengan tipedata integer. Kemudian membuat peryataan untuk menginputkan variabel suku, dan dilanjutkan dengan perulangan for dengan kondisi, yaitu a=1; a<=suku; a++ diikuti dengan pernyataan output variabel c dan memproses c = c \* 3 kemudian total = total + c / 3. Setelah for, kan menampilkan output variabel total.

1. **Tugas Praktikum ke-2 (For):**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  cout<<"DATA BULANAN PENJUALAN KENDARAAN"<<endl<<endl;  cout<<"1. HONDA\n2. YAMAHA\n3. KAWASAKI\n4. KTM\n5. SUZUKI"<<endl;  cout<<"your choice ? : ";  int choice,jumlah\_data,data;  cin>>choice;  cout<<"\n========================================================"<<endl<<endl;  string merek;  switch(choice) {  case 1 :  merek="HONDA";  break;  case 2 :  merek="YAMAHA";  break;  case 3 :  merek="KAWASAKI";  break;  case 4 :  merek="KTM";  break;  case 5 :  merek="SUZUKI";  break;  default :  merek="data tidak ada";  cout<<merek<<endl<<endl;  return 0;  }  cout<<"Input Data Kendaraan "<<merek<< " : ";  cin>>jumlah\_data;  for(int a = 1; a<=jumlah\_data; a++) {  cout<<"Pembelian ke- "<<a<<" = ";  cin>>data;  }  } |

1. Hasil *Running*



### Gambar 5.9 Hasil *Running* Program Penjualan Kendaraan

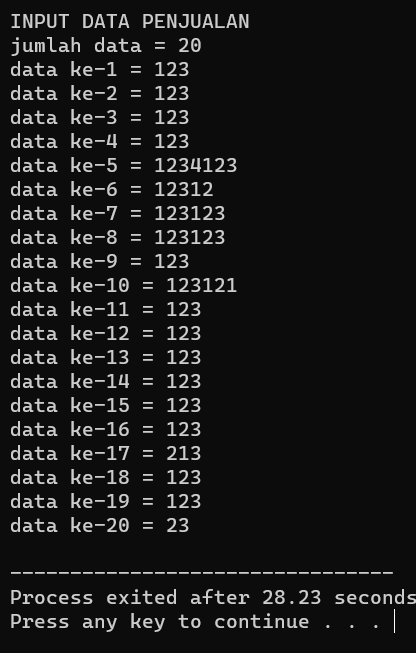
1. Analisa

Pemrogaman diatas menggunakan variabel choice, jumlah\_data, data, a, dengan tipedata integer dan merek menggunakan tipedata . Kemudian membuat peryataan untuk menginputkan variabel choice. Dilanjutkan dengan pernyataan switch case untuk variabel choice yang akan menampilkan variabael merek = “HONDA” untuk case 1, variabael merek = “YAMAHA” untuk case 2, variabael merek = "KAWASAKI" untuk case 3, variabael merek = "KTM" untuk case 4, variabael merek = "SUZUKI" untuk case 5, dan default akan menampilkan variabael merek = "data tidak ada". Kemudian membuat peryataan untuk menginputkan variabel jumlah\_data, dilanjutkan dengan perulangan for dengan pernyataan int a = 1; a<=jumlah\_data; a++ diikuti dengan output a dan input untuk variabel data.

1. **Tugas praktikum ke-3 (For)**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  string hari;  int a,b,jumlah\_nomer,jumlah;  int main() {  cout << "INPUT DATA PENJUALAN" << endl;  cout<<"jumlah data = ";  cin>>jumlah\_nomer;  for (a=1; a<=jumlah\_nomer; a++){  cout << "data ke-"<<a<<" = ";cin>>jumlah;  }  return 0;  } |

1. Hasil *Running*



### Gambar 5.10 Hasil *Running* Progam Input Data

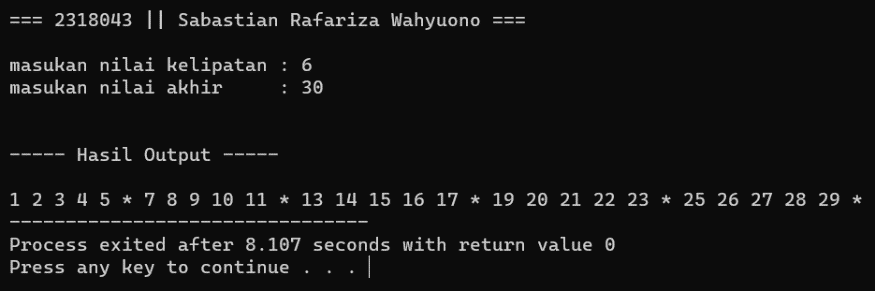
1. Analisa

Pemrogaman di atas menggunakan variabel a, b, jumlah\_nomer, jumlah danagan tipedata integer. Kemudian mengeluarkan stetmen input untuk variabel jumlah\_nomer, dilanjutkan dengan perulangan for dengan stetmen a=1; a<=jumlah\_nomer; a++ diikuti dengan pernyataan output variabel a dan input untuk variabel jumlah.

1. **Tugas Praktikum ke-4 (While)**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  main(){  int kelipatan, nilai\_akhir, i = 1, hasil;  cout << "=== 2318043 || Sabastian Rafariza Wahyuono ===" << endl;  cout << "\nmasukan nilai kelipatan\t: "; cin>>kelipatan;  cout << "masukan nilai akhir\t: "; cin>>nilai\_akhir;  cout << "\n\n----- Hasil Output -----" << endl << endl;  while(i <= nilai\_akhir){  if (i % kelipatan == 0){  cout<<"\* ";  }  else{  cout << i << " ";  }  i++;  }  } |

1. Hasil *Running*



### Gambar 5.11 Hasil *Running* Program angka kelipatan

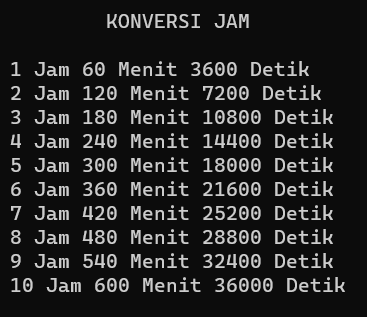
1. Analisa

Progam diatas menggunakan tipedata *integer* dengan variabel kelipatan, nilai\_akhir, i = 1, hasil. Pertama menggunakan perintah input untuk variabel kelipatan dan nilai\_akhir. Kemudian menggunakan perulangan while dengan variabel i <= nilai\_akhir dengan yang diulang menggunakan percabangan if i % kelipatan == 0 kemudian akan mengeluarkan "\* " dan jika tidak atau else akan mengeluarkan variabel i.

1. **Tugas Praktikum ke-5 ( Do While):**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int main() {  int jam = 1, menit = 60, m = 0, detik = 3600, d = 0;  cout<<"\tKONVERSI JAM"<<endl<<endl;  do{  m = m + menit;  d = d + detik;  cout<<jam<<" Jam "<<m<<" Menit "<<d<<" Detik"<<endl;  jam++;  }  while(jam<=10);  } |

1. Hasil *Running*



### Gambar 5.12 Hasil *Running* Program Konversi Jam

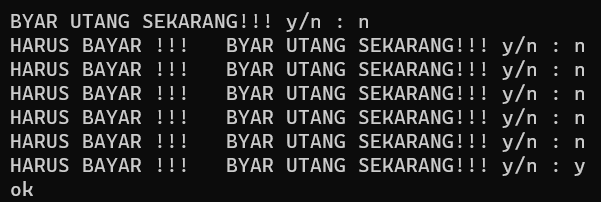
1. Analisa

Progam diatas menggunakan *tipedata* integer dengen variabel int jam = 1, menit = 60, m = 0, detik = 3600, d = 0 dengan rumus masing masing jam menggunakan nili perulangan 1 sampai 10, untuk merubah jam ke menit tinggal menjumblahkan saja dengan 60 pada masing-masing perulangan, untuk mencari detik pun juga begitu.

