



DASAR LOGIKA ALGORITMA & PEMROGRAMAN DENGAN BAHASA



SRI HARYATI

DASAR LOGIKA ALGORITMA DAN PEMROGRAMAN DENGAN BAHASA C++

Undang-Undang No. 28 Tahun 2014 Tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Perlindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap :

- i. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- iv. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp 100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).
3. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp 1.000.000.000,00 (satu miliar rupiah).
4. Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp 4.000.000.000,00 (empat miliar rupiah).

DASAR LOGIKA ALGORITMA DAN PEMROGRAMAN DENGAN BAHASA C++

Sri Haryati



Poliban Press

**DASAR LOGIKA ALGORITMA DAN PEMROGRAMAN DENGAN
BAHASA C++**

**Penulis :
Sri Haryati**

**ISBN :
978-623-7694-91-5**

**ISBN Elektronik:
978-623-7694-92-2 (PDF)**

**Editor dan Penyunting :
Reza Fauzan**

**Desain Sampul dan Tata letak :
Eko Sabar Prihatin; Rahma Indera**

**Penerbit :
POLIBAN PRESS**
Anggota APPTI (Asosiasi Penerbit Perguruan Tinggi Indonesia)
no.004.098.1.06.2019
Cetakan Pertama, 2022

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk
dan dengan cara apapun tanpa ijin tertulis dari penerbit

Redaksi :
Politeknik Negeri Banjarmasin, Jl. Brigjen H. Hasan Basry,
Pangeran, Komp. Kampus ULM, Banjarmasin Utara
Telp : (0511)3305052
Email : press@poliban.ac.id

Diterbitkan pertama kali oleh :
Poliban Press, Banjarmasin, Januari 2022

PRAKATA

Assalamu'alaikum wa rahmatullahi wa barakatuh.

Puji dan syukur kehadiran Allah SWT atas berkat rahmat dan karuniaNya-lah buku ajar Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++ dapat diselesaikan dengan tepat waktu dan semoga dapat bermanfaat sebagai media pembelajaran bagi para mahasiswa yang mengambil mata kuliah Logika Algoritma dan Pemrograman semester 2 (dua) pada Progam Studi D3 Komputerisasi Akuntansi di Jurusan Akuntansi Politeknik Negeri Banjarmasin, disamping itu buku ajar ini dapat digunakan sebagai salah satu literatur dalam memperkuat pemahaman dalam praktikum logika algoritma dan pemrograman dengan Bahasa C++.

Buku Ajar Dasar Logika Algoritma dan Pemrograman dengan bahasa C++ terdiri dari teori dasar pemahaman tentang logika, algoritma, alat bantu dalam menggambarkan algoritma (flowchart), Pseudocode, konsep pemrograman C++, struktur data, hingga program-program dalam contoh kasus akuntansi.

Pada kesempatan ini penulis mengucapkan terima kasih kepada pihak-pihak yang telah memberikan dukungan dan kesempatan untuk membuat buku ajar ini. Besar harapan agar buku ini dapat bermanfaat.

Banjarmasin, Agustus 2021

Penyusun,

Sri Haryati, S.Kom., M.Kom

DAFTAR ISI

PRAKATA	v
DAFTAR ISI	vi
BAB I PENDAHULUAN	1
1.1. Logika	1
1.2. Algoritma	1
1.3. Flowchart	2
1.4. Pseudocode.....	4
1.5. Latihan Soal	5
BAB II Pengenalan Bahasa C++	6
2.1 Pengantar Bahasa C++	6
2.2 Format Perintah dalam C++	6
2.3 Perintah Sederhana C++.....	7
2.4 Instalasi Dev C++	9
2.5 Latihan Soal	12
BAB III KONSEP PEMROGRAMAN C++	13
3.1 Tipe Data.....	13
3.2 Variabel.....	14
3.3 Konstanta	14
3.4 Contoh Soal dan penyelesain	15
3.5 Latihan Soal	23

BAB IV Struktur Data (ARRAY)	24
4.1 Array	24
4.2 Array Satu Dimensi	25
4.3 Deklarasi Array	26
4.4 Menyiapkan Array satu dimensi Bahasa pemrograman C++ 26	
4.5 Latihan Soal	48
BAB V ALGORITMA DALAM AKUNTANSI	49
5.1 Program Penjualan Bahasa C++	50
5.2 Program Pembelian Bahasa C++	54
5.3 Program Kasir Bahasa C++	55
5.4 Program Pengajian Bahasa C++	57
5.5 Program Saldo Bahasa C++	61
DAFTAR PUSTAKA	79

BAB I

PENDAHULUAN

Capaian Pembelajaran:

1. Memahami Logika
2. Memahami Algoritma
3. Memahami Flowchat
4. Memahami Pseudocode
5. Memahami Bahasa C++

1.1. Logika

Adapun beberapa pengertian dari logika dengan istilah, “marilah kita berfikri dengan logis” atau dengan pengertian “hari kita menggunakan logika kita” kiranya perkataan seperti ini kerap kali kita dengar di kehidupan sehari-hari. Secara umum kalimat "logika" atau logis merujuk kepada cara seseorang dalam berfikir dengan cara yang wajar serta memiliki alasan dan memiliki hubungan yang rasional serta dapat dimengerti walaupun belum tentu apa yang dilakukannya disetujui sebagai suatu yang benar atau salah.¹

Pengertian Logika merupakan hasil pertimbangan akal pikiran yang diutarakan lewat kata dan dinyatakan dalam bahasa. Kata Logika berasal dari Yunani kuno (logos) yang juga merupakan salah satu cabang ilmu filsafat. Sebagai sebuah ilmu, logika disebut dengan logike episteme (bahasa Latin: *logica scientia*) atau ilmu logika (ilmu pengetahuan) yang mempelajari kecakapan untuk berpikir secara lurus, tepat, dan teratur dan tepat sesuai dengan pertimbangan akal sehat yang sejalan dengan pemikiran yang aktif.²

1.2. Algoritma

Pengertian Algoritma merupakan urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Algoritma yang dapat menyelesaikan suatu permasalahan dalam waktu yang singkat memiliki tingkat kerumitan yang rendah, sementara algoritma yang

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

membutuhkan waktu lama untuk menyelesaikan suatu masalah membutuhkan tingkat kerumitan yang tinggi.³

Defenisi dari algoritma sangat lekat dengan kata logika, yaitu kemampuan seorang manusia untuk berfikir dengan akal tentang suatu permasalahan menghasilkan sebuah kebenaran, dibuktikan dan dapat diterima akal, logika seringkali dihubungkan dengan kecerdasan, seseorang yang mampu berlogika dengan baik sering orang menyebutnya sebagai pribadi yang cerdas. Dalam menyelesaikan suatu masalah pun logika mutlak diperlukan untuk menyelesaikan permasalahan tersebut.²





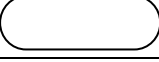


Algoritma ini sebuah buku maha karya hasil dari ciptaan dengan nama Muhammad bin Musa al-Khwarizmi, Dari Al-Khawarizmi kemudian berubah menjadi algorithm dalam Bahasa Inggris dan diterjemahkan menjadi algoritma dalam bahasa Indonesia yang perlu untuk kita ketahui bersama.³

1.3. Flowchart

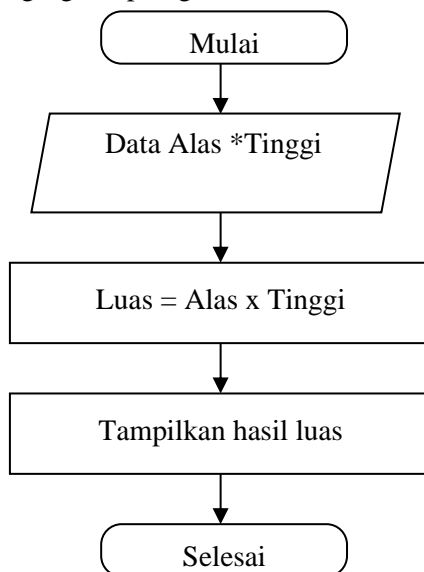
Defenisi dari Flowchart atau bagan alir adalah skema/bagan (*chart*) yang menunjukkan aliran (flow) di dalam suatu program secara logika. Flowchart merupakan alat yang banyak digunakan untuk menggambarkan algoritma dalam bentuk notasi-notasi tertentu. Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan antara proses digambarkan dengan garis penghubung. Dengan menggunakan flowchart akan memudahkan kita untuk melakukan pengecekan bagian yang terlupakan dalam analisis masalah. Di samping itu flowchart juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang akan dikembangkan dan dibuat dengan perancangan sistem terlebih dahulu.³

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

Tabel 1 Simbol dari Flowchart

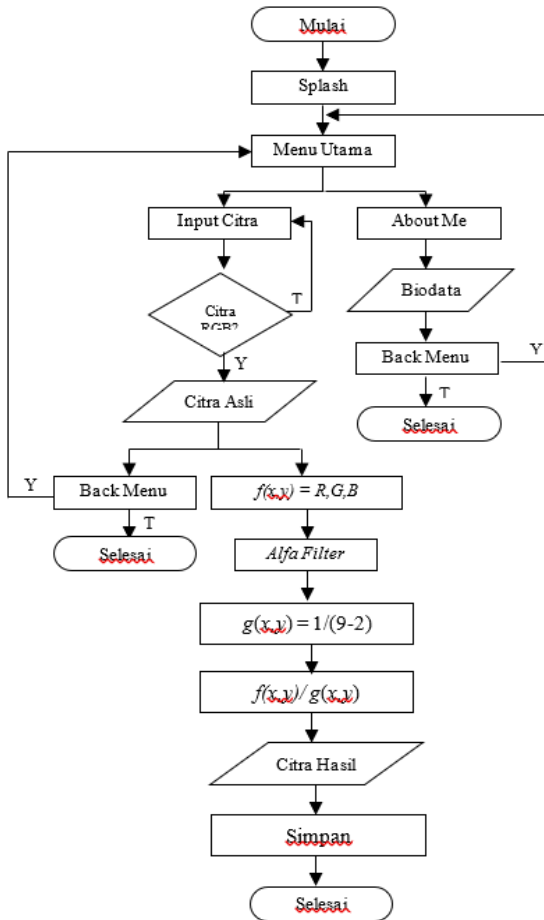
Simbol	Nama	Keterangan
	Proses	Mempresentasikan operasi
	Input/Output Data	Mempresentasikan input data atau output data yang diproses
	Keputusan	Keputusan dalam program
	Preparation	Proses inisialisasi / Proses pemberian nilai awal
	Terminal Points	Awal/Akhir program
	Magnetic Tape	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	Anak Panah	Arah aliran program / alur kerja

Contoh pembuatan flowchart sederhana dengan menghitung perumusan luas segitiga, seperti gambar dibawah ini



Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

Untuk Contoh kasus yang lebih komplrit dapat diambil pada Penelitian yang berjudul Penerapan Metode Alfa Trimmed Mean Filter Dalam Pengolahan Citra Berbasis Android, Adapun gambaran perancangan sistemnya dengan flowchart yaitu sebagai berikut;



1.4. Pseudocode

Defenisi dari Pseudocode adalah cara penulisan algoritma yang menyerupai bahasa pemrograman tingkat tinggi. Pseudocode menggunakan bahasa yang hampir menyerupai bahasa pemrograman. Biasanya pseudo-code menggunakan bahasa yang mudah dipahami secara universal dan juga lebih ringkas dari pada algoritma. Pseudocode

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

berisi deskripsi dari algoritma pemrograman komputer yang menggunakan struktur sederhana dari beberapa bahasa pemrograman tetapi bahasa tersebut hanya di tujukan agar dapat di baca manusia. Sehingga pseudocode tidak dapat dipahami oleh komputer. Supaya notasi pseudocode bisa dipahami oleh komputer maka harus diterjemahkan terlebih dahulu menjadi sintaks bahasa pemrograman seperti bahasa C/C++, Bahasa Pascal, Bahasa PHP dan Bahasa Java.²

Pengertian lain dari Pseudocode merupakan kode yang mirip dengan kode pemrograman yang sebenarnya seperti Pascal, atau C, sehingga lebih tepat digunakan untuk menggambarkan algoritma yang akan dikomunikasikan kepada pemrogram. Sedangkan algoritma disajikan dengan gambar, yaitu dengan alur diagram (Flowchart).³

Contoh Prseudocode untuk menghitung luas segitiga sebagai berikut ini:

1. Mulai
2. Baca alas, tinggi
3. $\text{Luas} = 0.5 * \text{alas} * \text{tinggi}$
4. Cetak Luas
5. Selesai

1.5. Latihan Soal

1. Buatkan Flowchart untuk mencari Luas persegi empat?
2. Buatkan Flowchart untuk mencari Luas Prisma?
3. Buatlah Algoritma dan Flowchart untuk menentukan kelulusan siswa? Dengan ketentuan:
Jika Nilai ≥ 70 maka Lulus
Jika Nilai ≤ 70 maka Tidak Lulus

BAB II

Pengenalan Bahasa C++

Capaian Pembelajaran:

1. Pengantar Bahasa C++
2. Mengetahui format perintah di C++
3. Mengetahui perintah-perintah sederhana di C++
4. Mengetahui cara instalasi Dev-C++
5. Mengetahui cara Compiler Dev-C++

2.1 Pengantar Bahasa C++

Pemrograman bahasa C++ diciptakan oleh Bjarne Stroustrup di laboratorium Bell pada awal tahun 1980-an, sebagai pengembangan dari bahasa C dan Simula. Saat ini, C++ merupakan salah satu bahasa yang paling populer untuk pengembangan software berbasis OOP.⁴

Dengan compiler atau eksekusi program untuk C++ telah banyak beredar di pasaran. Software developer yang paling diminati adalah Borland Inc. dan Microsoft Corp. Produk dari Borland untuk kompilasi C++ adalah Turbo C++, Borland C++, Borland C++ Builder. Sedangkan dari Microsoft adalah Ms. Visual C++ dan lain-lainnya.⁴

Terdapat banyak kompilasi yang tersedia, namun pada intinya bahasa pemrograman yang dipakai adalah C++. Untuk membangun sebuah program dasar memperkenalkan pengeksekusian dan eksekusi program C++.

2.2 Format Perintah dalam C++

Pengenalan perintah-perintah terbaik untuk belajar bahasa pemrograman adalah dengan langsung mempraktikannya. Cobalah program berikut:

```
//header program bahasa C++
#include <iostream>
#include <conio.h>
using namespace std;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
main()
{
//perintah mencetak ke layar
cout<<"Selamat Belajar Bahasa C++ Dasar";
getch();
}
```

Koding program di atas, misalnya dapat disimpan dengan nama `latihan1.cpp`. Cara untuk menyimpan dan mengkompile program berbeda-beda, tergantung kompiler yang dipakai di dalam Dev C++. Sesaat dilakukan Run, maka di layar akan muncul sebuah tulisan **“Selamat Belajar C++ Dasar”**. Contoh di atas, adalah sebuah contoh program sederhana menggunakan C++. Namun, penggalan program tersebut telah menyertakan sintak-sintak dasar bahasa C++.

