NOTASI BAHASA DAN STRUKTUR DASAR ALGORITMA PEMROGRAMAN

Oleh : Mawaddah Nur Asri

Jurusan Pendidikan Matematika Universitas Islam Negeri Sumatera Utara (UINSU)

email : mawaddah0802@gmail.com

Abstrak

Notasi Bahasa dan Struktur Dasar Algoritma Pemrograman nantinya sangat berguna dalam melakukan pemrograman dalam algoritma, hal ini akan sering dilakukan oleh para pelaku algoritma atau orang yang merancang algoritma sehingga notasi bahasa dan struktur dasar ini perlu diketahui bagi pelaku tersebut. Notasi bahasa sendiri banyak menggunakan bahasa C, C+, C++, namun banyak notasi bahasa lainnya yang digunakan dalam algoritma pemrograman. Namun, struktur dasarnya memiliki 3 struktur yang sangat mendasar, hal ini akan tersedia sampai ke penjelasan yang lebih lanjut dan lebih lengkap.

Kata kunci : Notasi Bahasa, Struktur Dasar, Algoritma, Pemrograman

PENDAHULUAN

Jika mendengar kata “Algoritma” maka kita pasti akan langsung berspekulasi pada arah istilah matematika yang tentu saja akan mengalami proses hitung – hitungan. Namun, sebenarnya Algoritma itu memiliki makna yang lebih daripada itu dan berbeda. Definisi Algoritma yang sesungguhnya adalah pernyataan yang berupa urutan atau bisa disebut dengan langkah yang logis tentang suatu penyelesaian yang masalahnya dapat disusun secara sistematis dan logis, namun pernyataan tersebut harus dibuat tanpa menghilangkan makna kebenarannya. Hal ini bermakna Algoritma itu memiliki arti bahwa langkah langkah suatu permasalahan yang akan dipecahkan dan berbeda dengan definisi matematika yang memecahkan soal dengan baku.

Sehingga Algoritma Pemrograman dapat disusun dengan cara yang lebih santai atau tidak baku seperti yang ada pada matematika.

Selanjutnya jika “Algoritma” ditambahkan dengan adanya “Struktur Data” akan menjadi suatu “Program” yang biasa digunakan dalam program komputer. Sehingga dapat ditarik sebuah kesimpulan bahwa “Program” adalah kumpulan dari beberapa pernyataan yang ada pada komputer, sedangkan metode dan tahapan sistematisnya yang ada di dalam program tersebut adalah algoritma.

Algoritma Pemrograman dalam penulisan bahasanya disebut dengan notasi bahasa Algoritma. Notasi bahasa itu tidak menggunakan hal atau sesuatu yang baku di dalam penulisannya. Namun, sebuah notasi Algoritma

merupakan sebuah notasi yang bukan berada pada bahasa di dalam pemrogramannya sehingga siapapun yang akan menuliskan dapat memuat notasi algoritma yang berbeda-beda setiap orangnya. Akan tetapi terdapat aturan-aturan dan kaidah - kaidah sehingga harus ditaati untuk menghindari kekeliruan dan ketidak tepatan akan notasi itu agar komputer dapat mengerti apa yang dimaksudkan.

Dalam Algoritma Pemrograman juga terdapat struktur yang menjadi dasar pemikiran Algoritma Pemrograman itu sendiri. Struktur dasar ini tentunya juga disusun dengan terurut guna menyusun Algoritma Pemrograman yang benar sesuai dengan kaidahnya.

METODE

Dengan artikel yang berjudul “Notasi Bahasa dan Struktur Dasar Algoritma Pemrograman” ini menggunakan metode penelitian dan pengembangan atau nama lainnya Research and Development Method. Metode ini merupakan metode penelitian yang jika digunakan dapat sangat berfungsi bagi kebutuhan pembaca sehingga metode ini memiliki sifat dapat menganalisis sebuah kebutuhan dan juga dapat berguna untuk menguji keefektifan suatu produk yang diujikan kepada bahan materi yang diuji. Ada banyak peneliti yang menggunakan metode ini seperti Borg & Gall, beliau meneliti di tahun 1989. Ia juga melakukan penelitian model ini dan menerapkan langkah-langkah yang dimulai dari tahap definisi masalah lalu ke tahap pengumpulan data sampai ke tahap produksi masal dari bahan

yang diteliti. Menurut penulis metode ini merupakan metode yang tepat untuk judul artikel ini karena penulis dapat menyajikan mulai dari pengertian sampai bagaimana bila masyarakat menggunakan produk yang diuji sehingga pembaca dapat memahami dan tertarik bila dibahas atau diteliti dengan menggunakan metode ini.