1. **Tugas praktikum ke-6 (While)**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  char bayar;  do{  cout<<"BYAR UTANG SEKARANG!!! y/n : ";cin>>bayar;  if (bayar =='y'){cout<<"ok";}  else{cout<<"HARUS BAYAR !!! ";}  }  while(bayar != 'y');  return 0;  } |

1. Hasil Running



### Gambar 5.13 Hasil *Running* Program Wajib Membayar Hutang

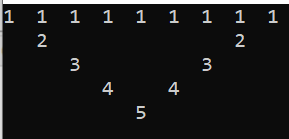
1. Analisa

Program ini adalah program C++ yang menggunakan *loop* do-while untuk terus meminta pembayaran utang sampai pengguna memasukkan karakter 'y'. Pertama meminta pengguna untuk memasukkan karakter 'y' atau 'n' untuk menentukan apakah pembayaran utang akan dilakukan atau tidak. Jika pengguna memasukkan 'y', program mencetak "ok". Jika tidak, program mencetak "HARUS BAYAR !!!".

1. **Tugas Praktikum ke-7 (Nested Loop):**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  for(int a = 1; a <= 5; a++) {  if(a == 1)  {  for(int b = 1; b <= 9; b++)  cout << a << " ";  }  else  {  for(int b = 1; b <= a - 1; b++)  cout << " ";  for(int b = 1; b <= 11 - 2 \* a; b++)  if(b == 1 || b == 11 - 2 \* a)  cout << a << " ";  else  cout << " ";  }  cout << endl;  }  } |

1. Hasil *Running*



### Gambar 5.14 Hasil *Running* Pola Piramida Terbalik

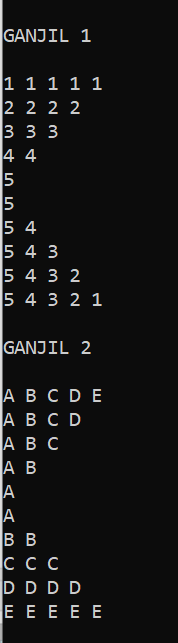
1. Analisa

Di program ini terdapat 4 perulangan dan 2 percabangan *if else* yang menggunakan tipe data *int*. Di perulangan pertama akan digunakan sebagai berapa baris yang akan dibuat dna di program ini menggunakan 5 baris. Lalu di beri percabangan *if else* jika variabel sama dengan 1 maka akan mencetak angka dan diberi “ ” sebagai spasi. Dan jika tidak maka akan mencetak sebuah spasi dan didalam perulangan ada percabangan *if else* lagi yang dimana jika variabel sama dengan 1 atau sama dengan 11 maka akan dikurangi 2 dan dikali dari variabel a yang pertama. Dan jika tidak maka akan mencetak “ “ atau spasi.

1. **Tugas Praktikum ke-8 (Nested Loop)**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int main (){  cout<<"\nGANJIL 1"<<endl<<endl;  int g1=5;  for(int a=1;a<=g1;a++){  for(int b=1;b<=g1-a+1;b++){  cout<<a<<" ";  }  cout<<endl;  }  for(g1;g1>=1;g1--){  for(int a=5;a>=g1;a--){  cout<<a<<" ";  }  cout<<endl;  }    cout<<"\nGANJIL 2"<<endl<<endl;  int g2=5;  for(int a2=0;a2<g2;a2++){  for(int j=0;j<g2-a2;j++){  char a2 = 'A'+j;  cout<<a2<<" ";  }  cout<<endl;  }  for(char b='A';b<='E';b++){  for(char yz='A';yz<=b;yz++){  cout<<b<<" ";  }  cout<<endl;  }  } |

1. Hasil *Running*



### Gambar 5.15 Program *Running* bangun datar NIM Ganjil

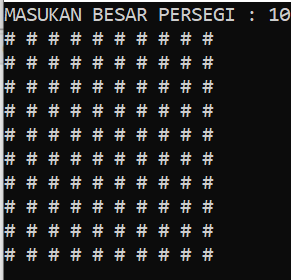
1. Analisa

Program ini tentang *nested loop* atau perulangan bersarang yang menggunakan tipe data *int* dan *char*. Dimana perulangan pertama digunakan untuk mengatur baris yang digunakan berapa, di perulangan kedua untuk kolom. Dan diperulangan ketiga untuk mengisi bilangan berapa yang akan diisi. Setelah itu akan mencetak *output* yaitu seperti bangun datar. Setelah itu program di akhiri i dengan perintah *return 0*.

1. **Tugas praktikum ke-9 (Nested Loop)**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  int isi;  cout<<"MASUKAN BESAR PERSEGI : ";cin>>isi;  for(int a=1;a<=isi;a++){  for(int b=1;b<=isi;b++){  cout<<"# ";  }  cout<<endl;  }  return 0;  } |

1. Hasil *Running*



### Gambar 5.16 Hasil *Running* Pencetak Pola

1. Analisa

Program ini merupakan *nested loop* atau perulangan bersarang menggunakan *input* yang digunakan untuk menjalankan perulangan yang menggunakan 2 perulangan *for*. Dan menggunakan tipe data *int*. salam 2 perulangan tersebut perulangan pertama digunakan untuk mengidentifikasi berapa baris yang digunakan. Dan perulangan kedua berguna untuk berapa kolom yang digunakan didalam perulangan kedua diberi *output* \* agar dapat mencetak \* berbentuk persegi. Program diakhiri dengan *return 0*.

1. **Kesimpulan**
2. Perulangan data atau yang biasa disebut dengan “*looping*” adalah proses yang dilakukan secara berulang-ulang sampai batas yang ditentukan. Perulangan digunakan untuk mengulang sekumpulan perintah sesuai dengan kondisi yang diberikan.
3. Struktur For merupakan salah satu bentuk fungsi *looping*, digunakan untuk melaksanakan pernyataan berulang kali terhadap sejumlah nilai yang telah ditetapkan. *Looping* yang dilakukan oleh *for* telah diketahui batas awal, batas kondisi dan perubahannya.
4. Struktur for di dalam C++ dapat juga melibatkan lebih dari satu variabel namun yang jelas satu diantaranya akan digunakan sebagai indeks pengulangan.
5. Struktur perulangan *WHILE* adalah pengulangan yang melakukan pemeriksaan kondisi di awal blok struktur. Pengulangan hanya akan dilakukan jika kondisi yang didefinisikan di dalamnya terpenuhi (bernilai benar
6. Jika kondisi yang didefinisikan tidak terpenuhi (bernilai salah) maka statemen-statemen yang terdapat dalam blok pengulangan pun tidak akan pernah dieksekusi oleh program.
7. Pada struktur *do*-*while* kondisi ditempatkan di bagian akhir. Hal ini tentu menyebabkan struktur pengulangan ini minimal akan melakukan satu kaliproses walaupun kondisi yang didefinisikan tidak terpenuhi (bernilai salah).
8. Pernyataan nested for adalah suatu perulangan for di dalam perulangan for yang lain. Di dalam penggunaan nested for, perulangan yang di dalam terlebih dahulu dihitung hingga selesai, kemudian perulangan yang di luar diselesaikan.
9. Nested loops digunakan untuk membuat aplikasi matematika yang menggunakan baris dan kolom.
10. Loop luar, biasanya digunakan untuk mendefinisikan kolom. Sedangkan loop dalam, digunakan untuk mendefinisikan baris.