Sedikit penjelasan mengenai sebuah baris komentar. Semua baris, yang ditandai dengan dua buah tanda slash (`//`), akan dianggap sebagai baris komentar dan tidak akan berpengaruh pada hasil.

2.3 Perintah Sederhana C++

Adapun beberapa perintah yang berupa pada bahasa C++ didalam software Dev C++ sebagai berikut ini:

a. `include`

Perintah `#include <iostream.h>` pernyataan yang diawali dengan tanda (`#`) merupakan pernyataan untuk menyertakan preprocessor. Pernyataan ini bukan untuk dieksekusi. `#include <iostream.h>` berarti memerintahkan kompiler untuk menyertakan file header `iostream.h`.⁴

b. `using namespace std;`

Perintah penulisan kode ini adalah untuk melakukan penulisan kode langsung tanpa penulisan scope `std::`. Hal ini dikarenakan, pada `iostream C++` melakukan pembungkusan semua fungsi input-output dalam namespace `std` (akan dijelaskan kemudian pada topik “namespace”).

c. `int main`

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

Perintah `int main ()` Baris ini menandai dimulainya kompiler akan mengeksekusi program. Atau dengan kata lain, pernyataan `main` sebagai penanda program utama. Adalah suatu keharusan, dimana sebuah program yang ditulis dalam bahasa C++ memiliki sebuah `main`.

d. `main`

Perintah `main` diikuti oleh sebuah tanda kurung `()` karena `main` merupakan sebuah fungsi. Dalam bahasa C++ sebuah fungsi harus diikuti dengan tanda `()`, yang nantinya dapat berisi argumen. Dan sintak formalnya, sebuah fungsi dimulai dengan tanda `{}`, seperti dalam contoh program.

e. `cout`

Perintah ini merupakan hal yang akan dieksekusi oleh compiler dan merupakan perintah yang akan dikerjakan. `cout` termasuk dalam file `iostream`. `cout` merupakan perintah untuk menampilkan ke layar.

f. `cin`

Fungsi **`cin`** (c input) adalah fungsi untuk mengambil input dari keyboard. Fungsi **`cin`** membutuhkan variabel untuk menyimpan data yang diinputkan.

g. `return 0`

Perintah ini menyatakan hasil keluaran dari fungsi `main ()` adalah 0. Pada C++, `return 0` dalam `main()` menyatakan bahwa program berakhir dengan normal.

h. `getch()`

Perintah ini digunakan untuk mendapatkan 1 (satu) karakter dari user. Ingat! Hanya 1 (satu) karakter. Dan, karena karakteristiknya tersebut, seringkali perintah `getch()` hanya digunakan untuk melakukan “penundaan” sebelum perintah berikutnya dieksekusi. Biasanya juga digunakan untuk membuat agar tampilan output tidak langsung tertutup.⁴

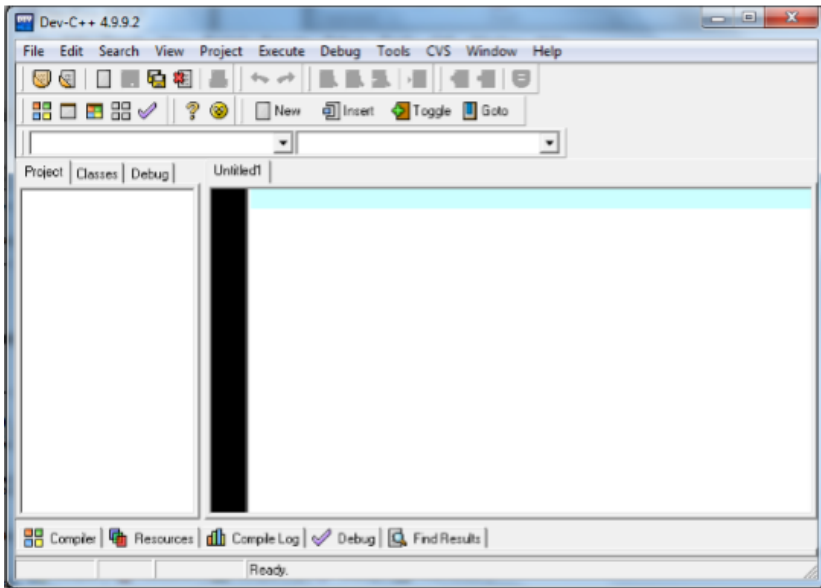
Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

2.4 Instalasi Dev C++

IDE merupakan singkatan dari Integrated Development Environment, merupakan Lembar kerja terpadu untuk pengembangan program. IDE dari Dev C++, dapat digunakan untuk:

- a. Menulis Naskah Program.
- b. Mengkompilasi Program (Compile)
- c. Melakukan Pengujian Program (Debugging)
- d. Mengaitkan Object dan Library ke Program (Linking)
- e. Menjalankan Program (Running)

1) Tampilan Program IDE Dev C++



Gambar 2.1 IDE Dev C++ 4.9.9.2

IDE pada Dev C++, terbagi menjadi 4 (empat) bagian, yaitu:

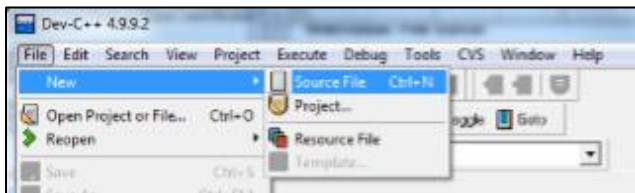
- a. Menu Utama (Menubar) Menu utama terdiri dari: File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Windows dan Help
- b. Jendela Text Edit tempat untuk mengetikan program dan membuat program. Jika pertama kali anda membuat program, nama file jendela editor adalah Untitled1.cpp

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

- c. Jendela Message tempat untuk menampilkan pesan-pesan pada proses kompilasi dan link program.
- d. Baris Status Baris dimana menampilkan keterangan-keterangan pada saat anda mengaktifkan menu bar dan sub menu.

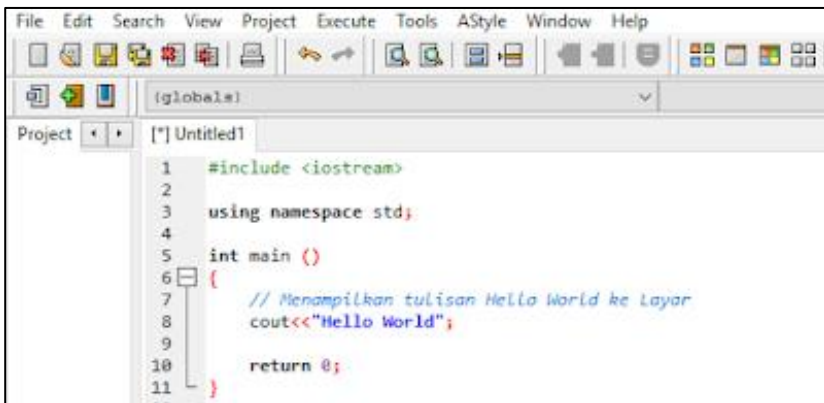
2) Membuat File Editor

File Editor merupakan File Program yang dapat dikompile, dan dijalankan untuk menampilkan hasilnya serta mempunyai ekstensi.cpp. Cara mengaktifkannya: Klik Menu File Gambar 2.2 Klik New, Source File.



Gambar 2.2 Membuat File Baru

Kita dapat mengetik koding program pada ruang kerja software Dev C++ dimana pengetikan / penulisan koding program harus dengan menggunakan bahasa C++, seperti Gambar 2.3 dibawah ini:



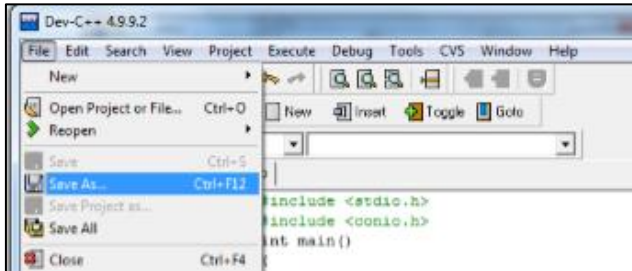
Gambar 2.3 Mengetik Koding Program

3) Menyimpan File Editor

Penjelasan Setelah selesai mengetikan naskah program yang baru pada jendela Text Edit, maka selanjutnya disimpan dengan cara:

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

- a. Kik Menu File Save
- b. Menekan Ctrl + S. Selanjutnya tampil jendela Save File As, seperti gambar dibawah ini:

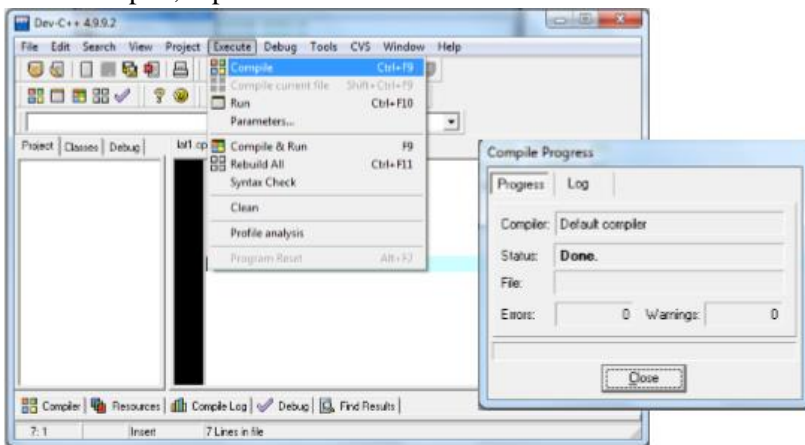


Gambar 2.4 Menyimpan File

4) Compile Program Bahasa C++

Proses Compile merupakan suatu proses menterjemahkan program dari bahasa manusia kedalam bahasa yang dimengerti oleh komputer yaitu bahasa mesin. Caranya adalah:

- a. Kik Menu Project Compile
- b. Menekan HotKey Ctrl + F9 Selanjutnya tampil kotak dialog Compile, seperti dibawah ini:



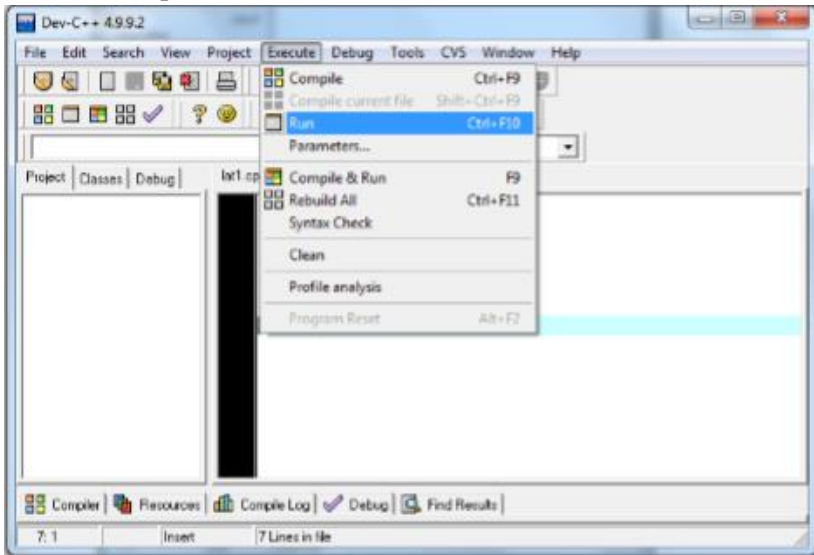
Gambar 2.5 Compile Program Bahasa C++

5) Menjalankan Program

Penjelasan Proses Run merupakan suatu proses menterjemahkan program, melakukan proses linking, membuat file eksekusi (.exe) dan sekaligus menjalankan program. Caranya adalah:

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

- c. Klik Menu Debug Run
- d. Menekan HotKey Ctrl + F10 Selanjutnya tampil kotak dialog Run, seperti dibawah ini:



Gambar 2.6 Menjalankan Program Bahasa C++

Catatan:

Jika program yang dijalankan tidak muncul, untuk melihat hasil compile dapat dijalankan di command prompt.

2.5 Latihan Soal

1. Buatlah Koding Program Luas Permukaan dan Volume Kubus dengan Bahasa C++?
2. Buatlah Koding Program Cetak Pada Layar“Saya Semangat Belajar Program” C++?

BAB III

KONSEP PEMROGRAMAN C++

Capaian Pembelajaran:

1. Pengenalan Tipe Data Variabel dan Konstanta dalam bahasa C++
2. Mampu membuat Algoritma, Flowchart, Pseudocode dan koding program bahasa C++

Variabel, konstanta dan tipe data merupakan tiga hal yang akan selalu kita jumpai ketika kita membuat program. Bahasa pemrograman apapun dari yang paling sederhana sampai yang paling kompleks, mengharuskan kita untuk mengerti ketiga hal tersebut.

3.1 Tipe Data

Definisi Tipe Data berfungsi untuk menentukan nilai apa saja yang dapat di masukkan pada suatu variabel. Sebagai contoh kita memberikan label angka pada umur, maka ketika kita memberikan nilai huruf atau string ke dalam variabel umur akan ditolak.¹

//Contoh program tipe data

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int x;
    float y;
    char z;
    double w;
    x = 10;
    y = 9.45;
    z = 'S';
    w = 3.45E+20;
    cout<<"Nilai x ="<<x<<endl;
    cout<<"Nilai y ="<<y<<endl;
    cout<<"Nilai z ="<<z<<endl;
```


Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"Nilai w="<<w;  
getch();  
}
```

3.2 Variabel

Definisi Variabel adalah tempat dimana kita dapat mengisi atau mengosongkan nilainya dan memanggil kembali apabila dibutuhkan. Setiap variabel akan mempunyai nama (identifier) dan nilai.

//Contoh nama variabel dan nilai.³

```
Nama = "Udin"  
Harga = 2500  
HargaTotal = 34000  
  
//contoh koding program variabe dan nilai bahasa C++  
#include <iostream>  
#include <conio.h>  
using namespace std;  
int main()  
{  
Char nama;  
int harga;  
float harga_total;  
nama="udin";  
harga=2500;  
harga_total=34000;  
cout<<nama<<endl;  
cout<<harga<<endl;  
cout<<harga_total;  
getch();  
}
```

3.3 Konstanta

Definisi Konstanta adalah variabel yang nilai datanya bersifat tetap dan tidak bisa diubah. Jadi konstanta adalah juga variabel bedanya adalah pada nilai yang disimpannya. Jika nilai datanya sepanjang program berjalan tidak berubahubah, maka sebuah varibel lebih baik diperlakukan sebagai konstanta.³

Sebagai contoh, jika kita membuat program perhitungan matematik yang menggunakan nilai PHI (3.14) yang mungkin akan muncul

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

dibanyak tempat pada kode program, kita dapat membuat PHI sebagai konstanta.