Notasi Bahasa Algoritma Pemrograman dengan Bentuk Suatu Kalimat Deskriptif

Penulisan Algoritma dapat dilakukan berbagai cara, bahkan dapat dilakukan seperti kegiatan yang sering kita lakukan dengan bahasa sehari-hari sehingga hal ini merupakan notasi bahasa yang digunakan dalam keadaan yang santai serta mengandung unsur kemudahan bagi penggunaannya. Penulisan Algoritma bisa dilakukan dengan menggunakan kalimat deskriptif yang di dalamnya menjelaskan instruksi atau urutan yang akan dilakukan dengan jelas. Mengingat makna deskriptif sendiri adalah kalimat yang isinya sebuah pemaparan atau penjelasan yang bersifat rinci dan jelas. Pemaparan atau penggambaran ini umumnya melibatkan kosa kata yang dapat memberikan pengalaman lebih pada indera pembaca atau pendengar kalimat. Penulisan algoritma dengan kalimat deskriptif ini merupakan cara yang paling mudah dan sederhana untuk dibuat namun memiliki beberapa kekurangan seperti akan adanya notasi bahasa yang panjang sehingga membuat kinerja komputer kurang maksimal. Kekurangan lainnya berupa kalimat deskriptif

yang bersifat ambigu sehingga kalimat deskriptif tidak memiliki karakteristik yang pasti dan kalimat deskriptif ini biasanya jarang digunakan untuk menyelesaikan masalah seperti pokok bahasan kita mengenai Algoritma Pemrograman yang mengurutkan jalan pemecahan masalah. Suatu algoritma harus memuat bahasa yang jelas dan tidak boleh memiliki arti lebih atau makna ganda dalam satu kosa katanya. Namun, karena notasi ini merupakan notasi yang paling mudah untuk dipelajari maka dapat dijadikan rekomendasi untuk para pemula yang ingin mempelajari pemrograman dalam algoritma.

Seperti halnya kalimat deskriptif yang biasa digunakan memiliki beberapa bagian meliputi :

a. Head / Judul

Pada bagian Head atau judul ini memuat nama atau suatu hal yang nantinya akan dibuat ke dalam algoritma. Kebanyakan pengguna yang menggunakan notasi ini kebanyakannya memuat nama yang biasanya sudah dibuat sehingga dapat memberikan sebuah gambaran pada aturan dari penyelesaian suatu masalah dan juga masalah yang nantinya akan diselesaikan.

b. Deklarasi / Pernyataan

Deklarasi atau pernyataan merupakan bagian yang menyatakan jenis dari setiap elemen data atau sering disebut dengan sebuah variabel yang nantinya akan diperlukan dalam algoritma.

c. Deskripsi / Penjelasan

Deskripsi atau penjelasan merupakan pusat dari sebuah prosedur guna menjadi penyelesaian dalam masalah Algoritma Pemrograman, hal ini meliputi sebuah atau lebih dari pernyataan atau bisa disebut dengan operasi, fungsi, penjelasan, dan sebagainya.

Berikut merupakan contoh algoritma dengan menggunakan kalimat deskriptif :

Algoritma pada Luas Bangun Datar (Trapezium)

Adapun program menghitung luas trapesium dan menampilkannya ke layar dengan inputan berupa panjang sisi trapesium. Dengan deklarasi :

Panjang sisi atas = real

Panjang sisi bawah = real

Tinggi yang menggabungkan kedua panjang sisi = real

luas = real (luas trapesium dengan tipe data bilangan bulat)

Maka didapat deskripsi sebagai berikut :

1. Baca kedua panjang sisi yang berbeda tersebut
2. Hitung luas trapesium = $\frac{(\text{panjang sisi atas} + \text{panjang sisi bawah}) \times \text{tinggi trapesium}}{2}$
3. Tampilkan hasil luas ke layar
4. Selesai.