**BAB VI  
Function (Non-Void) dan Prosedure (Void)**

|  |  |  |
| --- | --- | --- |
| **Durasi Pertemuan** | **:** | 120 Menit |
| **Tujuan Praktikum** | **:** | 1. Mampu memahami konsep Fungsi Void dan Non-Void 2. Mampu membuat program dengan bantuan Fungsi Void dan Non-Void |
| **Alat / bahan** | **:** | 1. Seperangkat *computer*. 2. Perangkat lunak: Dev C++. 3. Modul Praktikum Algoritma dan Pemrogramman 2023. |

1. **Landasan Teori**
2. Non Void (Fungsi dengan Nilai Kembalian)

Fungsi merupakan statemen yang dikelompokkan menjadi satu bagian kode (blok program) untuk menyelesaikan tugas spesifik tertentu. Intinya kita ingin **membagi-bagi program dari program utama (main program)** sehingga program dapat lebih rapih dan juga lebih terstruktur dan dapat dijalankan berulang kali.

Dalam sebuah program C++ minimal kita memiliki 1 method, karena pada dasarnya main() juga merupakan sebuah method. Method main adalah program utama kita.

* 1. Jenis-jenis fungsi ada dua yaitu:
     + 1. Non *void* adalah fungsi dengan nilai balik atau fungsi dengan *return* *value* ketika fungsi dipanggil. Fungsi Non *Void*  wajib memiliki *return* *value* ketika ingin digunakan.
       2. *Void* (Prosedur) adalah Fungsi yang tidak mengembalikan nilai atau fungsi yang tidak menghasilkan (*return value*) ketika fungsi dipanggil, atau biasa disebut dengan sebuah prosedur. Namun jika *Void* tetap diberikan *return value*, maka fungsi tersebut tetap bisa dijalankan.
  2. Keuntungan memakai fungsi :

1. Menguraikan tugas pemrograman rumit menjadi langkah-langkah yang lebih sederhana atau kecil.
2. Mengurangi duplikasi kode (kode yang sama ditulis berulang-ulang) dalam program.

**Fungsi *non-void***  disebut juga *function*. Disebut *non-void* karena mengembalikan nilai kembalian yang berasal dari keluaran hasil proses *function* tersebut .

Ciri-ciri dari jenis fungsi *non-void* adalah sebagai berikut:

* + - 1. Ada *keyword* ***return***
      2. Ada tipe data yang mengawali fungsi
      3. Tidak ada *keyword* ***void***
      4. Memiliki nilai kembalian

Bentuk umumnya:

|  |
| --- |
| tipe\_data\_kembalian nama\_fungsi (parameter1, parameter2,…)  {  /\* Badan Fungsi \*/  statemen\_yang\_akan\_dilakukan;  **return** nilai\_yang\_akan\_dikembalikan;  } |

Dari bentuk umum tersebut, yang perlu dijelaskan, yaitu:

* + - 1. tipe\_nilai\_balik

Tipe nilai balik adalah tipe data dari nilai yang akan dikembalikan oleh fungsi. Khusus untuk Fungsi yang tidak ditulis tipenya maka akan dianggap bertipe int atau char.

* + - 1. nama\_fungsi

Nama fungsi adalah nama yang akan digunakan dalam pemanggilan. Pemanggilan fungsi harus disesuaikan dengan daftar parameter yang disertakan pada saat pendefinisian fungsi.

* + - 1. Parameter

Fungsi bisa saja memiliki satu atau lebih parameter. Parameter itu sendiri merupakan variabel berisi nilai yang akan disertakan ke dalam proses yang dilakukan oleh fungsi. Jika tak ada parameter, judul fungsi berupa:

|  |
| --- |
| tipe\_nilai\_balik nama\_fungsi() |

* + - 1. Badan fungsi

Kode atau statemen-statemen yang akan dilaksanakan di dalam fungsi guna menyelesaikan tugas spesifik tertentu. Untuk bagian ini harus diawali dengan tanda **{** dan diakhiri dengan tanda **}.**

* + - 1. Return Value

Perintah return pada fungsi artinya mengembalikan nilai yang tersimpan dalam sebuah variabel kepada kode yang akan memanggil function tersebut. Tujuan dari sebuah return adalah untuk mengembalikan sebuah nilai pada function.

1. Fungsi dengan parameter

Fungsi dengan parameter yang mana parameter ini akan digunakan sebagai nilai masukkan (input) dalam sebuah fungsi. Nilai tersebut kemudian akan diproses oleh fungsi untuk menghasilkan sebuah nilai kembalian (*return* *value*).

Secara teori, parameter yang terdapat pada pendefinisian sebuah fungsi disebut dengan parameter formal, sedangkan parameter yang terdapat pada saat pemanggilan disebut dengan parameter aktual.

1. Fungsi tanpa parameter

Fungsi tanpa parameter bisa didefinisikan sebagai fungsi yang setiap kali pemanggilan fungsi tersebut, hasil yang didapatkan selalu bernilai tetap.

Perlu diingat, bahwa program C++ membaca program selalu dimulai dari main. Inisialisasi fungsi di awal diperlukan, karena sebagai pemberitahuan bahwa ada fungsi diluar dari main. Sehingga, saat eksekusi main, fungsi yang tertera di main akan dieksekusi, walaupun fungsi tersebut diluar dari main.

1. Void (Fungsi tanpa Nilai Balik)

C++ tidak mengenal istilah prosedur (procedure) seperti pada saat kita melakukan pemrograman dengan menggunakan bahasa Pascal. Dalam bahasa Pascal, prosedur didefinisikan sebagai suatu proses yang tidak mengembalikan nilai. Untuk melakukan hal ini di dalam C++, maka harus membuat suatu fungsi dengan tipe void, yang berarti tidak memiliki nilai balik (return value).

Ciri-ciri dari jenis fungsi void adalah sebagai berikut:

1. Secara umum tidak ada keyword return.
2. Tidak adanya tipe data di dalam deklarasi fungsi.
3. Menggunakan keyword void.

Bentuk umum dari pembuatan fungsi tanpa nilai balik ini adalah seperti yang terlihat di bawah ini.

|  |
| --- |
| void nama\_fungsi (parameter1, parameter2, …)  {  Statemen\_yang\_akan\_dilakukan;  ....  } |

Pada contoh diatas, kita memiliki sebuah fungsi void dengan *statement* perulangan*.* Disana, tidak ada statement *return* karenatidak ada nilai yang dikembalikan*.* Untuk pemanggilan fungsi *void* tersebut tinggal menambahkan kata(); didalam fungsi utama.

Dari fungsi void, kita tidak bisa mengembalikan nilai apa pun, tapi kita **bisa mengembalikan sesuatu selain nilai.** Beberapa diantaranya seperti di bawah ini.