//Contoh koding program konstanta di bahasa C++

```
#include <iostream>
using namespace std;
#define PANJANG 8
#define LEBAR 5
int main()
{
    cout << "Panjang = " << PANJANG;
    cout << "Lebar = " << LEBAR;
    cout << "Luas (panjang*lebar) = " <<
PANJANG*LEBAR;
    cout << "\n";
    return 0;
}
```

3.4 Contoh Soal dan penyelesain

a. Algoritma dari Kasus Menghitung Luas Lingkaran.

Penyelesaian Algoritmanya:

1. Mulai
2. Menentukan int jari; Float luas, PHI = 3.14
3. Masukan jari = 3;
4. Menghitung luas = PHI * jari * jari; (3.14 * 3 * 3)
5. Menampilkan Luas Lingkaran = 28.26
6. Selesai

Flowchart Sistem:

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++



Pseudocode:

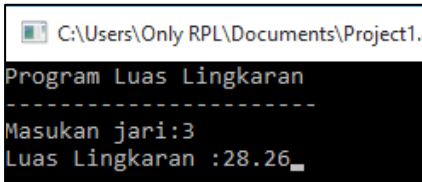
```
Program hitung luas_lingkaran
Var: Int Jari, Float L, PHI=3.14
Input jari = 3
luas ← PHI * Jari*Jari
Print Luas
end
```

Koding Program :

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int jari;
    float luas, PHI=3.14;
    cout<<"Program Luas Lingkaran"<<endl;
    cout<<"-----"<<endl;
    cout<<"Masukan jari:";
    cin>>jari;
    luas = PHI * jari * jari;
    cout<<"Luas Lingkaran:"<<luas;
    getch();
}
```

Hasil Run Program:

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

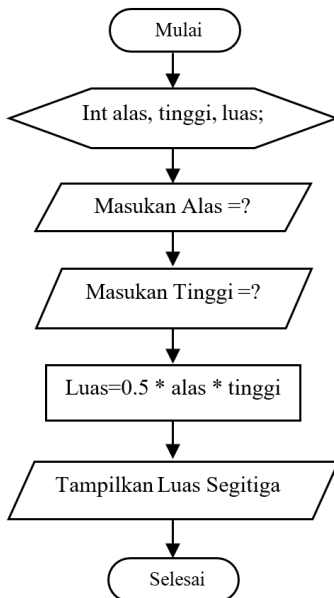


```
C:\Users\Only RPL\Documents\Project1.  
Program Luas Lingkaran  
-----  
Masukan jari:3  
Luas Lingkaran :28.26_
```

b. Algoritma Kasus Menghitung Luas Segitiga

1. Mulai
2. Menentukan int alas, tinggi, luas;
3. Masukan nilai alas = 4 (variabel bebas)
4. Masukan nilai tinggi = 6 (variabel bebas)
5. Menghitung Luas = $0.5 * \text{alas} * \text{tinggi}$;
6. Menampilkan hasil luas segitiga adalah 12
7. Selesai

Flowchart Sistem:



Pseudocode:

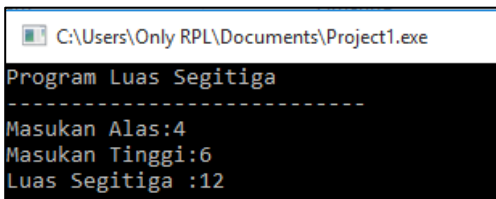
Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
Program hitung luas_segitiga
Var: int alas, tinggi, luas
    input nilai alas
    input nilai tinggi
    Luas ← 0.5 * alas * tinggi
Print Luas
end
```

Koding Program:

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int alas, tinggi, luas;
    cout<<"Program Luas Segitiga"<<endl;
    cout<<"-----"<<endl;
    cout<<"Masukan Alas:";
    cin>>alas;
    cout<<"Masukan Tinggi:";
    cin>>tinggi;
    luas = 0.5 * alas * tinggi;
    cout<<"Luas Segitiga:"<<luas;
    getch();
}
```

Hasil Run Program



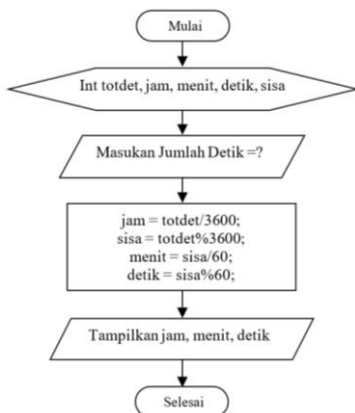
```
C:\Users\Only RPL\Documents\Project1.exe
Program Luas Segitiga
-----
Masukan Alas:4
Masukan Tinggi:6
Luas Segitiga :12
```

c. Algoritma Menghitung Konversi Detik Menjadi Jam Menit Detik

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

1. Mulai
2. Menentukan int (total detik) totdet, jam, menit, detik, sisa
3. Masukan jumlah detik
4. Menghitung jumlah jam = (Rumus: jumlah detik / 3600)
5. Menghitung sisa = (Rumus: jumlah detik % 3600)
6. Menghitung Menit= (Rumus: jumlah sisa / 60)
7. Menghitung Detik = (Rumus: jumlah sisa % 60)
8. Menampilkan Hasil Jumlah jam, Menit, dan Detik
9. Selesai

Flowchart sistem:



Pseudocode:

```
Program hitung
konversi_Detik_menjadi_jam_menit_detik
Var: int totdet, sisa, jam, menit, detik
    input detik
    Jam ← totdet/3600
    Sisa ← totdet%3600
    Menit ← sisa/60
    Detik ← sisa%60
Print jam, menit, detik
end
```

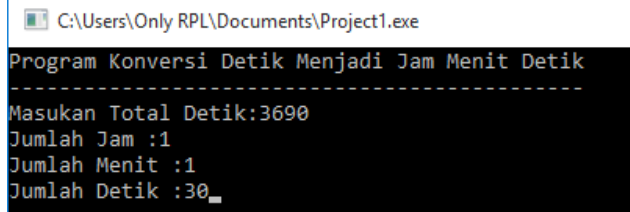
Koding Program:

```
#include <iostream>
```


Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
#include <conio.h>
using namespace std;
int main()
{
    int totdet, jam, sisa, menit, detik;
    cout<<"Program Konversi Detik Menjadi Jam Menit
Detik"<<endl;
    cout<<"-----"<<endl;
    cout<<"Masukan Total Detik:";
    cin>>totdet;
    jam = totdet / 3600;
    sisa = totdet % 3600;
    menit = sisa / 60;
    detik = sisa % 60;
    cout<<"Jumlah Jam:"<<jam<<endl;
    cout<<"Jumlah Menit:"<<menit<<endl;
    cout<<"Jumlah Detik:"<<detik;
    getch();
}
```

Hasil Run Program:



```
C:\Users\Only RPL\Documents\Project1.exe
Program Konversi Detik Menjadi Jam Menit Detik
-----
Masukan Total Detik:3690
Jumlah Jam :1
Jumlah Menit :1
Jumlah Detik :30_
```

d. Algoritma Fungsi IF ELSE

Pengambilan keputusan (kondisi if) digunakan untuk mengantisipasi kondisi yang terjadi saat jalannya program dan menentukan tindakan apa yang akan diambil sesuai dengan kondisi. Ada beberapa statement/kondisi diantaranya adalah if, else dan elseif. Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar True.

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

Diketahui sebuah IF (Percabangan) pada sebuah kasus pembelian barang, jika harga dibeli minimal total Rp.50.000 keatas maka akan mendapatkan diskon 20%, jika harga dibeli dibawah harga Rp.50.000 maka akan mendapatkan diskon 10%.

Jika Harga beli = 50000

Diskon = 20 %

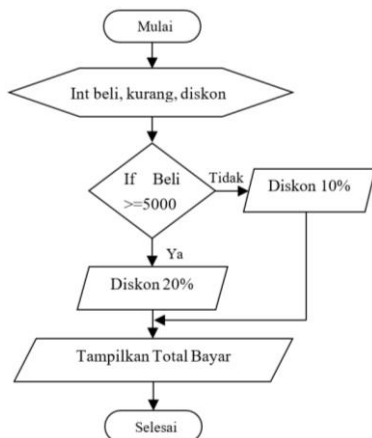
Total bayar = $50000 \times 0.2 = 40000$

ika Harga Beli = 40000

Diskon = 10%

Total Bayar = $40000 \times 0.1 = 36000$

Flowchart Sistem:



Pseudocode:

```
Program IF_pembelian barang
Var: int beli, kurang, diskon
Data beli
    IF beli = 50000
        Diskon 20%
    Else
        Diskon 10%
Print total bayar
End
```

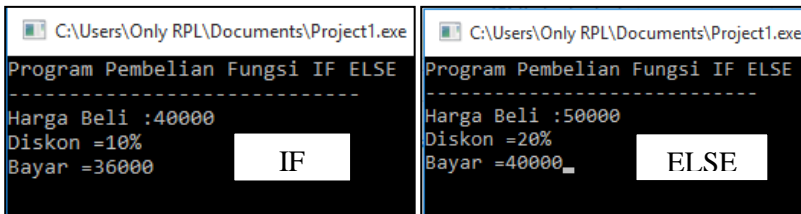
Koding Program:

```
#include <iostream>
#include <conio.h>
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
using namespace std;
main()
{
    int beli, kurang, diskon;
    cout<<"Program Pembelian Fungsi IF ELSE"<<endl;
    cout<<"-----"<<endl;
    cout<<"Harga Beli:";
    cin>>beli;
    if(beli>=50000)
    {
        diskon=beli * 0.2;
        kurang= beli - diskon;
        cout<<"Diskon =20%"<<endl;
        cout<<"Bayar ="<<kurang;
    }else{
        diskon=beli * 0.1;
        kurang= beli - diskon;
        cout<<"Diskon =10%"<<endl;
        cout<<"Bayar ="<<kurang;
    }
    getch();
}
```

Hasil program:



e. Algoritma Fungsi FOR

Diketahui nilai perulangan (Loop) menggunakan fungsi perulangan seperti FOR dan WHILE untuk mengulang sebuah pernyataan (Statement) yang akan dibuat seperti program dibawah ini:

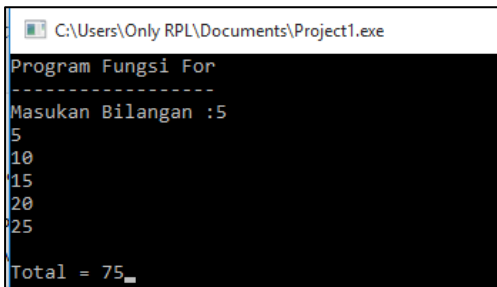
Koding Program:

```
#include <iostream>
#include <conio.h>
using namespace std;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
main()
{
int x, y, z, total=0;
cout<<"Program Fungsi For"<<endl;
cout<<"-----"<<endl;
cout<<"Masukan Bilangan:";
cin>>x;
for (y=5; y<=5; y++)
for (z=5; z<=x*5; z=z+5)
{
    cout<<z<<endl;
    total=total+z;
}
cout<<endl;
cout<<"Total = "<<total;
getch();
}
```

Hasil Run Program:



```
C:\Users\Only RPL\Documents\Project1.exe
Program Fungsi For
-----
Masukan Bilangan :5
5
10
15
20
25
Total = 75_
```

3.5 Latihan Soal

1. Buatlah koding program bahasa C++ mengenai menghitung Luas Persegi?
2. Buatlah koding program bahasa C++ mengenai menghitung Luas dan Volume Bola?

BAB IV

Struktur Data (ARRAY)

Capaian Pembelajaran:

1. Memahami Array (Struktur Data)
2. Memahami Proses Tumpukan Sorting, Insertion Sort, Selection Sort, Merge Sort, Rekursif dan Quick Sort, Searching, Stack (Tumpukan), Queue,
3. Mampu Membuat Program Tumpukan Dalam Bahasa C++

4.1 Array

Sebuah Array atau larik adalah suatu peubah hanya dapat menyimpan sebuah nilai saja dan tidak dapat menyimpan beberapa nilai yang bertipe sejenis. sedangkan dalam pemrograman terkadang kita sering mengolah kumpulan data yang memiliki tipe data yang sama. Sehingga diperlukarr sebuah variabel yang dapat menampung kumpulan data dengan tipe data yang sama. oleh karena itu, untuk mengatasi permasalahan pemrograman tersebut diperlukan sebuah larik atau Array untuk menampung data tersebut. Suatu lokasi memori untuk suatu array tidak dapat ditambah atau dikurangi selama program dijalankan. untuk mengubah ukuran dari lokasi memori suatu array haruslah cliperbaiki dalam listirtg programnva, karena array termasuk ke dalam struktur data statis.¹

Adapun Array adalah sebuah struktur data yang terdiri atas banyak variabel dengan tipe data sama, dimana masing-masing elemen variabel mempunyai nilai indeks.⁵

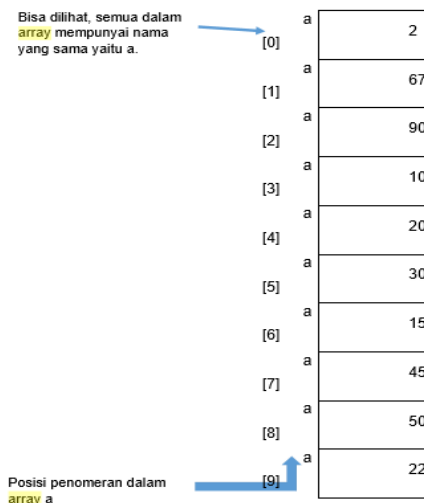
Data Array merupakan sekelompok lokasi memori yang berdekatan yang semuanya memiliki tipe data yang sama. Untuk merujuk pada suatu lokasi atau elemen tertentu dalam array, kita bisa menentukan nama, jumlah elemen dan posisi array tertentu didalamnya. Ada beberapa sifat yang dimiliki oleh sebuah array:

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

1. Sekelompok lokasi memori (data) yang mempunyai tipe data yang sama dan menggunakan nama yang sama.
2. Sekelompok variabel dapat menggunakan nama yang sama
3. Perbedaan antara variabel dengan variable yang lain adalah apa yang terkandung didalam subscript.
4. Subscript adalah bilangan yang terdapat di dalam kurung kotak []
5. Cara mengakses sebuah array adalah dengan memanggil variable yang ada dalam subscript.