Notasi Bahasa Algoritma Pemrograman dengan Notasi Flow Chart

Notasi bahasa Flow chart adalah notasi bahasa berupa suatu penggambaran yang grafis dari suatu notasi algoritma. Dengan penggunaan flow chart ini kita dapat melakukan notasi dengan menggambarkan urutan atau langkah - langkah yang berisikan suatu pernyataan dalam penulisan algoritma yang akan diprogram. Flow chart ini biasanya berisikan sekumpulan simbol - simbol yang di dalamnya menggambarkan suatu proses tertentu. Suatu algoritma yang ditulis menggunakan notasi bahasa satu ini dapat menggunakan simbol-simbol sebagai berikut :

Simbol	Maksud
Terminal (START, END)
Input/Output (READ, WRITE)
Proses
Pemilihan atau pencabangan (YES, NO)
Menampilkan
Alur
Titik sambungan pada halaman yang sama
Titik konektor yang berada pada halaman lain
Call (Memanggil subprogram)
Dokumen
Stored Data (Simpanan Data)
Preparation (Pemberian nilai awal suatu variabel)

Contoh penerapan notasi bahasa Flow Chart dalam menghitung bangun ruang (trapesium)

Notasi Bahasa Algoritma Pemrograman dengan Notasi Pseudo Code

Notasi bahasa Pseudo code adalah notasi algoritma yang penggunaannya dibidang menyerupai notasi bahasa pemrograman tingkat tinggi. Ada beberapa keuntungan jika kita menggunakan pseudo code, diantaranya adalah kemudahan dalam mentranslasi ke notasi bahasa pemrograman, hal ini karena terdapat beberapa kemiripan dengan notasi bahasa sebuah pemrograman. Seorang programmer atau yang akan melakukan pemrograman algoritma biasanya ingin menerapkan algoritma tertentu, terutama yang lebih kompleks atau bisa menggunakan algoritma baru, biasanya programmer tersebut akan memulai programnya dengan membuat deskripsi dalam bentuk pseudo code. Setelah pseudo code tersebut sudah siap diaplikasikan, maka langkah yang akan dilakukan berikutnya hanya tinggal membuat

terjemahannya saja ke bahasa pemrograman yang akan dituju.

Berikut notasi notasi bahasa Pseudo code yaitu sebagai berikut :

Pernyataan	Notasi algoritmik	Maksud
Penulisan	Cout (x)	Nilai x dicetak di piranti keluaran
	Cout (x,y)	Nilai x dan y dicetak di piranti keluaran
	Cout ("Hello")	Text Hello dicetak di piranti keluaran
Pembacaan	read(a)	Baca nilai a
	read(a,b)	Baca nilai a,b
Penugasan	bil←x	Isikan nilai variabel x kedalam variabel bil
Komentar	// komentar	Komentar untuk 1 (satu) baris dimulai dengan //
	/* Komentar baris 1	Komentar untuk banyak baris dimulai dengan /* dan diakhiri dengan */
	Komentar baris 2	
	--- Komentar baris n */	
Ekspresi	a > b	Ekspresi boolean yang akan memiliki nilai kembalian true jika nilai a lebih besar dari nilai b , apabila tidak maka nilai kembaliannya false
	a >= b	Ekspresi boolean yang akan memiliki nilai kembalian true jika a >= b , apabila tidak maka nilai kembaliannya false
	a < b	Ekspresi boolean yang akan memiliki nilai kembalian true jika nilai a < nilai b , apabila tidak maka nilai kembaliannya false
	a <= b	Ekspresi boolean yang akan memiliki nilai kembalian true jika nilai a <= b , apabila tidak maka nilai kembaliannya false
	a == b	Ekspresi boolean yang akan memiliki nilai kembalian true jika nilai a = nilai b , apabila tidak maka nilai kembaliannya false
	a != b	Ekspresi boolean yang akan memiliki nilai kembalian true jika nilai a tidak sama dengan nilai b , apabila tidak maka nilai kembaliannya false
	a AND b	Ekspresi boolean yang akan memiliki nilai kembalian true jika kedua kondisi (a dan b) bernilai true , apabila tidak maka nilai kembaliannya false
	a OR b	Ekspresi boolean yang akan memiliki nilai kembalian true jika salah satu