1. Return di fungsi void

Fungsi void tidak dapat mengembalikan nilai apa pun. Tapi kita bisa menggunakan pernyataan return. return digunakan untuk menunjukkan bahwa fungsi tersebut **dihentikan**.

1. Fungsi void dapat mengembalikan fungsi void lainnya

Fungsi void dapat me return kan fungsi lain, dengan syarat fungsi tersebut **harus fungsi void** juga.Tampilan :

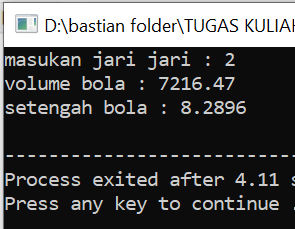
1. Fungsi void dengan banyak parameter

Fungsi void dapat memakai lebih dari satu parameter dengan mendeklarasikan tipe\_data kemudian nama\_parameter.

1. **Langkah – Langkah Praktikum**
2. Buka Aplikasi Dev C++.
3. Buka Modul Praktikum Algoritma dan Pemrogramman.
4. Menjalankan script dari modul.
5. **Latihan Praktikum ke-1: Program menghitung Volume Bola**
6. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  float volume(int r){  r+=10;  float v = 1.33\*3.14\*r\*r\*r;  return v;  }  float volume\_setengah(int r){  float vs = 0.66\*3.14\*r\*r;  return vs;  }  int main() {  int r;  cout << "masukan jari jari : "; cin>>r;  cout << "volume bola : "<<volume(r) <<endl;  cout << "setengah bola : "<<volume\_setengah(r)<<endl;  return 0;  } |

1. Hasil Running



### Gambar 6.1 Hasil *Running* Program Menghitung Volume

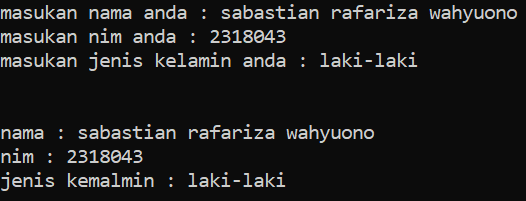
1. Analisa

Program ini menggunakan dua fungsi dengan parameter, yaitu Kalkulasi Bola yang menghitung volume bola berdasarkan rumus (4/3)πr³ dan Kalkulasi Setengah Bola yang menghitung volume setengah bola berdasarkan rumus 0.5 \* (4/3)πr³ dengan nilai r yang diambil dari variabel nim. Perhitungan volume bola dan setengah bola menggunakan fungsi-fungsi tersebut dengan nilai radius yang sama. Hasil perhitungan ditampilkan sebagai output yang menyatakan volume bola dan volume setengah bola dengan radius yang telah dihitung.

1. **Latihan Praktikum ke-2: Program Mencetak Biodata Mahasiswa**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  void biodata(const string& nama, const string& nim, const string& kelamin){  cout << "nama : "<< nama <<endl;  cout << "nim : "<<nim<<endl;  cout << "jenis kemalmin : "<<kelamin<<endl;  }  int main() {  string nama, nim, kelamin;  cout << "masukan nama anda : ";  getline(cin,nama);  cout << "masukan nim anda : ";  getline(cin,nim);  cout << "masukan jenis kelamin anda : ";  getline(cin,kelamin);  cout << endl<<endl;  biodata(nama,nim,kelamin);  return 0;  } |

1. Hasil Running



### Gambar 6.2 Hasil Running Program Biodata Mahasiswa

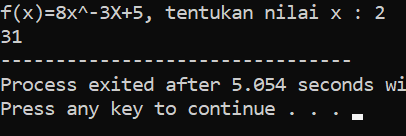
1. Analisa

Program diawali dengan mengimpor modul iostream. Kemudian, terdapat fungsi string. Fungsi ini digunakan untuk mencetak biodata mahasiswa dengan informasi nama, NIM, dan jenis kelamin. Program meminta pengguna untuk memasukkan biodata, seperti nama, NIM, dan jenis kelamin. Input nama menggunakan `getline` untuk menerima lebih dari satu kata. Dilanjutkan dengan tampilkan biodata sesuai data yang telah dimasukkan.

1. **Latihan Praktikum ke-3: Program Sub-rutin Menentukan nilai fungsi**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int prnuh(int a){  int b = 8\*(a\*a)-3\*a+5;  return b;  }  int main() {  int x;  cout << "f(x)=8x^-3X+5, tentukan nilai x : ";cin >> x;  cout << prnuh(x);  return 0;  } |

1. Hasil Running



### Gambar6.3 Hasil Running Program menentukan nilai fungsi

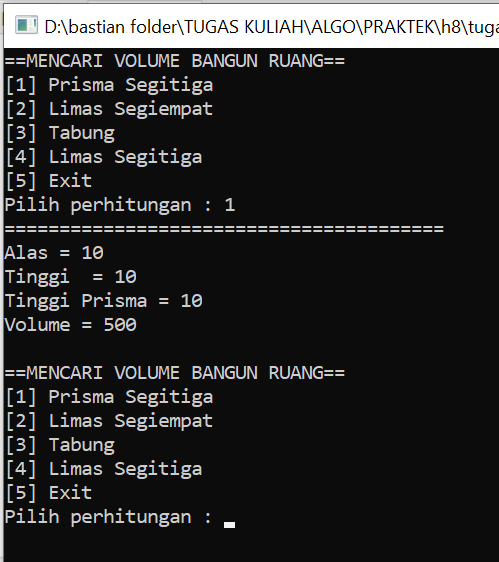
1. Analisa

Program diawali dengan using namespace std agar tidak mengetik nama class pada awal fungsi cout dan cin. Program akan mengawali pemanggilan kepada fungsi main(). Fungsi kalkulasiFungsi() digunakan untuk menghitung nilai dari fungsi matematika f(x) = 8x^2 - 3x + 5 dengan menerima parameter x bertipe int. Di dalam fungsi main(), sebuah variabel x diinisialisasi dengan nilai 2. Nilai x kemudian diubah menjadi hasil perhitungan dari fungsi kalkulasiFungsi(x). Hasil perhitungan tersebut kemudian dicetak menggunakan cout.

1. **Tugas Praktikum ke-1:**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  int prisma\_segitiga(int a, int t, int tp){  double v = 0.5\*a\*t\*tp;  return v;  }  int limas\_segiempat(int la, int t){  double v = 0.33\*la\*t;  return v;  }  int tabung(int r, int t){  double v = 3.14\*r\*r\*t;  return v;  }  int limas\_segitiga(int la, int tl){  double v = 0.33\*la\*tl;  }  int main(){  int pilih, ulang = 1, a;    do{  cout << "==MENCARI VOLUME BANGUN RUANG=="<<endl;  cout <<"[1] Prisma Segitiga"<<endl;  cout <<"[2] Limas Segiempat"<<endl;  cout <<"[3] Tabung"<<endl;  cout <<"[4] Limas Segitiga"<<endl;  cout <<"[5] Exit"<<endl;  cout <<"Pilih perhitungan : ";cin>>pilih;  cout <<"========================================"<<endl;  switch (pilih){  case 1 ... 5 :  if(pilih == 1){  int a, t, tp;  cout << "Alas = ";cin>>a;  cout << "Tinggi = ";cin>>t;  cout << "Tinggi Prisma = ";cin>>tp;  cout << "Volume = "<<prisma\_segitiga(a,t,tp)<<endl<<endl;  ulang = 0;  }  else if(pilih == 2){  int t,la;  cout << "Tinggi = ";cin>>t;  cout << "Luas Alas = ";cin>>la;  cout << "Volume = "<<limas\_segiempat(la,t)<<endl<<endl;  ulang = 0;  }  else if(pilih == 3){  int j,t;  cout << "Jari Jari = ";cin>>j;  cout << "tinggi = ";cin>>t;  cout << "Volume = "<<tabung(j,t)<<endl<<endl;  ulang = 0;  }  else if(pilih == 4){  int lt,la;  cout << "Tinggi Limas = ";cin>>lt;  cout << "Luas Alas = ";cin>>la;  cout << "Volume = "<<limas\_segitiga(lt,la)<<endl<<endl;  ulang = 0;  }  else{  cout<<"Anda Telah keluar";  }  break;  default:  ulang = 0;  break;  }  }  while(ulang == 0);  return 0;  } |