4.2 Array Satu Dimensi

Bahwa elemen diatas mempunyai susunan tertentu. Susunan tersebut bisa berupa satu dimensi, dua dimensi bahkan N dimensi. Tentu susunan yang termudah adalah satu dimensi atau berdimensi satu, sehingga biasa disebut array satu dimensi atau array dimensi satu (one dimension array). Gambar dibawah ini menunjukkan bahwa array tersebut mempunyai nama a, yang mengandung 10 elemen. Elemen – elemen ini diberikan nama berdasarkan letak atau posisi nomor elemen yang ditandai dengan kurung kotak ([]). Array pertama didalam urutan atau tata letak, selalu dimulai dari urutan ke Nol (0). Dalam pemberian nama pada sebuah array bias.⁵



Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

Array satu dimensi atau biasa disebut vector (karena mempunyai satu arah) dapat digunakan untuk berbagai macam keperluan.

4.3 Deklarasi Array

Agar mendeklarasikan sebuah array satu dimensi dalam sebuah Bahasa pemrograman biasanya digunakan tanda [] (Kurung Kotak) / Bracket yang berfungsi untuk menyatakan berapa banyak jumlah elemen yang akan terbentuk pada sebuah array tersebut. Adapun bentuk umum dari pendeklarasian array satu dimensi:

<code>Tipe_data <spasi> Nama_array [Jumlah_Elemen];</code>
<code>Int Nilai[5]</code>

Pada contoh yang diberikan diatas dinyatakan bahwa array tersebut mempunyai:

Tipe data Array : Integer

Nama Array : Nilai

Jumlah elemen Array : 5

4.4 Menyiapkan Array satu dimensi Bahasa pemrograman C++

Cara menyiapkan array satu dimensi adalah dengan menyebut tipe_data, Nama_array dan jumlah_elemen array seperti contoh dibawah ini.

a. Sorting

Sorting adalah metode mengurutkan data baik secara menurun maupun naik dari data yang dimasukkan secara acak. Dalam metode sorting sendiri ada beberapa cara yaitu dengan bubble sort, Insertion sort, Selection sort, dan shell sort.

Koding Program Bahasa C++:

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
```

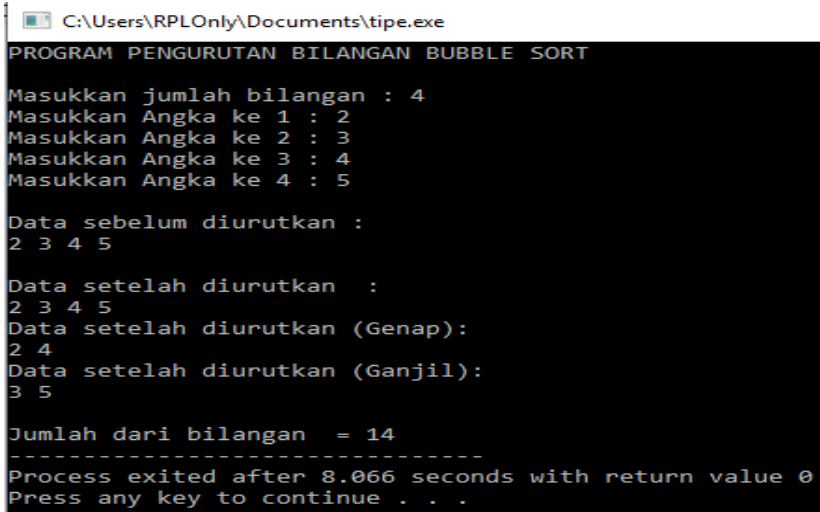
Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
int data[10];
int i, j, k, tmp, jumlah=0;
cout<<"PROGRAM    PENGURUTAN    BILANGAN    BUBBLE
SORT\n\n";
cout<<"Masukkan jumlah bilangan: "; cin>>k;
for(i=0; i<k; i++) {
cout<<"Masukkan Angka ke "<<(i+1)<<": ";
jumlah=jumlah+data[i];
}
cout<<"\n Data sebelum diurutkan: "<<endl;
{
cout<<data[i]<<" ";
}
cout<<endl;
for( i=0;i<k;i++)//Disini Proses Bubble Sort
{
for(j=i+1;j<k;j++)
{
if(data[i]>data[j])
{
tmp=data[i];
data[i]=data[j]; data[j]=tmp;
}      }      }
cout<<"\nData setelah diurutkan: "<<endl;
for(i=0; i<k; i++)
{ {
cout<<data[i]<<" ";
}      }
cout<<"\n Data setelah diurutkan (Genap): "<<endl;
for(i=0; i<k; i++)
{
if (data[i]%2==0)
{
```


Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<data[i]<<" ";
}      }
cout<<"\nData setelah diurutkan (Ganjil): "<<endl;
for(i=0; i<k; i++)
{
if (data[i]%2!=0)
{
cout<<data[i]<<" ";
}
}
cout<<"\n\nJumlah dari bilangan = "<<jumlah;
return 0;
}
```

Hasil Run Program:



```
C:\Users\RPLOnly\Documents\tipe.exe
PROGRAM PENGURUTAN BILANGAN BUBBLE SORT

Masukkan jumlah bilangan : 4
Masukkan Angka ke 1 : 2
Masukkan Angka ke 2 : 3
Masukkan Angka ke 3 : 4
Masukkan Angka ke 4 : 5

Data sebelum diurutkan :
2 3 4 5

Data setelah diurutkan :
2 3 4 5
Data setelah diurutkan (Genap):
2 4
Data setelah diurutkan (Ganjil):
3 5

Jumlah dari bilangan = 14
-----
Process exited after 8.066 seconds with return value 0
Press any key to continue . . .
```

b. Insertion Sort

Data sortasi penyisipan merupakan algoritma penyortiran sederhana yang membangun array terakhir yang diurutkan satu per satu. Ini jauh lebih efisien pada daftar besar daripada algoritma yang lebih maju

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

seperti quicksort, heapsort, atau semacam penggabungan. Namun, jenis penyisipan memberikan beberapa keuntungan:

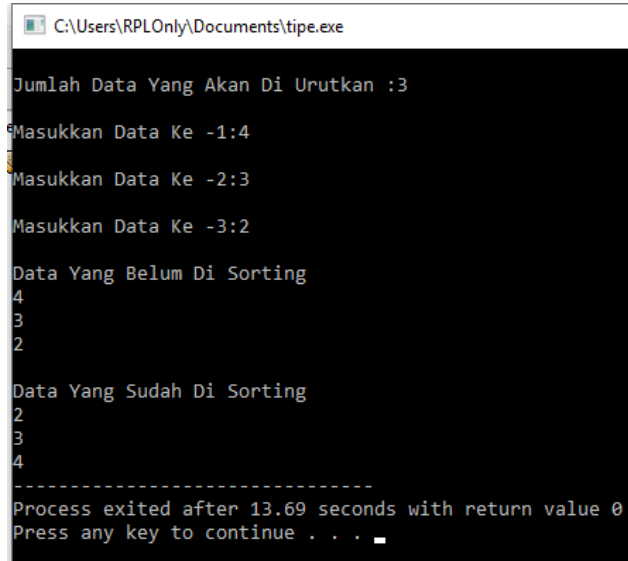
Koding Program:

```
#include <conio.h>
#include <iostream>
#include <stdio.h>
using namespace std;
main()
{
int i,j,n,temp,data[10];
cout<<"\nJumlah Data Yang Akan Di Urutkan:";
cin>>n;
for(i=1;i<=n;i++){
cout<<"\nMasukkan Data Ke -"<<i<<":";
cin>>data[i];
}
cout<<"\nData Yang Belum Di Sorting \n";
for(i=1;i<=n;i++){
cout<<data[i]<<"\n";
}
cout<<"\nData Yang Sudah Di Sorting ";
for(i=1;i<=n;i++){
for(j=1;j<=n;j++){
if(data[i]<data[j]){
temp=data[i];
data[i]=data[j];
data[j]=temp;
}
}
}
for(i=1;i<=n;i++){
cout<<"\n"<<data[i];
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
}  
return 0;  
}
```

Hasil Run Program:



```
C:\Users\RPLOnly\Documents\tpi.exe  
Jumlah Data Yang Akan Di Urutkan :3  
Masukkan Data Ke -1:4  
Masukkan Data Ke -2:3  
Masukkan Data Ke -3:2  
  
Data Yang Belum Di Sorting  
4  
3  
2  
  
Data Yang Sudah Di Sorting  
2  
3  
4  
-----  
Process exited after 13.69 seconds with return value 0  
Press any key to continue . . .
```

c. Selection sort

Algoritma pengurutan perbandingan di tempat. Ini memiliki kompleksitas waktu $O(n^2)$, yang membuatnya tidak efisien pada daftar besar, dan umumnya berkinerja lebih buruk daripada jenis penyisipan yang sama.

Koding Program:

```
#include <iostream>  
#include <conio.h>  
#include <stdio.h>  
using namespace std;  
main()  
{  
    int x[10];  
    int i,bd,temp,minindex,j;  
    cout<<" >> PROGRAM SELECTION SORT << \n" <<endl;  
    cout<<"\nMasukkan Banyak Data [MAKS 10]:"; cin>>bd;  
    for(i=1; i<=bd; i++)  
    {
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"Masukkan Nilai Ke"<<i<<" = ";cin>>x[i];
}
cout<<"\nData sebelum di sort:"; for(i=1; i<=bd;i++)
{
cout<<"\n"<<x[i];
}
for(i=1; i<=bd-1; i++)
{
minindex=i;
for(j=i+1; j<=bd; j++)
{
if(x[minindex]>x[j])
{
minindex=j;
} }
temp=x[i]; x[i]=x[minindex]; x[minindex]=temp;
}
cout<<"\nData setelah di sort:"; for(i=1; i<=bd; i++)
{
cout<<"\n"<<x[i];
}
getch();
}
```

Hasil Run Program:



```
C:\Users\RPLOnly\Documents\tipe.exe
>> PROGRAM SELECTION SORT <<

Masukkan Banyak Data [MAKS 10] :5
Masukkan Nilai Ke1 = 5
Masukkan Nilai Ke2 = 4
Masukkan Nilai Ke3 = 3
Masukkan Nilai Ke4 = 2
Masukkan Nilai Ke5 = 6

Data sebelum di sort :
5
4
3
2
6
Data setelah di sort :
2
3
4
5
6
```

d. Merge Sort

Merge sort merupakan algoritma pengurutan dalam ilmu komputer yang dirancang untuk memenuhi kebutuhan pengurutan atas suatu rangkaian data yang tidak memungkinkan untuk ditampung dalam

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

memori komputer karena jumlahnya yang terlalu besar. Algoritma ini ditemukan oleh John von Neumann pada tahun 1945.

Prinsip utama yang diimplementasikan pada algoritma *merge-sort* seringkali disebut sebagai *pecah-belah dan taklukkan* (*divide and conquer*). Cara kerja algoritma merge sort adalah membagi larik data yang diberikan menjadi dua bagian yang lebih kecil. Kedua larik yang baru tersebut kemudian akan diurutkan secara terpisah. Setelah kedua buah list tersusun, maka akan dibentuk larik baru sebagai hasil penggabungan dari dua buah larik sebelumnya.

Koding Program:

```
#include <stdio.h>
#define MAX 10 int Data[MAX];
int temp[MAX];
void merge(int Data[], int temp[], int kiri, int
tengah, int kanan)
{
int i, left_end, num_elements, tmp_pos; left_end =
tengah - 1;
tmp_pos = kiri;
num_elements = kanan - kiri + 1;
while ((kiri <= left_end) && (tengah <= kanan))
{
if (Data[kiri] <= Data[tengah])
{
temp[tmp_pos] = Data[kiri]; tmp_pos = tmp_pos + 1;
kiri = kiri + 1;
} else {
temp[tmp_pos] = Data[tengah]; tmp_pos = tmp_pos + 1;
tengah = tengah + 1;
}
}
while (kiri <= left_end)
{
temp[tmp_pos] = Data[kiri]; kiri = kiri + 1;
tmp_pos = tmp_pos + 1;
}
}
```

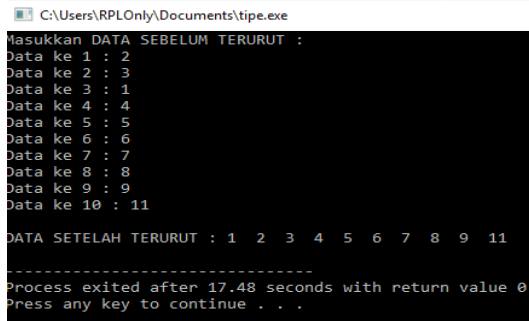
Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
while (tengah <= kanan)
{
temp[tmp_pos] = Data[tengah]; tengah = tengah + 1;
tmp_pos = tmp_pos + 1;
}
for (i=0; i <= num_elements; i++)
{
Data[kanan] = temp[kanan]; kanan = kanan - 1;
}
}
// Prosedur membuat kumpulan data
void m_sort(int Data[], int temp[], int kiri, int
kanan)
{
int tengah;
if (kanan > kiri)
{
tengah = (kanan + kiri) / 2; m_sort(Data, temp, kiri,
tengah); m_sort(Data, temp, tengah+1, kanan);
merge(Data, temp, kiri, tengah+1, kanan);
}
}
void mergeSort(int Data[], int temp[], int
array_size)
{
m_sort(Data, temp, 0, array_size - 1);
}
int main()
{
int i;
printf("Masukkan DATA SEBELUM TERURUT: \n");
for (i = 0; i < MAX; i++)
{
printf ("Data ke %i: ", i+1); scanf ("%d", &Data[i]);
}
mergeSort(Data, temp, MAX); printf("\nDATA SETELAH
TERURUT: ");
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
for (i = 0; i < MAX; i++) printf("%d ", Data[i]);  
printf("\n");  
//scanf("%d"); return(0);  
}
```

Hasil Program:



```
C:\Users\RPLOnly\Documents\type.exe  
Masukkan DATA SEBELUM TERURUT :  
Data ke 1 : 2  
Data ke 2 : 3  
Data ke 3 : 1  
Data ke 4 : 4  
Data ke 5 : 5  
Data ke 6 : 6  
Data ke 7 : 7  
Data ke 8 : 8  
Data ke 9 : 9  
Data ke 10 : 11  
  
DATA SETELAH TERURUT : 1 2 3 4 5 6 7 8 9 11  
  
-----  
Process exited after 17.48 seconds with return value 0  
Press any key to continue . . .
```

Contoh Koding Marge Sort Kedua:

Koding Program:

Contoh Kode Program Kedua:

```
#include <iostream>  
// Merge Sort  
using namespace std;  
void merge(int low, int mid, int up); void  
mergeSort(int low, int up);  
int a[50]; int main()  
{  
    int jumlahBil,i;  
    cout<<"Masukkan Jumlah element Array"<< endl;  
    cin>>jumlahBil;  
    for(int i=0; i<jumlahBil;i++)  
    {  
        cout<<"Bilangan ke:"<< i+1 << endl;  
        cin>>a[i];  
    }  
    mergeSort(1,jumlahBil);  
    for(i=1;i<=jumlahBil;i++)    cout<<a[i]<<"    ";  
    cout<<endl;
```