		kondisi (a dan b) bernilai true , apabila tidak maka nilai kembaliannya false
	Not a	Ekspresi boolean yang akan memiliki nilai kembalian true jika hasil evaluasi nilai a adalah false , apabila tidak maka nilai kembaliannya false
Kondisi	If <kondisi> then <pernyataan>	Jika kondisi true / benar maka pernyataan akan dijalankan
	if <condi> then <case 1> else <case 2>	apabila kondisi true / benar maka pernyataan 1 akan dijalankan sebaliknya apabila kondisi false / salah maka pernyataan 2 yang akan dijalankan.
Pengulangan	while <cond> do { case }	Pengulangan pernyataan akan dijalankan selama kondisi true / benar apabila kondisi false / salah maka pengulangan akan dihentikan
	repeat { case } until <cond>	Pengulangan pernyataan akan dijalankan selama kondisi true / benar apabila kondisi false / salah maka pengulangan akan dihentikan
	for variable = nilai1 to nilai2 { pernyataan }	Pengulangan pernyataan akan dijalankan dari variable nilai1 sampai dengan nilai2

Berikut ini adalah contoh suatu algoritma menggunakan pseudo code untuk mencetak pesan lulus apabila nilai ujian adalah lebih besar dari 60.

Algoritma Lulus { mencetak pesan Lulus jika nilai ujian >= 60 }

Dengan deklarasi sebagai berikut :

nilai : integer

Deskripsi: read(nilai)

if nilai >= 60 then

print('Lulus')

else

print('Tida k Lulus')

endif

Notasi pseudo code ini juga dapat digunakan dalam berbagai bentuk seperti dalam notasi kalimat deskriptif, dan notasi notasi yang ada

dalam pemrograman sehingga notasi ini mencakup suatu bahasan yang sangat luas.

Notasi Bahasa Algoritma Pemrograman dengan Notasi Pascal

Bahasa Pascal juga merupakan salah satu bahasa pemrograman yang digolongkan ke dalam bahasa tingkat tinggi. Definisi Bahasa Pascal sendiri adalah bahasa yang digunakan dalam pemrograman yang dibuat oleh Niklaus Wirth pada tahun 1970. Bahasa Pascal ini diciptakan atas landasan kesederhanaan dalam membuat program dan sebagai alat untuk mempelajari pemrograman, bahasa ini juga diciptakan dengan menggunakan bahasa ALGOL 68 yang juga dibuat oleh Niklaus Wirth dengan menggunakan bahasa yang lebih terstruktur yaitu hanya memiliki satu buah pintu masuk dan satu buah pintu keluar sehingga hal ini menjadi faktor yang menyebabkan notasi bahasa ini menjadi mudah jika digunakan karena notasi ini tidak menggunakan membuat program tidak menjadi sebuah alur program tidak menjadi sebuah alur yang berbelit – belit. Namun, tentu saja harus menggunakan aturan penulisan program yang harus diikuti sehingga dapat menjadikan seseorang yang baru saja belajar bahasa pemrograman menjadi lebih teliti dan konsisten dalam pembuatan program dan dapat menghindari pembuatan program dengan pola penulisan sekehendak programmer sehingga sulit untuk ditiru atau dilakukan plagiasi oleh orang lain karena pola penulisan yang tidak baku.

Sampai sekarang bahasa Pascal sudah memiliki banyak varian, jenis, atau tipe. Terlebih lagi dengan munculnya standar bahasa Pascal pada tahun 1974 yang kemudian dikembangkan lebih jauh lagi oleh Borland International sehingga muncul sebuah produk yang cukup terkenal, yaitu “Turbo Pascal”.