1. Hasil Running



### Gambar 6.4 Hasil Running Program volume bangun ruang

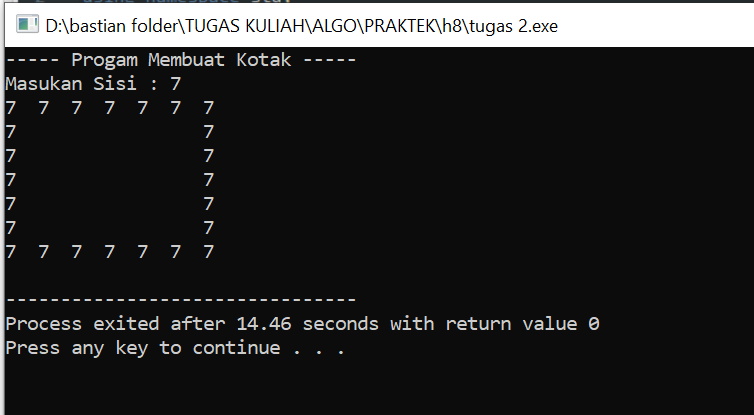
1. Analisa

Program diawali dengan fungsi-fungsi yang Kalkulasi balok, kubus, kerucut dan tabung. Di dalam fungsi main, program menampilkan menu untuk memilih jenis bangun ruang yang akan dihitung volumenya (kubus, balok, tabung, kerucut, atau keluar dari program). User diminta untuk memilih, kemudian program menggunakan switch case untuk menangani pilihan tersebut. Jika pilihan tidak ada, maka progam akan diulang. Hasil perhitungan volume ditampilkan menggunakan cout. Program berakhir jika user memilih opsi selesai dengan menampilkan pesan “Anda Telah Keluar" pada program.

1. **Tugas Praktikum ke-2:**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  void kotak(int sisi){  for (int i = 1; i<=sisi; i++){  for (int a = 1; a<=sisi; a++){  if (i==1||a==1||i==sisi||a==sisi){  if(sisi>=10){  cout << sisi <<" ";  }  else{cout << sisi <<" ";  }  }else{  cout << " ";  }  }  cout << endl;  }  }  int main(){  int sisi;  cout << "----- Progam Membuat Kotak -----"<<endl;  cout << "Masukan Sisi : ";cin >>sisi;  kotak(sisi);  return 0;  } |

1. Hasil Running



### Gambar 6.5 Hasil Running Program Membuat Persegi

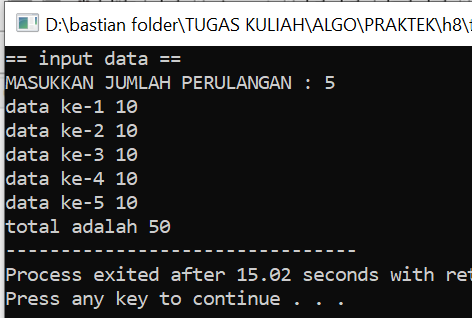
1. Analisa

Program ini digunakan untuk membuat segi empat dengan ukuran sesuai input sisi yang diberikan. Program menggunakan nested loop (loop bersarang) untuk mengatur pola output sesuai dengan sisi yang dimasukkan. Program meminta pengguna untuk memasukkan sisi yang akan digunakan untuk membuat segi empat.Setelah itu program selesai.

1. **Tugas praktikum ke-3**
2. Source Code

|  |
| --- |
| # include <iostream>  using namespace std;  void data(int jumlah) {  int data, total = 0;  for(int i = 1; i <= jumlah; i++) {  cout <<"data ke-"<<i<<" ";cin>>data;  total = total + data;  }  cout <<"total adalah "<<total;  }  int main() {  int jumlah;  cout << "== input data ==" << endl;  cout << "MASUKKAN JUMLAH PERULANGAN : ";cin >> jumlah;  data(jumlah);  return 0;  } |

1. Hasil Running



### Gambar 6.6 Program Perulangan *Input* Data

1. Analisa

Program yang menggunakan fungsi untuk melakukan perulangan untuk menginput data dengan menggunakn fungsi void. Pada fungsi perulangan, parameter pertama jumlah merupakan jumlah input yang ingin kita masukan. Selanjutnya, program akan mencetak nomer data yang ingin kita masukan dengan menggunakan menggunakan perintah *cin* dan *cout*.

1. **Kesimpulan**
2. Fungsi merupakan statemen yang dikelompokkan menjadi satu bagian kode (blok program) untuk menyelesaikan tugas spesifik tertentu. Intinya kita ingin membagi-bagi program dari program utama (main program) sehingga program dapat lebih rapih dan juga lebih terstruktur dan dapat dijalankan berulang kali.
3. Fungsi bisa saja memiliki satu atau lebih parameter. Parameter itu sendiri merupakan variabel berisi nilai yang akan disertakan ke dalam proses yang dilakukan oleh fungsi.
4. Fungsi tanpa parameter bisa didefinisikan sebagai fungsi yang setiap kali pemanggilan fungsi tersebut, hasil yang didapatkan selalu bernilai tetap.

**BAB VII  
Array**

|  |  |  |
| --- | --- | --- |
| **Durasi Pertemuan** | **:** | 120 Menit |
| **Tujuan Praktikum** | **:** | 1. Mampu memahami konsep array 2. Mampu membuat program penerapan array |
| **Alat / bahan** | **:** | 1. Seperangkat *computer*. 2. Perangkat lunak: Dev C++. 3. Modul Praktikum Algoritma dan Pemrogramman 2023. |

1. **Landasan Teori**
2. Pengertian Array

Selama ini kita menggunakan satu variabel untuk menyimpan 1 buah nilai dengan tipe data tertentu. Misalnya :

|  |
| --- |
| int a1, a2, a3, a4, a5; |

Deklarasi variabel diatas digunakan untuk menyimpan 5 data integer dimana masing masing variabel diberi nama a1, a2, a3, a4, dan a5. Jika kita memiliki 10 data, 100 data *integer* bahkan mungkin data yang ingin kita proses tidak kita ketahui atau bersifat dinamis? Kita tidak mungkin menggunakan variabel seperti diatas.

*Array* adalah sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama. setiap data tersebut menempati lokasi atau alamat *memory* yang berbeda-beda dan selanjutnya disebut dengan elemen array. Elemen *array* tersebut kemudian dapat kita akses melalui indeks yang terdapat di dalamnya namun penting sekali untuk diperhatikan bahwa dalam C++, Indeks *array* selalu di mualai dari 0, bukan 1.