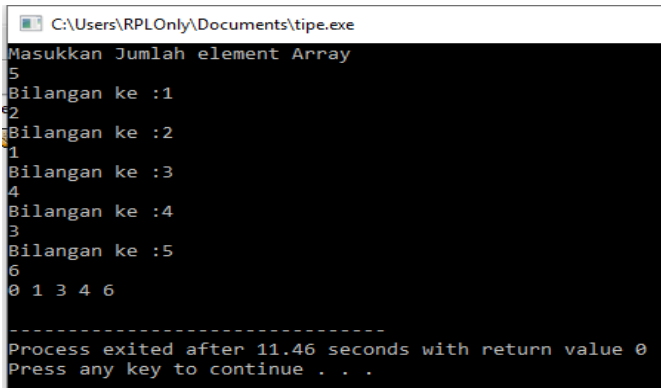
Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
return 0;
}
void merge(int low, int mid, int up)
{
    int h, i, j, k;
    int b[50]; h = low; i = low;
    j = mid+1; while((h<=mid)&&(j<=up))
    {
        if(a[h] < a[j])
        {
            b[i]=a[h]; h++;
        }
        else
        {
            b[i]=a[j]; j++;
        } i++;
    }
    if(h>mid)
    {
        for(k=j; k<=up; k++) { b[i]=a[k];
            i++;
        }
    }
    else
    {
        for(k=h; k<=mid; k++)
        {
            b[i]=a[k]; i++;
        }
    }
    for(k=low; k<=up; k++) a[k]=b[k];
}
void mergeSort(int low, int up)
{
```


Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
int mid; if (low < up)
{
mid = (low + up) / 2;
mergeSort (low, mid);           mergeSort (mid + 1, up);
merge (low, mid, up);
}
}
```

Hasil Run Program:



```
C:\Users\RPL Only\Documents\tipe.exe
Masukkan Jumlah element Array
5
Bilangan ke :1
5
Bilangan ke :2
2
Bilangan ke :3
1
Bilangan ke :4
4
Bilangan ke :5
3
0 1 3 4 6
-----
Process exited after 11.46 seconds with return value 0
Press any key to continue . . .
```

e. Rekursif dan Quick Sort

Algoritma metode quick sort c++ diperkenalkan pertama kali oleh C.A.R. Hoare pada tahun 1960. Program quick sort c++ adalah algoritma sorting yang berdasarkan perbandingan dengan metode divide and conquer (bagi dan kuasai). Disebut metode Quick Sort, karena Algoritma quick sort mengurutkan dengan sangat cepat. metode Quick sort c++ disebut juga dengan partition exchange sort, karena konsepnya membuat partisi-partisi, dan sorting dilakukan per partisi.

Kode program:

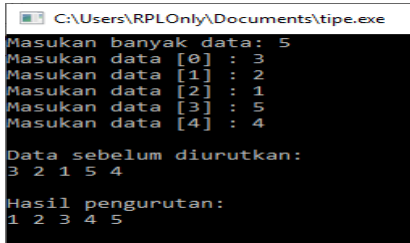
```
#include <iostream>
#include <conio.h>
using namespace std;
void quick_sort(int arr[], int left, int right)
{
int i = left, j = right; int tmp;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
int pivot = arr[(left+right)/2]
while (i<j)
{
while (arr[i] < pivot) i++;
while (arr[j] > pivot) j--;
if (i<=j){ tmp = arr[i];
arr[i] = arr[j]; arr[j] = tmp; i++;j--;
};
}; /* recursion */ if (left < j)
quick_sort(arr, left, j); if (i < right)
quick_sort(arr, i, right);
}
int main()
{
int i,n,data[50];
cout<<"Masukan banyak data: ";
cin>>n;
for(i=0;i<n;i++)
{cout<<"Masukan data ["<<i<<"] : ";
cin>>data[i];}
cout<<"\nData sebelum diurutkan: "<<endl;
for(i=0;i<n;i++)
{
cout<<data[i]<<" ";
}cout<<"\n";
quick_sort(data,0,n-1);
cout<<"\nHasil pengurutan:\n";
{
int i;
for (i=0;i<n;i++)
cout<<data[i]<<" ";
cout<<"\n";
}getch();
}
```

Hasil Run Program:

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++



```
C:\Users\RPLOnly\Documents\tipe.exe
Masukan banyak data: 5
Masukan data [0] : 3
Masukan data [1] : 2
Masukan data [2] : 1
Masukan data [3] : 5
Masukan data [4] : 4

Data sebelum diurutkan:
3 2 1 5 4

Hasil pengurutan:
1 2 3 4 5
```

Koding Program :

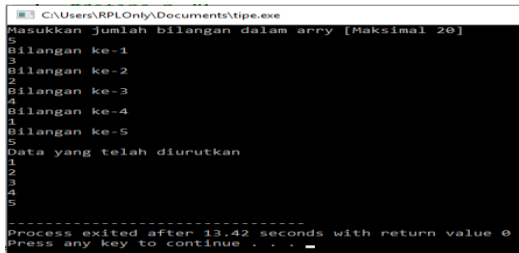
```
#include <iostream>
#define n 20
using namespace std;
int Ar[n];
void quickSort(int arr[], int left, int right);
int main()
{
    int jumlahBil=5;
    cout<<"Masukkan jumlah bilangan dalam array
[Maksimal 20]"<<endl; cin>>jumlahBil;
    int Ar[jumlahBil];
    for(int i=0; i<jumlahBil;i++)
    {
        cout<<"Bilangan ke-"<< i+1 << endl; cin>>Ar[i];
    }
    quickSort(Ar,0,jumlahBil-1 );
    cout<<"Data yang telah diurutkan"<<endl;
    for(int i=0; i<jumlahBil;i++)
    {
        cout<<Ar[i]<<"\n";
    }
}

void quickSort(int arr[], int left, int right)
{
    int i = left, j = right; int tmp;
    int pivot = arr[(left + right) / 2]; while (i <=
j) {
    while (arr[i] < pivot) i++;
    while (arr[j] > pivot) j--;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
    if (i <= j) { tmp = arr[i]; arr[i] = arr[j];  
arr[j] = tmp; i++;  
    j--;  
    }  
};  
    if (left < j) quickSort(arr, left, j); if (i <  
right) quickSort(arr, i, right);  
}
```

Hasil Run Program:



```
C:\Users\RPLOnly\Documents\type.exe  
Masukkan jumlah bilangan dalam array [Maksimal 20]  
5  
Bilangan ke-1  
3  
Bilangan ke-2  
2  
Bilangan ke-3  
4  
Bilangan ke-4  
1  
Bilangan ke-5  
5  
Data yang telah diurutkan  
1  
2  
3  
4  
5  
-----  
Process exited after 13.42 seconds with return value 0  
Press any key to continue . . .
```

f. Searching

Searching adalah proses pencarian suatu elemen di dalam array, ada dua macam tipe pencarian yaitu: pencarian sequential (sequential search) dan pencarian biner (binary search). Perbedaanannya terletak pada keadaan suatu elemen atau data yang berada pada array. Pencarian sequential digunakan apabila data dalam keadaan acak atau tidak urut, sedangkan pencarian biner digunakan pada data yang sudah dalam keadaan urut.

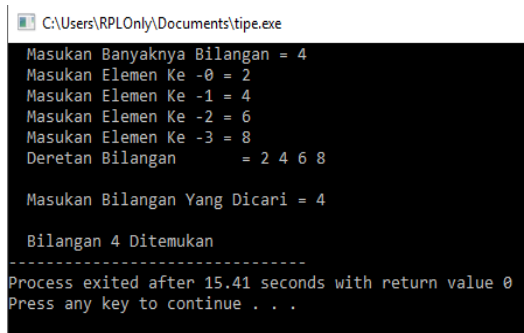
Koding Program:

```
#include<iostream>  
#include<conio.h>  
using namespace std;  
main() {  
    int Nilai[20];  
    int i,N,angka,bilangan;  
    cout<<" Masukkan Banyaknya Bilangan = ";  
    cin>>N;  
    //membaca elemen array
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
for(i=0;i<N;i++){
cout<<" Masukan Elemen Ke -"<<i<<" = ";
cin>>Nilai[i];
}
//mencetak elemen array
cout<<" Deretan Bilangan= ";
for(i=0;i<N;i++)
cout<<Nilai[i]<<" ";
cout<<endl<<endl;
cout<<" Masukan Bilangan Yang Dicari = ";
cin>>bilangan;
cout<<endl;
//melakukan pencarian i=0;
do{
if(Nilai[i]==bilangan) angka=Nilai[i];
i++;
}
while(i<N);
if(angka==bilangan)
cout<<" Bilangan "<<bilangan<<" Ditemukan";
else
cout<<" Bilangan "<<bilangan<<" Tidak Ditemukan";
getch();
}
```

Hasil Run Program:



```
C:\Users\RPLOnly\Documents\tipe.exe
Masukan Banyaknya Bilangan = 4
Masukan Elemen Ke -0 = 2
Masukan Elemen Ke -1 = 4
Masukan Elemen Ke -2 = 6
Masukan Elemen Ke -3 = 8
Deretan Bilangan      = 2 4 6 8

Masukan Bilangan Yang Dicari = 4

Bilangan 4 Ditemukan
-----
Process exited after 15.41 seconds with return value 0
Press any key to continue . . .
```

Koding Program Searching Kedua:

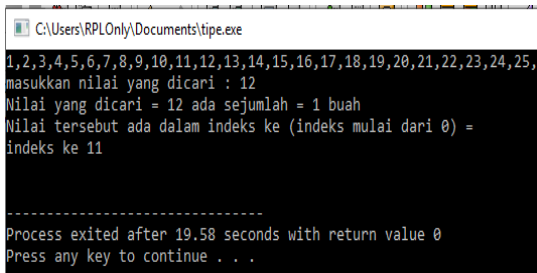
```
#include <iostream>
#include <conio.h>
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
using namespace std;
main ()
{
    Int
    z[]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,1
    9,20,21,22,23,24,25};
    int nilai,index[25],j; j=0;
    for (int i=0;i<25;i++)
    {
        cout<<z[i]<<" ";
    }
    cout<<endl;
    cout<<"masukkan nilai yang dicari: ";
    cin>>nilai;
    for (int i=0;i<=25;i++)
    {
        if (z[i]==nilai)
        {
            index[j]=i; j++;
        }
    }
    if (j>0)
    {
        cout<<"Nilai yang dicari = "<<nilai<<" ada sejumlah
        = "<<j<<" buah"<<endl; cout<<"Nilai tersebut ada
        dalam indeks ke (indeks mulai dari 0) = "<<endl;
        for (int i=0;i<j;i++)
        {
            cout<<"indeks ke "<<index[i]<<endl;
        }
        cout<<endl;
    }
    else
    {
        cout<<"Nilai tidak ditemukan dalam array"<<endl;
    }
    getch();
}
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

Hasil Run Program:



```
C:\Users\RPLOnly\Documents\type.exe
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,
masukkan nilai yang dicari : 12
Nilai yang dicari = 12 ada sejumlah = 1 buah
Nilai tersebut ada dalam indeks ke (indeks mulai dari 0) =
indeks ke 11

-----
Process exited after 19.58 seconds with return value 0
Press any key to continue . . .
```

g. Stack (Tumpukan)

Stack adalah suatu tumpukan dari benda. Konsep utamanya adalah LIFO (Last In First Out), benda yang terakhir masuk dalam stack akan menjadi benda pertama yang dikeluarkan dari stack. Pada gambar di bawah ini bisa kita lihat jika kita ingin mengambil suatu data dari tumpukan maka kita harus mengambil data paling atas dahulu kemudian selanjutnya data yang berada di bawahnya dan seterusnya sampai data yang paling bawah yang pertamakali dimasukan.

Koding Program:

```
#include <iostream>
#include <stdlib.h>
#include<cstring>
#include<windows.h>
using namespace std;
struct tumpukan
{
int top;
char tmp[10][10];
}
tumpukan;
void push (char data[10]); void pop();
void awal(); int isEmpty();
int isFull(); int main()
{
int pilih;
char data[10]; string i;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
awal();
do
{
system("cls");
cout<<"===== "<<endl;
cout<<"= CONTOH PROGRAM TUMPUKAN (STACK) ="<<endl;
cout<<"===== "<<endl<<
endl;
cout<<"====Program Tumpukan Kijang==== "<<endl;
cout<<" 1. PUSH    ="<<endl;
cout<<" 2. POP     ="<<endl;
cout<<" 3. EXIT    ="<<endl;
cout<<"===== "<<endl;
if (!isEmpty())
{
for(int i=tumpukan.top; i>=0; i--)
{
cout<<"["<<tumpukan.tmp[i]<<"] "<<endl;
}
}
else
{
cout<<"[tumpukan kijang dalam kondisi kosong]";
}
cout<<"\n Masukan Pilihan:"<<cin>>pilih;
switch (pilih)
{
case 1:
cout<<"Masukan Kijang kedalam tumpukan:"<<
cin>>data;
push(data);
break; case 2:
pop(); break;
case 3:
cout<<"Tekan enter untuk keluar:"<<
break;
default: cout<<"ERROR!";
```

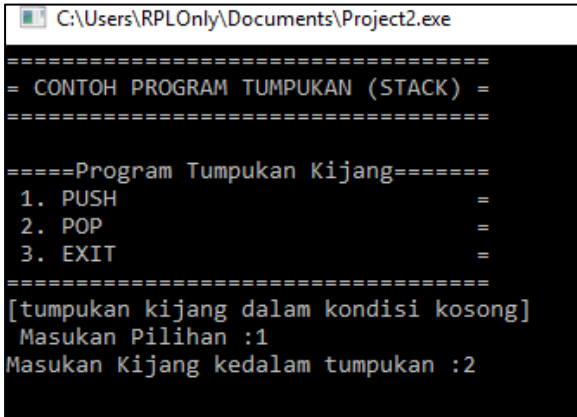

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
break;
}
}
while (pilih!=3);
}
void push (char data[10])
{
if (!isFull())
{
tumpukan.top=tumpukan.top+1;
strcpy(tumpukan.tmp[tumpukan.top],data);
}
else
cout<<"Kijang dalam tumpukan penuh";
}
void pop()
{
if (!isEmpty())
{
tumpukan.top--;
cout<<"kijang pada tumpukan ke-
"<<tumpukan.top+2<<"sudahdiambil";
}
else
cout<<"Kijang dalam tumpukan kosong";
}
void awal()
{
tumpukan.top= -1;
}
int isEmpty()
{
if (tumpukan.top==-1)
return 1;
else
return 0;
}
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
int isFull()  
{  
if (tumpukan.top==10-1)  
return 1;  
else  
return 0;  
}
```