Berikut ini merupakan beberapa tipe jenis notasi bahasa Pascal selain produk Turbo Pascal, yaitu sebagai berikut:

1. Free Pascal
2. MySQL (My Structured Query Language)

Untuk membuat program dengan menggunakan bahasa Pascal, terdapat beberapa hal dasar yang harus diketahui oleh programmer Pascal pemula sehingga dapat mengurangi kemunculan kesalahan yang bisa terjadi, yaitu dengan beberapa aturan sebagai berikut :

1. Struktur Program

Bahasa Pascal memiliki struktur atau aturan seperti berikut :

Program <nama_program>;

Uses <nama_unit>;

Type

.... <nama tipe variabel dan tipe

.... variabel>;

Const

.... <nama konstanta dan nilainya>

....

Var

.....<variabel/penampung yang akan

..... digunakan>;

Begin

.... <baris – baris perintah program

.... utama>

End.

Contoh program sederhana

Program contoh01;

Begin

```
writeln ('Hello, Anda telah  
berhasil melakukan percobaan');  
writeln ('membuat program  
bahasa Pascal yang lebih baik  
dari yang pertama!');
```

End.

Maka langkah - langkah yang dilakukan selanjutnya adalah sebagai berikut :

1. Simpan program tersebut yang telah disiapkan tadi, jika menjalankan editor vi, gunakan perintah :w (titik dua dan huruf w)

2. Komplikasikan program dari konsole/terminal dengan perintah :

fpc cth-01.pas

3. Jalankan program dengan memanggil nama file hasil kompilasi dengan cara :

./cth-01

Jika semua sudah benar maka akan menghasilkan tulisan sesuai dengan yang ada pada listing program cth01.pas. seperti berikut :

```
[kris@earth listing]$ fpc cth19.pas  
Free Pascal Compiler version 1.9.5 [2004/10/01] for i386  
Copyright (c) 1993-2004 by Florian Klaempfl  
Target OS: Linux for i386  
Compiling cth19.pas  
Linking cth19  
26 Lines compiled, 0.0 sec  
[kris@earth listing]$ ./cth19  
Nama      : Kris  
Alamat    : Sidodadi Barat 11  
Tahun lahir : 1980  
  
Nama Anda Kris  
Anda bertempat tinggal di Sidodadi Barat 11  
Usia Anda saat ini 25 tahun  
[kris@earth listing]$
```

Notasi Bahasa Algoritma Pemrograman

dengan Bahasa C, C#, C++

Bahasa C

Bahasa C adalah bahasa programan paling banyak dipakai oleh para programmer terlebih dalam Algoritma Pemrograman guna membuat program – program aplikasi, mulai dari aplikasi sederhana sampai aplikasi yang kompleks dan menguras pikiran. Bahasa C pertama kali dikembangkan oleh Dennis Ritchie sekitar tahun 1970-an di Bell Telephone Laboratories Inc.

Struktur dari program bahasa C sendiri dapat dilihat sebagai kumpulan dari satu atau lebih fungsi dengan meminimalkan satu fungsi, yaitu main. Setiap kunci dibuka dan ditutup dengan kurung kurawal buka dan tutup ({...}) yang diantara kurung kurawal tersebut ditulis pernyataan pernyataan program C yang biasanya ditulis dengan huruf kecil semua dan pada akhir sebuah pernyataan atau perintah diberi penutup titik koma (;)

Contoh sederhananya :

```
main()  
{ pernyataan – pernyataan; pernyataan yang  
panjang namun harus ditulis pada satu baris  
pada program, namun bisa juga berbentuk  
pernyataan yang pendek }
```

Namun, dapat diberi catatan bahwa gaya penulisan pada bahasa C ini tidak akan mempengaruhi perintah dalam program yang dibuat, akan tetapi mempengaruhi pengertian dan pemahaman tentang program tersebut.