Untuk mendeklarasian sebuah *array* kita harus menggunakan tanda [..](*bracket*). Bentuk umum dari variabel *array* dapat ditulis seperti dibawah ini :

|  |
| --- |
| Tipe\_data nama\_variabel[indeks]; //ketentuan  int data[4]; //contoh |

Pada contoh diatas, data yang dapat ditampung yaitu 4 data. Indeks *array* selalu dimulai dari angka 0. Sehingga pada kasus diatas, data yang tersimpan dimulai dari indeks ke 0 hingga 3.

Namun anda juga dapat mendeklarasikan *array* sekaligus memberikan inisialisasi tanpa harus memberikan tambahan deklarasi ukuran *array*, sebagai berikut:

|  |
| --- |
| int daftarNilai [] = {10,9,8,10,9}; |

1. Array 1 Dimensi

Contoh sebelumnya yang kita praktekkan yaitu merupakan *array* 1 dimensi. Cara diatas merupakan cara untuk melakukan inisialisasi. Untuk mengisikan data ke dalam *element*-*element* *array* kita dapat melakukannya langsung untuk setiap *element* , contohnya :

|  |
| --- |
| DATA[0]=1  DATA[1]=2  DATA[2]=3  DATA[3]=4  ......  Dst |

Namun cara di atas tidak efisien karena jika memerlukan data yang banyak harus menuliskan *script* yang banyak juga. Cara yang lebih umum dan banyak di gunakan untuk mengisikan nilai ke dalam element *array* adalah dengan menggunakan perulangan (*looping*).

1. Array 2 Dimensi

Hampir sama dengan *array* 1 dimensi, yang membuat *array* 2 dimensi berbeda adalah daya tampung datanya. Jika di *array* 1 dimensi, kita perlu inisialisasi 1 angka untuk menunjukkan berapa data yang bisa dimasukkan. Di *array* 2 dimensi, kita bisa menginisialisasikan 2 angka. *Array* seperti ini sering di gunakan untuk pemerosesan matrik, Bentuk umum dari array 2 dimensi adalah sebagai berikut :

|  |
| --- |
| tipedata nama\_array[jmlh baris][jmlah kolom];  float bil[2][2]; |

Ilustrasi tabel array sebagai berikut :

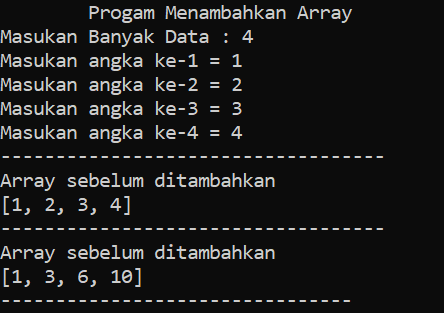
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [0][0] | [0][1] | [0][2] | [0][3] | [0][4] |
| [1][0] | [1][1] | [1][2] | [1][3] | [1][4] |

Contoh diatas menunjukkan bahwa kita dapat mengisi data dari array 00, 01, 02, 03, 04, 10, 11 dan seterusnya. Sehingga, data yang diisi totalnya menjadi 2 x 5 = 10 data.

1. **Langkah – Langkah Praktikum**
2. Buka Aplikasi Dev C++.
3. Buka Modul Praktikum Algoritma dan Pemrogramman.
4. Menjalankan script dari modul.
5. **Latihan Praktikum ke-1:**
6. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  int jumlahdata;    cout << "\tProgam Menambahkan Array"<<endl;  cout << "Masukan Banyak Data : ";cin>>jumlahdata;  int jumlah[jumlahdata];    for(int i=1;i<=jumlahdata;i++){  cout<<"Masukan angka ke-"<<i<<" = ";cin>>jumlah[i];  }    cout << "-----------------------------------"<<endl;  cout << "Array sebelum ditambahkan"<<endl;  cout<<"[";  for (int i=1; i <=jumlahdata; i++){  cout<<jumlah[i];  if (i<jumlahdata)cout<<", ";  }  cout<<"]"<<endl;  for (int i=1; i <=jumlahdata; i++){  jumlah[i]+=jumlah[i-1];  }  cout << "-----------------------------------"<<endl;  cout << "Array sebelum ditambahkan"<<endl;  cout<<"[";  for (int i=1; i <=jumlahdata; i++){  cout<<jumlah[i];  if (i<jumlahdata)cout<<", ";  }  cout<<"]";  return 0;  } |

1. Hasil *Running*



### Gambar 9.1 Hasil Running Program Menambahkan Array

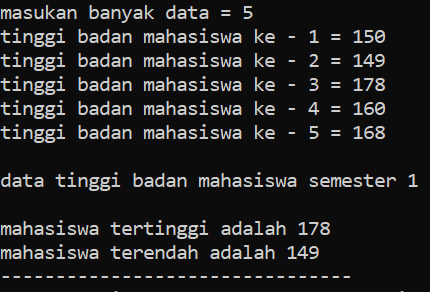
1. Analisa

Program meminta *user* untuk memasukkan jumlah data, kemudian menggunakan jumlah tersebut untuk mendefinisikan index *array* dan banyak perulangan, lalu didalam perulangan *user* kemudian diminta untuk memasukkan angka kedalam *array sebanyak* jumlah perulangan atau data yang telah dimasukkan tadi. Setelah semua angka dimasukkan, program akan menampilkan *array* menggunakan perulangan yang sama, kemudian program menjumlahkan elemen-elemen dari *array* data secara kumulatif ke dalam *array* data2, lalu program menampilkan *array* baru data2 yang berisi hasil penjumlahan kumulatif dari *array* data.

1. **Latihan Praktikum ke-2:**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  int data;  cout << "masukan banyak data = ";cin>>data;  int tinggi[data], tertinggi, terendah;  for (int a=0; a<data; a++){  cout << "tinggi badan mahasiswa ke - "<<a+1<<" = ";  cin>>tinggi[a];  if(a==0){  tertinggi = tinggi[a];  terendah = tinggi[a];  }  else{  if (tertinggi<tinggi[a])tertinggi = tinggi[a];  if (terendah>tinggi[a])terendah = tinggi[a];  }  }  cout << "\ndata tinggi badan mahasiswa semester 1"<<endl<<endl;  cout <<"mahasiswa tertinggi adalah "<<tertinggi<<endl<<"mahasiswa terendah adalah "<<terendah;  return 0;  } |

1. Hasil *Running*



### Gambar 9.2 Hasil Running Program Data Tinggi Badan

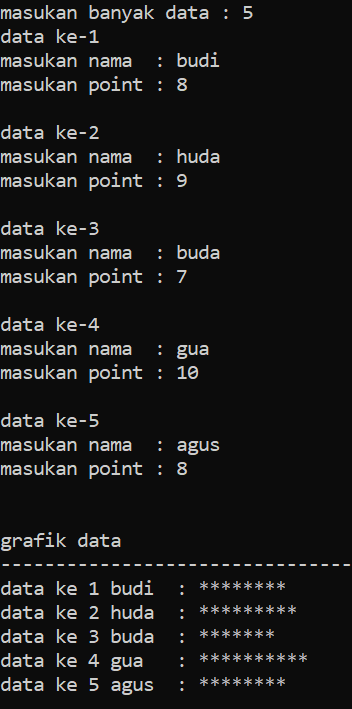
1. Analisa

Pada program ini *user* diminta untuk memasukkan banyaknya data tinggi badan, kemudian *user* diminta memasukkan tinggi badan mahasiswa sebanyak data yang di *input* kan tadi,lalu setelah itu akan dibandingkan menggunakan *if else* untuk mencari tinggi badan yang tertinggi dan terendah.