Hasil Run Program:



```
C:\Users\RPLOnly\Documents\Project2.exe  
=====  
= CONTOH PROGRAM TUMPUKAN (STACK) =  
=====  
=====Program Tumpukan Kijang=====  
1. PUSH                               =  
2. POP                                =  
3. EXIT                               =  
=====  
[tumpukan kijang dalam kondisi kosong]  
Masukan Pilihan :1  
Masukan Kijang kedalam tumpukan :2
```

h. Koding Program Stack (Tumpukan) Kedua:

```
#include<iostream>  
#include<conio.h>  
using namespace std;  
struct STACK  
{  
int data[5];  
int atas;  
}  
tumpukan;  
STACK tumpuk;  
int main ()  
{  
int pilihan,baru,i;  
tumpuk.atas=-1; do  
{  
cout<<"1.Push Data\n";  
cout<<"2.Pop Data\n";
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"3.Print Data\n";
cout<<"4.End Data\n";
cout<<"\nPilihan = ";
cin>>pilihan;
switch(pilihan)
{
case 1:
{
if(tumpuk.atas==5-1)
{
cout<<"Tumpukan Penuh";
cout<<endl;
getch();
}
else
{
cout<<"Data yang akan di-push = ";
cin>>baru; tumpuk.atas++;
tumpuk.data[tumpuk.atas]=baru;
}
cout<<"      \n";
break;
}
case 2:
{
if(tumpuk.atas==--1)
{
cout<<"Tumpukan Kosong"; cout<<endl;
getch();
}
else
{
cout<<"Data      yang      akan      di-pop      =
"<<tumpuk.data[tumpuk.atas];
tumpuk.atas--;
cout<<endl; getch();
}
}
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"    \n";
break;
}
case 3:
{
if (tumpuk.atas==-1)
{
    cout<<"Tumpukan Kosong";
    cout<<endl;
getch();
}
else
{
cout<<"\nData pada stack saat ini adalah: \n";
for(int i=tumpuk.atas; i>=0; i--)
{
cout<<"Nilai        ruang        ke        "<<i<<":
"<<tumpuk.data[i]<<endl;
}
cout<<endl;
getch();
}
cout<<"    \n";
break;
}
default:
cout<<"\nTidak ada dalam pilihan";
cout<<endl;
cout<<"    \n";
}
}
while (pilihan!=4);
return 0;
}
```

Hasil Run Program:

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
C:\Users\RPLOnly\Documents\Project2.exe
1.Push Data
2.Pop Data
3.Print Data
4.End Data

Pilihan = 1
Data yang akan di-push = 2
-----
1.Push Data
2.Pop Data
3.Print Data
4.End Data
1
Pilihan = 3
Data pada stack saat ini adalah :
Nilai ruang ke 0 : 2
-----
1.Push Data
2.Pop Data
3.Print Data
4.End Data
1
Pilihan = 4
Tidak ada dalam pilihan
-----
Process exited after 23.01 seconds with return value 0
Press any key to continue . . .
```

4.5 Latihan Soal

Buatlah koding program bahasa C++, Data Bubble Sort?

BAB V

ALGORITMA DALAM AKUNTANSI

Capaian Pembelajaran:

1. Memahami Program Akuntansi
2. Membuat Program Akuntansi dalam Bahasa C++

Setelah mempelajari materi pada pertemuan ini, mahasiswa mampu membuat aplikasi sederhana menggunakan perhitungan dalam akuntansi dengan menerapkan algoritma secara program dalam bahasa c++.

Dalam bab ini akan menjelaskan tentang implementasi algoritma dalam akuntansi seperti:

1. Membuat Program Penjualan,
2. Membuat Program Pembelian,
3. Membuat Program Kasir,
4. Membuat Program Pemasukan,
5. Membuat Program Neraca Saldo.

Akuntansi sering disebut sebagai bahasa komunikasi antar perusahaan yang membutuhkan laporan - laporan keuangan kepada pihak-pihak yang berkepentingan. Adapun pihak yang memerlukan hasil dari proses akuntansi dapat dibedakan menjadi dua yaitu:

1. Pihak internal yaitu pengelola perusahaan (manajemen sendiri).
Pihak manajemen membutuhkan laporan akuntansi untuk proses perencanaan, penyusunan maupun kebijakan perusahaan. Pihak-pihak internal meliputi: Pimpinan Perusahaan (Manajer, Direktur, beserta jajaran manajemen).
2. Pihak eksternal terdiri atas:
 - a) Penanam modal (investor),
 - b) Calon penanam modal (Para Pemegang Saham, Calon Investor),
 - c) Pihak bank dan leveransir,
 - d) Badan pemerintah dan

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

- e) federasi buruh. Pihak eksternal memerlukan informasi untuk mengukur hasil usaha yang telah dicapai perusahaan selama ini dan prospek usaha dimasa yang akan datang sebagai dasar penentuan kebijakan investasi modalnya.

Pihak bank (leveransir) memerlukan informasi untuk mengetahui posisi keuangan dan kemampuan perusahaan dalam memperoleh keuntungan agar dapat mengetahui pinjaman yang diperlukan tersebut terjamin atau tidak. Badan pemerintah, memerlukan informasi mengenai aktifitas keuangan perusahaan sebagai alat bantu dalam penetapan pajak penghasilan bagi perusahaan yang bersangkutan. Federasi buruh memerlukan informasi keuangan untuk mensejahterakan para buruh, antarlain dengan menuntut kenaikan gaji, menuntut kehidupan yang lebih layak.⁶

Algoritma dibutuhkan dalam bidang akuntansi mengenai permasalahan-permasalahan yang dihadapi bagaimana cara untuk membuat sebuah sistem program dalam perhitungan dan materi akuntansi dalam bahasa C++.

Pengerjaan program akuntansi dengan beberapa materi seperti pembahasan diatas dapat dibuat dalam pembuatan program bahasa ++ adapun program-program tersebut berupa dibawah ini.

5.1 Program Penjualan Bahasa C++

Penjualan sistem akuntansi penjualan merupakan pembelian sesuatu (barang atau jasa dari suatu pihak kepada pihak lainnya dengan mendapatkan ganti uang dari pihak tersebut.

Dalam pembuatan program penjualan dengan bahasa C++ ini menggunakan Fungsi **IF**, **ELSE IF** dan **ELSE**. Fungsi kondisi **IF ELSE IF** adalah sebuah struktur logika program yang di dapat dengan cara menyambung beberapa perintah **IF ELSE** menjadi sebuah kesatuan. Jika kondisi pertama tidak terpenuhi atau bernilai **false**, maka kode program akan lanjut ke kondisi **IF** di bawahnya. Jika ternyata tidak juga terpenuhi, akan lanjut lagi ke kondisi **IF** di bawahnya lagi,

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

dst hingga blok **ELSE** terakhir atau terdapat kondisi **IF** yang menghasilkan nilai **true**.

Koding Program Penjualan:

```
#include <iostream.>
#include <stdio.h>
#include <conio.h>
using namespace std;
main()
{
char pilihanMerk, *merkBuku, beliLgi;
int ttlBayar, Kembalian, jmlPembelian, hargaProduk,
totalHarga;
atas:
cout<<"=====
===== "<<endl;
    cout<<"\tProgram        Sederhana        Penjualan        Buku
Ajar"<<endl;
cout<<"=====
===== "<<endl;
    cout<<endl;
    cout<<"-----
----- "<<endl;
    cout<<"Pilihan Merek Buku"<<endl;
    cout<<"-----
----- "<<endl;
    cout<<"1. Buku Algoritma"<<endl;
    cout<<"2. Buku Logika"<<endl;
    cout<<"3. Buku Struktur Data"<<endl;
    cout<<"4. Buku Akuntansi"<<endl;
    cout<<"5. Buku Pemrogramman C++"<<endl;
    cout<<"-----
----- "<<endl;
```


Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

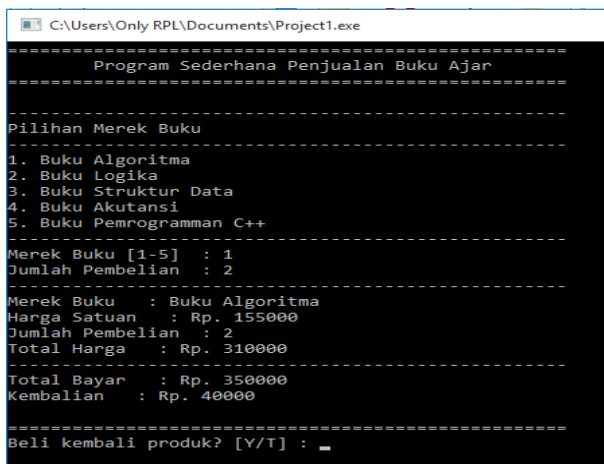
```
cout<<"Merek Buku [1-5] : ";
cin>>pilihanMerk;
cout<<"Jumlah Pembelian : ";
cin>>jmlPembelian;
cout<<"-----
-----"<<endl;

if(pilihanMerk=='1'){merkBuku="Buku Algoritma";
hargaProduk=155000;}
else if(pilihanMerk=='2'){merkBuku="Buku Logika";
hargaProduk=120000;}
else if(pilihanMerk=='3'){merkBuku="Buku Struktur
Data";
hargaProduk=150000;}
else if(pilihanMerk=='4'){merkBuku="Buku
Akuntansi";
hargaProduk=130000;}
else if(pilihanMerk=='5'){merkBuku="Buku
Pemrogramman C++";
hargaProduk=185000;}
else{
cout<<"Maaf, Merek Buku tidak tersedia"<<endl;
cout<<"Pilih kembali";getch();
cout<<endl;
goto atas;
}
totalHarga=hargaProduk*jmlPembelian;
cout<<"Merek Buku : "<<merkBuku<<endl;
cout<<"Harga Satuan : Rp. "<<hargaProduk<<endl;
cout<<"Jumlah Pembelian : "<<jmlPembelian<<endl;
cout<<"Total Harga : Rp. "<<totalHarga<<endl;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"-----"
-----"<<endl;
cout<<"Total Bayar : Rp. ";
cin>>ttlBayar;
Kembalian=ttlBayar-totalHarga;
cout<<"Kembalian : Rp. "<<Kembalian<<endl<<endl;
cout<<"=====
===== "<<endl;
cout<<"Beli kembali produk? [Y/T]: ";cin>>beliLgi;
cout<<endl;
if(beliLgi=='Y' || beliLgi=='y')
{goto atas;}
else{
cout<<"Terima Kasih Atas Pembelian Produk Kami";
cout<<endl;
}
getch();
}
```

Hasil Run Program:



The screenshot shows a Windows command prompt window titled "C:\Users\Only RPL\Documents\Project1.exe". The program output is as follows:

```
-----
Program Sederhana Penjualan Buku Ajar
-----
Pilihan Merek Buku
-----
1. Buku Algoritma
2. Buku Logika
3. Buku Struktur Data
4. Buku Akutansi
5. Buku Pemrogramman C++
-----
Merek Buku [1-5] : 1
Jumlah Pembelian : 2
-----
Merek Buku : Buku Algoritma
Harga Satuan : Rp. 155000
Jumlah Pembelian : 2
Total Harga : Rp. 310000
-----
Total Bayar : Rp. 350000
Kembalian : Rp. 40000
-----
Beli kembali produk? [Y/T] : _
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

5.2 Program Pembelian Bahasa C++

Pembelian merupakan suatu kegiatan yang dilakukan untuk pengadaan barang yang dibutuhkan perusahaan dalam menjalankan usahanya dimulai dari pemilihan sumber sampai memperoleh barang.

Koding Program:

```
#include <iostream>
#include <conio.h>
using namespace std;

main()
{
    int kode, harga, jumlah, total;
    char nama [20];
    cout << " * Program Penjualan Barang Toko Buku Uni
*\n";
    cout << "-----\n";

    cout << "Masukan Kode Barang : ";
    cin >> kode;
    cout << "Masukan Nama Barang : ";
    cin >> nama;
    cout << "Masukan Harga Barang : ";
    cin >> harga;
    cout << "Jumlah Beli      : ";
    cin >> jumlah;
    cout << "-----" <<
endl;

    total = harga * jumlah;

    cout << "Proses Total - harga x jumlah beli" << endl;

    cout << " * STRUK PEMBAYARAN *" << endl;
    cout << "/////////////////////////////////////" <<
endl;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

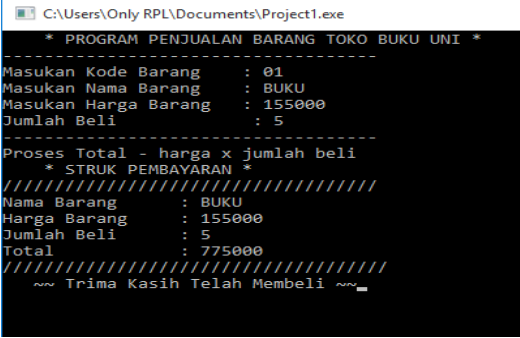
```
cout << "Nama Barang : " << nama << endl;

cout << "Harga Barang : " << harga << endl;

cout << "Jumlah Beli : " << jumlah << endl;

cout << "Total : " << total << endl;
cout << "/////////////////////////" <<
endl;
cout << " ~~ Trima Kasih Telah Membeli ~~";
getch();
}
```

Hasil Run Program:



```
C:\Users\Only RPL\Documents\Project1.exe
* PROGRAM PENJUALAN BARANG TOKO BUKU UNI *
Masukan Kode Barang : 01
Masukan Nama Barang : BUKU
Masukan Harga Barang : 155000
Jumlah Beli : 5
Proses Total - harga x jumlah beli
* STRUK PEMBAYARAN *
/////////////////////////
Nama Barang : BUKU
Harga Barang : 155000
Jumlah Beli : 5
Total : 775000
/////////////////////////
~~ Trima Kasih Telah Membeli ~~
```

5.3 Program Kasir Bahasa C++

Program kasir sangat membantu pemilik bisnis dalam melakukan proses penghitungan transaksi dan pencatatan penjualan. Awalnya, banyak pemilik bisnis yang mengandalkan mesin kasir atau cash register untuk mengatur proses transaksi. Meski mesin kasir bisa membuat proses penghitungan transaksi dan pencatatan penjualan menjadi lebih praktis, namun tetap saja ada kekurangannya tersendiri. Mesin kasir belum memiliki sistem yang terintegrasi sehingga membuat pemilik usaha harus menyediakan waktu khusus untuk memeriksa data secara manual setiap harinya. Hal ini yang akhirnya mendorong banyak pemilik bisnis untuk beralih menggunakan aplikasi kasir.

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

Fungsi Looping atau perulangan seperti for, while, do, dowhile dan lain sebagainya. kemudain pengkondisian atau statement control seperti, if, if else, if else bersarang, swit case dan lain sebagainya.

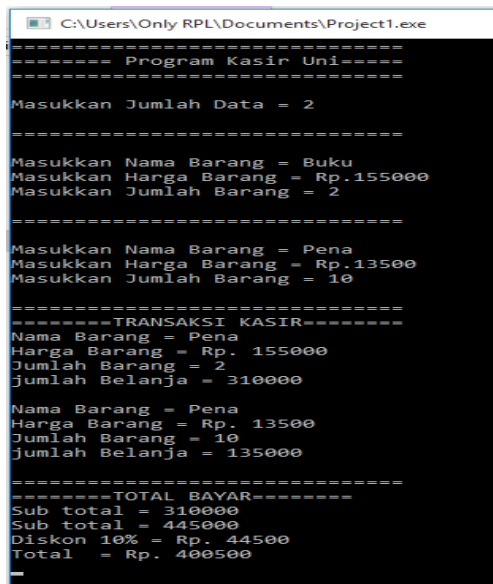
Koding Program:

```
#include <iostream>
#include <conio.h>
using namespace std;
main () {
    int diskon,i,j, x, total;
    int bayar[10], hrga[10],jbrg[10], totali=0;
    char nama[30][x];
    cout<<"===== "<<endl;
    cout<<"===== Program Kasir Uni===== "<<endl;
    cout<<"===== "<<endl;
    cout<<endl;
    cout<< "Masukkan Jumlah Data = ";
    cin>>x;
    cout << endl;
    for(i=0;i<x;i++){
        cout<<"===== "<<endl;
        cout<<endl;
        cout<< "Masukkan Nama Barang = ";
        cin>>nama[i];
        cout<< "Masukkan Harga Barang = Rp.";
        cin>>hrga[i];
        cout<< "Masukkan Jumlah Barang = ";
        cin>>jbrg[i];
        cout<<endl;
    }
    cout<<"===== "<<endl;
    cout<<"=====TRANSAKSI KASIR===== "<<endl;
    for(i=0;i<x;i++){
        cout<<"Nama Barang = "<<nama[i]<<endl;
        cout<<"Harga Barang = Rp. "<<hrga[i]<<endl;
        cout<<"Jumlah Barang = "<<jbrg[i]<<endl;
        bayar[i]=hrga[i]*jbrg[i];
    }
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"jumlah Belanja = "<<bayar[i]<<endl;
cout<<endl;
}
cout<<"=====TOTAL BAYAR===== "<<endl;
for(i=0; i<x; i++){
    totali=totali+bayar[i];
    cout<<"Sub total = "<<totali<<endl;
}
if (totali>30000){
    diskon=0.1*totali;
    cout<<"Diskon 10% = Rp. "<<diskon<<endl;
    total=totali-diskon;
    cout<<"Total = Rp. "<<total<<endl;
}
getch();
}
```

Hasil Program:



```
=====
----- Program Kasir Uni-----
=====
Masukkan Jumlah Data = 2
=====
Masukkan Nama Barang = Buku
Masukkan Harga Barang = Rp.155000
Masukkan Jumlah Barang = 2
=====
Masukkan Nama Barang = Pena
Masukkan Harga Barang = Rp.13500
Masukkan Jumlah Barang = 10
=====
-----TRANSAKSI KASIR-----
Nama Barang = Pena
Harga Barang = Rp. 155000
Jumlah Barang = 2
jumlah Belanja = 310000

Nama Barang = Pena
Harga Barang = Rp. 13500
Jumlah Barang = 10
jumlah Belanja = 135000
=====
=====TOTAL BAYAR=====
Sub total = 310000
Sub total = 445000
Diskon 10% = Rp. 44500
Total = Rp. 400500
=====
```

5.4 Program Pengajian Bahasa C++

Gaji adalah suatu bentuk pembayaran periodik dari seorang majikan pada karyawannya yang dinyatakan dalam suatu kontrak kerja. Di

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

Indonesia sendiri ukuran gaji biasanya dilakukan dalam periode per bulan.

Gaji merupakan sejumlah imbalan yang dibayarkan perusahaan kepada karyawan perusahaan dalam waktu yang tetap dalam waktu tertentu. Jadi imbalan yang kamu dapatkan setiap bulan setiap tanggal tertentu yang berisi basic salary, tunjangan tetap (jika ada) dan tunjangan tidak tetap (jika ada) dapat disebut gaji.

Dalam pembuatan program penjualan dengan bahasa C++ ini menggunakan Fungsi **IF**, **ELSE IF** dan **ELSE**. Fungsi kondisi **IF ELSE IF** adalah sebuah struktur logika program yang di dapat dengan cara menyambung beberapa perintah **IF ELSE** menjadi sebuah kesatuan. Jika kondisi pertama tidak terpenuhi atau bernilai **false**, maka kode program akan lanjut ke kondisi **IF** di bawahnya. Jika ternyata tidak juga terpenuhi, akan lanjut lagi ke kondisi **IF** di bawahnya lagi, dst hingga blok **ELSE** terakhir atau terdapat kondisi **IF** yang menghasilkan nilai **true**

Koding Program:

```
#include <iostream>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
using namespace std;
main()
{
    int jam_kerja, golongan_jabatan;
    char nama[30], golongan_pendidikan[9];
    long int gaji_pokok = 300000, honor_lembur_per_jam
= 3500, honor_lembur, tunjangan_jabatan,
tunjangan_pendidikan, total_honor;

    //Input Data
    cout<<" \n PROGRAM HITUNG GAJI KARYAWAN\n";
    cout<<"\n PT. UNI SAYANG\n";
    printf("\n=====
=====\\n\\n");
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
printf(" Nama Karyawan : "); scanf("%s", &nama);
printf(" Golongan Jabatan : "); scanf("%d",
&golongan_jabatan);
printf(" Golongan Pendidikan : "); scanf("%s",
&golongan_pendidikan);
printf(" Jumlah Jam Kerja : "); scanf("%d",
&jam_kerja);

if(golongan_jabatan==1)
{
tunjangan_jabatan =5 * gaji_pokok /100;
}
else if(golongan_jabatan==2)
{
tunjangan_jabatan = 10 * gaji_pokok /100;
}
else if(golongan_jabatan==3)
{
tunjangan_jabatan = 15 * gaji_pokok /100;
}

if ((golongan_pendidikan,"sma")||
(golongan_pendidikan,"SMA"))
{
tunjangan_pendidikan = 2.5 * gaji_pokok /100;
}
else if ((golongan_pendidikan,"d1")||
(golongan_pendidikan,"D1"))
{
tunjangan_pendidikan = 5 * gaji_pokok /100;
}
else if ((golongan_pendidikan,"d3")||
(golongan_pendidikan,"D3"))
{
tunjangan_pendidikan = 20 * gaji_pokok /100;
}
```


Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
else
if ((golongan_pendidikan, "s1") || (golongan_pendidikan
, "S1"))
{
    tunjangan_pendidikan = 30 * gaji_pokok /100;
}

if (jam_kerja >= 8)
{
    honor lembur = jam_kerja *
honor_lembur_per_jam;
}
else if (jam_kerja <=8)
{
    honor_lembur = jam_kerja * 0;
}

cout<<"\n=====
===== ";
cout<<"\n=====SLIP
GAJI===== ";

total_honor = tunjangan_jabatan +
tunjangan_pendidikan + honor_lembur + gaji_pokok;
//Ouput Data
cout<<" \n Karyawan yang bernama %s      \n", nama;
cout<<" Honor yang diterima.\n\n";
printf(" Tunjangan Jabatan : Rp. %Id
\n", tunjangan_jabatan);
printf(" Tunjangan Pendidikan : Rp. %Id
\n", tunjangan_pendidikan);
printf(" Honor Lembur : Rp. %Id \n", honor_lembur);
printf(" Gaji Pokok : Rp. %Id \n", gaji_pokok);
printf(" _____+\n");
printf(" Total Honor : Rp. %Id
\n\n\n", total_honor);
getch();
}
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

Hasil Run Program:

```
PROGRAM HITUNG GAJI KARYAWAN

PT. UNI SAYANG

=====
Nama Karyawan      : RPLubs
Golongan Jabatan   : 3
Golongan Pendidikan : S1
Jumlah Jam Kerja   : 21
=====
=====SLIP GAJI=====
Karyawan yang bernama %s
Honor yang diterima.

Tunjangan Jabatan      : Rp. 45000
Tunjangan Pendidikan   : Rp. 7500
Honor Lembur           : Rp. 73500
Gaji Pokok             : Rp. 300000
Total Honor            : Rp. 426000 +
```

5.5 Program Saldo Bahasa C++

Koding Program:

```
#include <iostream>
#include <conio.h>
using namespace std;
int main() //awal program
{ struct identitas
{
int no_rekening;
char nama[20], nama_belakang[50], alamat[100],
j_t[20];
long saldo, setor, tarik, t_t, t_s, transfer,
terima, tt, ts;
float bunga;
};
int a=-1;
identitas id[100];
char jawab, lagi, pass[6];
int n, t, input, menu, ma, mn;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
{
    menu:
    m:

    cout<<"\n+=====+\n"
;
    cout<<"| Program Tabungan Bank Uni |\n";
    cout<<"| Data Saldo Pelanggan Bank Uni |\n";
    cout<<"+=====+\n";
    cout<<"\n";
    cout<<"+=====+\n";
    cout<<"| 1. Menu Admin Bank |\n";
    cout<<"| 2. Menu Nasabah |\n";
    cout<<"| 3. Exit |\n";
    cout<<"+=====+\n";
    cout<<"\nPilih Menu 1/2/3: ";cin>>menu;
    if (menu==1)
    { //awal menu pil 1
    m_a:

    tanya:
    cout<<"\n+=====+\n";
    cout<<"| Menu Admin Bank |\n";
    cout<<"+=====+\n";
    cout<<"| 1. Input Data Nasabah |\n";
    cout<<"| 2. List Nasabah |\n";
    cout<<"| 3. Laporan Penarikan dan Setoran |\n";
    cout<<"| 4. Exit |\n";
    cout<<"+=====+\n";
    cout<<"\n";
    cout<<"Pilih Menu 1/2/3/4: ";cin>>ma;
    switch (ma)
    { //awal case of ma

    case 1: //jika admin memilih menu Input Nasabah
    { //awal menu 1
    mal: //label
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
a++;

cout<<"=====+\n";
cout<<"| Input Data Nasabah |\n";
cout<<"=====+\n";
cout<<"Nomor Rekening: ";cin>>id[a].no_rekening;
cout<<"Nama Depan : ";cin>>id[a].nama;
cout<<"Nama Belakang : ";cin>>id[a].nama_belakang;
cout<<"Alamat : ";cin>>id[a].alamat;
cout<<"Jenis Tabungan: ";cin>>id[a].j_t;
cout<<"\n";
cout<<"=====+\n";
cout<<"| Data berhasil diproses |\n";
cout<<"| |\n";
cout<<"| Terima kasih |\n";
cout<<"=====+\n";
ask: //label
cout<<"Input Data Nasabah Selanjutnya? (y/t):
";cin>>lagi;
if (lagi=='y' || lagi=='Y')
{
goto mal; }
else if (lagi=='t' || lagi=='T')
{
goto m_a;
}
else
{
cout<<"Input salah!!! \n";
goto ask;
}
getch();
} //akhir menu 1
case 2: //jika admin memilih menu list nasabah
{ //awal menu 2
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
ma2:
cout<<"=====+
\n";
    cout<<"| List Nasabah |\n";
cout<<"=====+
\n";
    for (n=0;n<=a;n++) //pengulangan untuk mencari dan
menampilkan data nasabah
    {
        cout<<"No. : "<<(n+1)<<endl;
        cout<<"Nomor Rekening :
"<<id[n].no_rekening<<endl; //endl; untuk perintah
newline atau pindah baris
        cout<<"Nama Lengkap : "<<id[n].nama<<"
"<<id[n].nama_belakang<<endl;

cout<<"=====+
\n";
    }
ask1:
cout<<"\nLihat Detail Nasabah? (y/t):
";cin>>jawab;
if (jawab=='y' || jawab=='Y')
{ nomo_rek:
    cout<<"\nMasukkan Nomor Rekening untuk detail
Nasabah: ";cin>>input;
    for (n=0;n<=a;n++)
    {
        if (input==id[n].no_rekening)
        {

            cout<<"=====+\n";
            cout<<"| Detail Data Nasabah |\n";
            cout<<"=====+\n";
            cout<<"Nomor Rekening :
"<<id[n].no_rekening<<endl;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"Nama Lengkap : "<<id[n].nama<<"
"<<id[n].nama_belakang<<endl;
cout<<"Alamat : "<<id[n].alamat<<endl;
cout<<"Jenis Tabungan : "<<id[n].j_t<<endl;
cout<<"Jumlah Saldo : Rp."<<id[n].saldo<<endl;
id[n].bunga=id[n].saldo*0.05;
cout<<"Bunga (5%/bulan): Rp."<<id[n].bunga<<endl;

cout<<"=====
=====\n";
ask2:
cout<<"\nKembali ke List Nasabah? (y/t):
";cin>>jawab;
if (jawab=='y' || jawab=='Y')
{
goto ma2;
}
else if (jawab=='t' || jawab=='T')
{
goto m_a;
}
else
{
cout<<"\n+=====+";
cout<<"\n| Input Salah!!! |";
cout<<"\n+=====+";
goto ask2;
}
}
}

cout<<"\nNomor Rekening Salah!!!\n";
goto nomo_rek;
}
else if (jawab=='t' || jawab=='T')
{
goto m_a;
}
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
else
{
cout<<"\nInput Salah!!!";
goto ask1;
}
getch();
}
case 3: // jika admin bank memilih menu laporan
dan penarikan dari setiap nasabah
{
cout<<"++++++\n";
cout<<"| Laporan Penarikan dan Setoran |\n";
cout<<"++++++\n";
cout<<" \n";
for (n=0;n<=a;n++)
{
cout<<"+++++++"
\n";
cout<<"No. : "<<(n+1)<<endl;
cout<<"Nomor Rekening :
"<<id[n].no_rekening<<endl;
cout<<"Nama Lengkap : "<<id[n].nama<<"
"<<id[n].nama_belakang<<endl;
cout<<"Alamat : "<<id[n].alamat<<endl;
cout<<"Total Setoran : Rp."<<id[n].t_s<<endl;
cout<<"Total Penarikan : Rp."<<id[n].t_t<<endl;
cout<<"Total Transfer : Rp."<<id[n].ts<<endl;
cout<<"Total Terima : Rp."<<id[n].tt<<endl;
id[n].bunga=id[n].saldo*0.05;
cout<<"Bunga (5%/bulan): Rp."<<id[n].bunga<<endl;
cout<<"Sisa Saldo : RP."<<id[n].saldo<<endl;
cout<<" \n";

cout<<"+++++++"
+";
}
cetak:
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"\nCetak Laporan? (y/t): ";cin>>jawab;
{
if (jawab=='y' || jawab=='Y')
{
for (n=0;n<=a;n++)
{

cout<<"=====
=====\\n";

cout<<"No. : "<<(n+1)<<endl;
cout<<"Nomor Rekening :
"<<id[n].no_rekening<<endl;
cout<<"Nama Lengkap : "<<id[n].nama<<"
"<<id[n].nama_belakang<<endl;
cout<<"Alamat : "<<id[n].alamat<<endl;
cout<<"Total Setoran : Rp."<<id[n].t_s<<endl;
cout<<"Total Penarikan : Rp."<<id[n].t_t<<endl;
cout<<"Total Transfer : Rp."<<id[n].ts<<endl;
cout<<"Total Terima : Rp."<<id[n].tt<<endl;
cout<<"Bunga (5%/bulan): Rp."<<id[n].bunga<<endl;
cout<<"Sisa Saldo : RP."<<id[n].saldo<<endl;
cout<<" \\n";

cout<<"=====
=====\\n";