1. **Latihan Praktikum ke-3**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  int data;  cout << "masukan banyak data : ";cin>>data;  string nama[data];  int point[data];    for(int a=0; a<data; a++){  cout <<"data ke-"<<a+1<<endl;  cout <<"masukan nama : ";cin>>nama[a];  cout <<"masukan point : ";cin>>point[a];  cout <<endl;  }  cout<<"\ngrafik data"<<endl;  cout<<"--------------------------------"<<endl;  for(int b=0; b<data; b++){  cout <<"data ke "<<b+1<<" "<<nama[b]<<"\t: ";  for(int c=0; c<point[b]; c++){  cout<<"\*";  }  cout<<endl;  }  return 0;  } |

1. Hasil *Running*



### Gambar 9.3 Hasil Running Program Grafik Poin

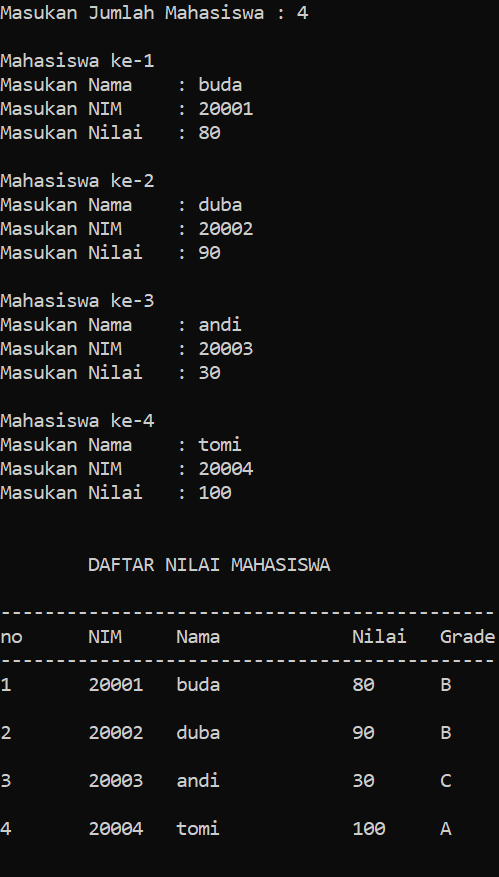
1. Analisa

Program di atas adalah untuk menampilkan grafik poin dalam bentuk bintang.lalu menggunakan inputan dari user untuk jumlah berapa banyak data *array* yang diinginkan,setelah itu *array* yang tadinya berjumlah dari inputan user akan ditampilkan menjadi bintang dengan menggunakan perintah *for.*

1. **Tugas Praktikum ke-1:**
2. Source Code

|  |
| --- |
| #include<iostream>  using namespace std;  void garis(int panjang){  for(int g=1; g<=panjang; g++){  cout<<"-";  }cout<<endl;  }  int main(){  int jumlah;  cout<<"Masukan Jumlah Mahasiswa : ";cin>>jumlah;    string nama[jumlah];  int nim[jumlah];  int nilai[jumlah];    for (int b=0; b<jumlah; b++){  cout<<"\nMahasiswa ke-"<<b+1<<endl;  cout<<"Masukan Nama\t: ";cin>>nama[b];  cout<<"Masukan NIM\t: ";cin>>nim[b];  cout<<"Masukan Nilai\t: ";cin>>nilai[b];  }  cout<<"\n\n\tDAFTAR NILAI MAHASISWA\n"<<endl;  garis(45);  cout<<"no\tNIM\tNama\t\tNilai\tGrade"<<endl;  garis(45);  for (int a=0; a<jumlah; a++){  char grade;  if(nilai[a]<=65){  grade = 'C';  }  else if(nilai[a]<=90){  grade = 'B';  }  else if(nilai[a]<=100){  grade = 'A';  }  cout<<a+1<<"\t"<<nim[a]<<"\t"<<nama[a]<<"\t\t"<<nilai[a]<<"\t"<<grade;  cout<<endl<<endl;  }  return 0;  } |

1. Hasil *Running*



### Gambar 9.4 Hasil Running Program Pendataan Nilai Mahasiswa

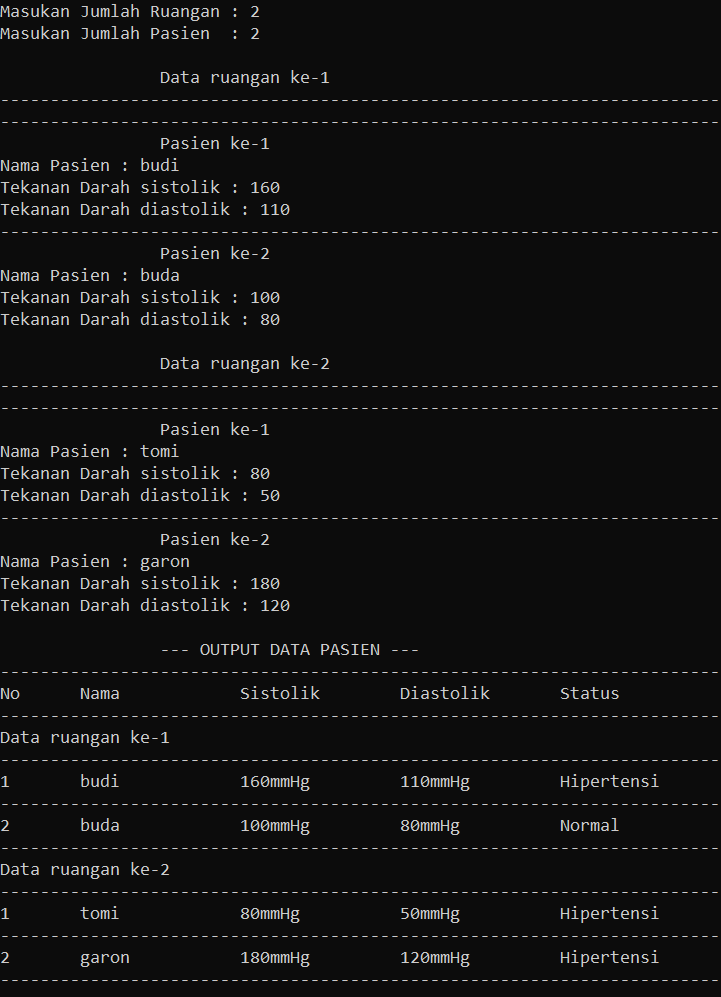
1. Analisa

Program diatas adalah program untuk pendataan nilai dari mahasiswa,dengan cara pengguna diminta memasukkan data mahasiswa, yaitu nama dan NIM dari mahasiswa, lalu setelah itu nilai akan dicek menggunakan perintah *if else* untuk mengetahui *grade* yang terdiri dari ‘F’ sampai dengan ‘A’.

1. **Tugas Praktikum ke-2:**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  void garis(){  cout<<"------------------------------------------------------------------------"<<endl;  }  int main() {  int ruangan, pasien;  cout<<"Masukan Jumlah Ruangan : ";cin>>ruangan;  cout<<"Masukan Jumlah Pasien : ";cin>>pasien;  int sis[ruangan][pasien], dia[ruangan][pasien];  string nama[ruangan][pasien];    for (int a = 0; a < ruangan; a++) {  cout<<"\n\t\tData ruangan ke-"<<a+1<<endl;  garis();  for(int b = 0; b < pasien; b++) {  garis();  cout<<"\t\tPasien ke-"<<b+1<<endl;  cout<<"Nama Pasien : ";cin>>nama[a][b];  cout<<"Tekanan Darah sistolik : ";cin>>sis[a][b];  cout<<"Tekanan Darah diastolik : ";cin>>dia[a][b];  }  }  cout<<"\n\t\t--- OUTPUT DATA PASIEN ---"<<endl;  garis();  cout<<"No\tNama\t\tSistolik\tDiastolik\tStatus"<<endl;  garis();  for(int a = 0; a < ruangan; a++) {  cout<<"Data ruangan ke-"<<a+1<<endl;  garis();  for (int b = 0; b < pasien; b++) {  string status;  if (sis[a][b]>140 && dia[a][b]>80){  status = "Hipertensi";  }  else if(sis[a][b]<90 && dia[a][b]<60){  status = "Hipertensi";  }else{  status = "Normal";  }  cout<<b+1<<"\t"<<nama[a][b]<<"\t\t"<<sis[a][b]<<"mmHg\t\t"<<dia[a][b]<<"mmHg\t\t"<<status<<endl;  garis();  }  }  return 0;  } |

1. Hasil *Running*



### Gambar 9.5 Hasil Running Program Pendataan Tekanan Darah

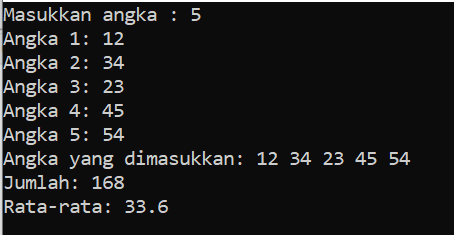
1. Analisa

Program diatas adalah untuk memantau tekanan darah pasien dalam beberapa ruangan. Pertama,memasukkan berapa ruangan yang diinginkan,begitu juga dengan pasien,setelah itu memasukkan data diri dari pasien yang dimasukkan,setelah itu darah akan di cek menggunakan *if-else* untuk menampilkan apakah darah normal atau hipotensi,setelah itu ditampilkan menggunakan *cout.*

1. **Tugas praktikum ke-3**
2. Source Code

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  int ukuran;  cout << "Masukkan angka : ";cin >> ukuran;  int angka[ukuran];  for (int i = 0; i < ukuran; ++i) {  cout << "Angka " << i + 1 << ": ";  cin >> angka[i];  }    int jumlah = 0;  for (int i = 0; i < ukuran; ++i) {  jumlah += angka[i];  }    double rata\_rata = static\_cast<double>(jumlah) / ukuran;    cout << "Angka yang dimasukkan: ";  for (int i = 0; i < ukuran; ++i) {  cout << angka[i] << " ";  }  cout << endl;    cout << "Jumlah: " << jumlah << endl;  cout << "Rata-rata: " << rata\_rata << endl;    return 0;  } |

1. Hasil *Running*



### Gambar 8.6 Running Program Nilai Rata - Rata Array

1. Analisa

Progam diatas merupakan progam untuk mencari rata-rata menggunakan array, dengan menggunakan perulangan for dan setiap perulangan diminta untuk memasukan isi *index array.* Selain itu masing-masing index array yang diinputkan juga dijumlahkan dengan menggunakan variable yang berbeda . kemudian setelah perulangan tersebut selesai variable yang diggunakan untuk menjumlah dibagi dengan jumlah perulangan.

1. **Kesimpulan**
2. *Array* adalah sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama. setiap data tersebut menempati lokasi atau alamat *memory* yang berbeda-beda dan selanjutnya disebut dengan elemen *array*.
3. Elemen *array* tersebut kemudian dapat kita akses melalui indeks yang terdapat di dalamnya namun penting sekali untuk diperhatikan bahwa dalam C++, Indeks *array* selalu di mualai dari 0, bukan 1.
4. Untuk mendeklarasian sebuah *array* kita harus menggunakan tanda [..](*bracket*).

# KESIMPULAN

Dengan adanya kegiatan praktikum Algoritma dan Pemrogramman. Penulis dapat menyimpulkan sebagai berikut :

1. C++ merupakan bahasa pemrograman yang tinggi dan sensitif terhadap sebuah kalimat,kata,tanda baca,jadi kita haru lebih berkonsentrasi dalam mengerjakan sesuatu dialam bahasa pemrograman C++.
2. Operator merupakan simbol yang biasa dilibatkan dalam program untuk melakukan suatu operasi atau manipulasi.
3. If else adalah suatu pemilihan statement yang akan dieksekusi di mana pemilihan tersebut didasarkan atas kondisi tertentu.
4. Pernyataan *switch* adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan pernyataan, berdasarkan nilai dari sebuah ungkapan dan nilai penyeleksian.
5. Perulangan data atau yang biasa disebut dengan “*looping*” adalah proses yang dilakukan secara berulang-ulang sampai batas yang ditentukan.
6. Struktur For merupakan salah satu bentuk fungsi *looping*, digunakan untuk melaksanakan pernyataan berulang kali terhadap sejumlah nilai yang telah ditetapkan.
7. Struktur perulangan *WHILE* adalah pengulangan yang melakukan pemeriksaan kondisi di awal blok struktur.
8. Struktur perulangan *do*-*while* adalah pengulangan yang melakukan pemeriksaan kondisdi bagian akhir.
9. Pernyataan nested for adalah suatu perulangan for di dalam perulangan for yang lain
10. Fungsi merupakan statemen yang dikelompokkan menjadi satu bagian kode (blok program) untuk menyelesaikan tugas spesifik tertentu.
11. *Array* adalah sebuah variabel yang menyimpan sekumpulan data yang memiliki tipe sama.

# DAFTAR PUSTAKA

Laboratorium Pemrograman Komputer. 2023. *Modul Praktikum Algoritma Dan Pemrogramman.* Laboratorium Pemrograman Komputer Teknik Informatika Insitut Teknologi Nasional Malang.

**LEMBAR ASISTENSI PRAKTIKUM ALGORITMA DAN PEMROGRAMAN**

**SEMESTER GANJIL TAHUN AKADEMIK 2023/2024**

**Nama : Sabastian Rafariza Wahyuono**

**NIM : 2318043**

**Kelompok : 12**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Tanggal** | **Asistensi** | | | | **Paraf** |
| **Konsep** | | **Program** | **Hasil Akhir** |  |
| **1** |  | **Instruktur** | Pengenalan C++  Operator dan Statement I/O  Percabangan  Perulangan  Fungsi Void dan Non-Void  Array |  |  |  |
| **2** |  | **Dosen** |  |  |  |  |
| **Batas Akhir:** | | | | | | |

**Asisten Laboratorium**

**(Muhammad Alief Atho'illah)**

2218037

Malang, Desember 2023

**Dosen Pembimbing,**

**(Joseph Dedy Irawan, ST, MT)**

**NIP.197404162005011002**