}
}
else if (jawab=='t' || jawab=='T')
{
goto m_a;
}
else
{
cout<<"\nInput Salah!!!";
goto cetak;
}
}
```


Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
tanyal:
cout<<"\nKeterangan:";
cout<<"\n(Y/y) = untuk kembali ke menu admin
bank";
cout<<"\n(T/t) = untuk kembali ke menu utama";
cout<<"\nKembali Ke Menu Admin? (y/t):
";cin>>jawab;
if (jawab=='y' || jawab=='Y')
{
goto m_a;
}
else if (jawab=='t' || jawab=='T')
{
goto menu;
}
else
{
cout<<"\nPilihan tidak ada!!!\n";
goto tanyal;
}
getch();
} //akhir menu 3
case 4: //jika admin memilih menu keluar (exit)
{ //awal menu 4
goto menu;
getch();
} //akhir menu 4
default: cout<<"\nMenu yang dipilih tidak
ada!!!\n"; //jika admin memilih menu selain 4
goto tanya;
getch();
} //akhir case of ma (menu admin)
getch();
} //akhir menu pil 1
if (menu==2)
{ //awal menu 2
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
awal:
cout<<"\nMasukan Nomor Rekening: ";cin>>input;
for (n=0;n<=a;n++)
{
if (input==id[n].no_rekening)
{
m_n:

mn:
cout<<"\n+=====+\n";
cout<<"| Menu Nasabah |\n";
cout<<"|=====|\n";
cout<<"| 1. Setoran |\n";
cout<<"| 2. Penarikan |\n";
cout<<"| 3. Transfer |\n";
cout<<"| 4. Cek Saldo |\n";
cout<<"| 5. Exit |\n";
cout<<"+=====+\n";
cout<<"\n";
cout<<"\nMasukan Pilihan 1/2/3/4: ";cin>>mn;
switch (mn)
{
case 1:
{ //awal menu nasabah 1
cout<<"+=====+\n";
cout<<"| Penyetoran |\n";
cout<<"+=====+\n";
cout<<"\n";
{
{
cout<<"+=====+\n";
cout<<"| Detail Data Nasabah |\n";
cout<<"+=====+\n";
cout<<"Nomor Rekening: "<<id[n].no_rekening<<endl;
cout<<"Nama Lengkap : "<<id[n].nama<<"
"<<id[n].nama_belakang<<endl;
cout<<"Alamat : "<<id[n].alamat<<endl;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"Jenis Tabungan: "<<id[n].j_t<<endl;
cout<<"Jumlah Saldo : Rp."<<id[n].saldo<<endl;
cout<<"\n";
cout<<"Masukan Jumlah Uang yang akan disetor:
Rp.";cin>>id[n].setor;
cout<<"\n";
{
if (id[n].setor<20000)
{

cout<<"=====
==+\n";
cout<<"| Maaf setoran anda tidak dapat di proses
|\n";
cout<<"| Minimal setoran Rp.20.000 |\n";
cout<<"=====
==+\n";
}
else
{ id[n].saldo=id[n].setor+id[n].saldo;
//id[n].bunga=id[n].saldo*0.05;
id[n].t_s=id[n].setor+id[n].t_s;
cout<<"=====+\n";
cout<<"| Setoran berhasil diproses |\n";
cout<<"=====+\n";
cout<<"\n"<<endl;
cout<<"Jumlah saldo anda: Rp."<<id[n].saldo;
cout<<"\n"<<endl;
}
}
}
}
tanya2:
cout<<"\n (Y/y) = Kembali Ke Menu Nasabah";
cout<<"\n (T/t) = Kembali Ke Menu Utama";
cout<<"\nKembali Ke Menu Nasabah? (y/t) :
";cin>>jawab;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
if (jawab=='y' || jawab=='Y')
{
goto m_n;
}
if (jawab=='t' || jawab=='T')
{
goto menu;
}
else
{
cout<<"\nPilihan tidak ada!!!\n";
goto tanya2;
}

getch();
}
case 2:
{

cout<<"=====+\\n";
cout<<"| Penarikan |\\n";
cout<<"=====+\\n";
cout<<"\\n";
{
{
cout<<"=====+\\n";
cout<<"| Detail Data Nasabah |\\n";
cout<<"=====+\\n";
cout<<"Nomor Rekening: "<<id[n].no_rekening<<endl;
cout<<"Nama Lengkap : "<<id[n].nama<<"
"<<id[n].nama_belakang<<endl;
cout<<"Alamat : "<<id[n].alamat<<endl;
cout<<"Jenis Tabungan: "<<id[n].j_t<<endl;
cout<<"Jumlah Saldo : Rp."<<id[n].saldo<<endl;
cout<<"\\n";
cout<<"Masukan Jumlah Uang yang akan ditarik:
Rp.";cin>>id[n].tarik;
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"\n";
if (id[n].saldo-id[n].tarik<10000)
{
cout<<"+++++++\n";
cout<<"| Maaf Saldo Anda Tidak Mencukupi |\n";
cout<<"| Sisa Saldo Minimal Rp.10.000 |\n";
cout<<"+++++++\n";
}
else
{
id[n].saldo=id[n].saldo-id[n].tarik;
//id[n].bunga=id[n].saldo*0.05;
id[n].t_t=id[n].tarik+id[n].t_t;

cout<<"+++++\n";
cout<<"| Penarikan berhasil diproses |\n";
cout<<"| Silahkan Ambil Uang Anda |\n";
cout<<"+++++\n";
cout<<"\n"<<endl;
cout<<"Sisa saldo anda: Rp."<<id[n].saldo;
cout<<"\n"<<endl;
}
}
}
tanya3:
cout<<"\n (Y/y) = Kembali Ke Menu Nasabah";
cout<<"\n (T/t) = Kembali Ke Menu Utama";
cout<<"\nKembali Ke Menu Nasabah? (y/t): ";
    cin>>jawab;
if (jawab=='y' || jawab=='Y')
{
goto m_n;
}
if (jawab=='t' || jawab=='T')
{
goto menu;
}
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
else
{
cout<<"\nPilihan tidak ada!!!\n";
goto tanya3;
}
getch();
}
case 3:
{
cout<<"++++++++++++++++++++++++++++++++++++\n";
cout<<"| Transfer |\n";
cout<<"++++++++++++++++++++++++++++++++++++\n";
mn4:
cout<<"\nMasukan Nomor Rekening Tujuan:
";cin>>input;
for (t=0;t<=a;t++)
{
if (input==id[t].no_rekening)
{
cout<<"\n++++++++++++++++++++++++++++++++++++
++++++++++++++++++++\n";
cout<<"Nomor Rekening: "<<id[t].no_rekening<<endl;
cout<<"Nama Lengkap : "<<id[t].nama<<
"<<id[t].nama_belakang<<endl;
cout<<"Alamat : "<<id[t].alamat<<endl;
cout<<"\nMasukan jumlah Transfer:
Rp.";cin>>id[t].transfer;
tanya5:
cout<<"\nData Sudah Benar?";
cout<<"\nProses Transfer? (y/t): ";cin>>jawab;
if (jawab=='y' || jawab=='Y')
{
if (id[n].saldo-id[t].transfer<20000)
{
cout<<"++++++++++++++++++++++++++++++++++++\n";
cout<<"| Maaf Saldo Anda Tidak Mencukupi |\n";
cout<<"| Sisa Saldo Minimal Rp.20.000 |\n";
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"=====+\n";
}
else
{ id[n].saldo=id[n].saldo-id[t].transfer;
id[t].saldo=id[t].transfer+id[t].saldo;
id[t].tt=id[t].transfer+id[t].tt;
id[n].ts=id[t].transfer+id[n].ts;

cout<<"=====+\n";
cout<<"| Transfer Berhasil Diperoses |\n";
cout<<"| Terima Kasih |\n";
cout<<"=====+\n";
cout<<"\n Sisa Saldo: "<<id[n].saldo<<endl;
}
tanya7:
cout<<"\n=====+\n";
cout<<"\nKeterangan =";
cout<<"\n(Y/y) = Kembali Ke Menu Nasabah";
cout<<"\n(T/t) = Kembali Ke Menu Utama";
cout<<"\nKembali ke Menu Nasabah? (y/t) :
";cin>>jawab;
if (jawab=='y' || jawab=='Y')
{
goto m_n;
}
else if (jawab=='t' || jawab=='T')
{
goto menu;
}
else
{
cout<<"\nInput Salah!!!\n";
goto tanya7;
}
}
else if (jawab=='t' || jawab=='T')
{
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
goto mn4;
}
else
{
cout<<"\nInput Salah!!!\n";
goto tanya5;
}
}
}

cout<<"\nNomor Rekening tidak terdaftar!!!\n";
tanya6:
cout<<"\nKembali Ke Menu? (y/t): ";cin>>jawab;
if (jawab=='y' || jawab=='Y')
{
goto m_n;
}
else if (jawab=='t' || jawab=='T')
{
goto mn4;
}
else
{
cout<<"\nInput Salah!!!\n";
goto tanya6;
}
getch();
} //akhir menu nasabah 3
case 4:
{
cout<<"++++++++++++++++++++++++++++++++++++\n";
cout<<"| Cek Saldo |\n";
cout<<"++++++++++++++++++++++++++++++++++++\n";
{
{
cout<<"\n++++++++++++++++++++++++++++++++++++\n";
cout<<"| Detail Data Nasabah |\n";
cout<<"++++++++++++++++++++++++++++++++++++\n";
```


Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
cout<<"\n";
cout<<"Nomor Rekening :
"<<id[n].no_rekening<<endl;
cout<<"Nama Lengkap : "<<id[n].nama<<"
"<<id[n].nama_belakang<<endl;
cout<<"Alamat : "<<id[n].alamat<<endl;
cout<<"Jenis Tabungan : "<<id[n].j_t<<endl;
id[n].bunga=id[n].saldo*0.05;
cout<<"Bunga (5%/bulan): Rp."<<id[n].bunga<<endl;
cout<<"Jumlah Saldo : Rp."<<id[n].saldo<<endl;
}
}
tanya4:
cout<<"\n (Y/y) = Kembali Ke Menu Nasabah";
cout<<"\n (T/t) = Kembali Ke Menu Utama";
cout<<"\nKembali Ke Menu Nasabah? (y/t):
";cin>>jawab;
if (jawab=='y' || jawab=='Y')
{
goto m_n;
}
if (jawab=='t' || jawab=='T')
{
goto menu;
}
else
{
cout<<"\nPilihan tidak ada!!!\n";
goto tanya4;
}
getch();
} // akhir menu nasabah 4
case 5:
{ //awal menu nasabah 5
goto menu;
} //akhir menu nasabah 5
default:
```

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
{
cout<<"Menu yang anda pilih tidak ada!!!\n";
goto mn;
}
getch();
}
}
}
cout<<"\nNomor Rekening Salah!!!\n";
goto awal;
} //akhir menu 2
else if (menu>3)
{
cout<<"\nMenu yang dipilih tidak ada!!!\n";
goto m;
}
} //akhir program menu awal
//getch();
return 0;
} //akhir program
```

Hasil Run Program:

Dasar Logika Algoritma dan Pemrograman dengan Bahasa C++

```
C:\Users\Only RPL\Documents\Project1.exe
Kembali Ke Menu Nasabah ? (y/t) : y

+-----+
|           Menu Nasabah           |
+-----+
| 1. Setoran                       |
| 2. Penarikan                     |
| 3. Transfer                      |
| 4. Cek Saldo                     |
| 5. Exit                         |
+-----+

Masukan Pilihan 1/2/3/4 : 4

+-----+
|           Cek Saldo              |
+-----+

+-----+
|           Detail Data Nasabah    |
+-----+

Nomor Rekening   : 751123
Nama Lengkap    : Rizky Lubis
Alamat          : Jalan
Jenis Tabungan   : Puri
Bunga (5%/bulan) : Rp.10000
Jumlah Saldo     : Rp.200000

(Y/y) = Kembali Ke Menu Nasabah
(T/t) = Kembali Ke Menu Utama
Kembali Ke Menu Nasabah ? (y/t) : _
```

DAFTAR PUSTAKA

1. Susanto, Wahyu Eko AS. *Algoritma. Log dan Algoritma untuk Pemula*. Published online 2020.
2. Barakbah AR, Karlita T, Ahsan AS. *Logika Dan Algoritma Pemrograman*. Published online 2013:1-102.
3. Prajarini D. *Pemrograman dasar*. Published online 2013:1-49.
4. Muhidin A. *Pemrogram Bahasa C ++. Pemrogram Bhs C ++*. Published online 2010:1-56.
5. Pemrograman ADAN, Prasetyo AB, Kom S, Kom M, Pemrograman ADAN. *Algoritma Dan Pemrograman 1*.
6. Rakhmanita A. *Modul akuntansi dasar dan praktik*. Published online 2019:15.

DASAR LOGIKA ALGORITMA & Pemrograman Dengan Bahasa C++

SRI HARYATI

Pengertian Algoritma merupakan urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis. Algoritma yang dapat menyelesaikan suatu permasalahan dalam waktu yang singkat memiliki tingkat kerumitan yang rendah, sementara algoritma yang membutuhkan waktu lama untuk menyelesaikan suatu masalah membutuhkan tingkat kerumitan yang tinggi.

Defenisi dari algoritma sangat lekat dengan kata logika, yaitu kemampuan seorang manusia untuk berfikir dengan akal tentang suatu permasalahan menghasilkan sebuah kebenaran, dibuktikan dan dapat diterima akal, logika seringkali dihubungkan dengan kecerdasan, seseorang yang mampu berlogika dengan baik sering orang menyebutnya sebagai pribadi yang cerdas. Dalam menyelesaikan suatu masalah pun logika mutlak diperlukan untuk menyelesaikan permasalahan tersebut.



Penerbit Poliban Press

Redaksi :

Politeknik Negeri Banjarmasin, Jl. Brigjen H. Hasan Basry,
Pangeran, Komp. Kampus ULM, Banjarmasin Utara

Telp : (0511)3305052

Email : press@poliban.ac.id

ISBN 978-623-7694-92-2 (PDF)



ISBN 978-623-7694-91-5