Ada beberapa alasan mengapa bahasa C lebih sering digunakan programmer dalam memprogram Algoritma, yaitu sebagai berikut :

1. Karena bahasa C tersedia di hampir semua jenis komputer, baik mikro, mini maupun komputer besar (*Mainframe Computer*),
2. Kode bahasa C bersifat portabel, sehingga aplikasi yang ditulis dengan bahasa ini dapat digunakan di berbagai komputer dengan hanya sedikit modifikasi,
3. Bahasa C hanya menggunakan sedikit kata kunci, yaitu 32 (standar ANSI) atau 39 (Turbo C). Semakin sedikit kata kunci suatu program tersebut maka semakin mudah bagi pemakai untuk mempelajari dan menggunakan bahasa tersebut. Akibatnya proses *executable program* bahasa ini lebih cepat.
4. Mempunyai dukungan pustaka cukup banyak,
5. Bahasa C adalah bahasa yang terstruktur sehingga mudah dipahami
6. Bahasa C adalah *compiler* yang menghasilkan *executable program* yang banyak dibutuhkan oleh pasar pasar komersial.

Contoh penggunaan bahasa C dalam Program Algoritma yaitu sebagai berikut :

```
/*PROGRAM KALENDER*/
/*Terimakasih Kepada Niloy Mondal*/
#include<iostream.h>
#include <conio.h>
```

```
/*Memberikan perhitungan hari mulai dari
1/1/1900 sampai bulan dan tahun yang
diinginkan*/
```

```
Unsigned long days=0;
```

```
/*Menyatakan jumlah hari yang akan
ditampilkan*/
```

```
void show_calender(int n)
```

```
{int i,row=5,column,k;
```

```
Cout<< "\tMinggu Senin Selasa Rabu Kamis
Jumat Sabtu";
```

```
for(i=1;i<=n;i++,days++)
```

```
{k=days%7; /*Membagi hari dengan 7*/
```

```
/*Menampilkan angka bentuk kalender*/
```

```
gotoxy(k*10+8,row);
```

```
cout<<i;
```

```
if (k==6)/*Ganti baris bila harinya adalah
Sabtu*/row++;}}
```

```
void main()
```

```
{int i,month,year;
```

```
clrscr();
```

```
cout<<"Masukkan Bulan dan Tahun:-";
```

```
cin>>month>>year;
```

```
for(i=1900;i<year;i++)
```

```
{if(i%4==0)
```

```
/*Jika tahun bisa dibagi 4, maka harinya
menjadi 366*/
```

```
days+=365; }
```

```
for(i=1;i<month;i++)
```

```
{if(i==1||i==3||1==5||1==7||i==8||i==10||i==1
2)
```

```
/*Untuk Januari, Maret, Mei, Juli, Agustus,
Oktober, dan days+=31;
```

```
else if(i==4||i==6||i==9||i==11)
```



```

/*Untuk April, Juni, September, dan
November hitung 30 hari*/
    days+=30;
else if(i==2&&year%4==0)
/*Jika bulannya adalah Februari dan tahunnya
bisa dibagi 4,
    days+=29;
else
    /*Jika bulannya adalah Februari dan
    tahunnya tidak bisa dibagi 4, maka hitung 28
    hari*/
    days+=28; }
if(i==1||i==3||i==5||i==7||i==8||i==10||i==12)
    show_calender(31);
else if(i==4||i==6||i==9||i==11)
    show_calender(29);
else
    show_calender (28);
getch();}

```

Maka akan keluar hasil output sebagai berikut:

Masukkan Bulan dan Tahun:- >5 2004						
>ENTER						
Minggu	Senin	Selasa	Rabu	Kamis	Jumat	Sabtu
						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	26	27	28	29
30	31					

Bahasa C#

Bahasa C# atau dikenal dengan Visual C-Sharp merupakan salah satu bahasa pemrograman berorientasi objek yang dikeluarkan Microsoft,

ditandatangani oleh Anders Helsberg dan diperkenalkan untuk pertama kali pada bulan Juli 2000. Visual C# merupakan bahasa pemrograman modern berorientasi objek yang menjadi bahasa pemrograman utama dalam platform Microsoft.NET Framework. Visual C# dianggap sebagai kombinasi antara efisiensi pemrograman C++, kesederhanaan pemrograman java, dan penyederhanaan dari pemrograman Visual Basic. Notasi ini dapat ditemukan dalam paket Microsoft Visual Studio.

Bahasa C# ini sendiri tentunya memiliki aturan yang terstruktur guna membedakan dari bahasa C# yang salah dan yang benar. Berikut beberapa struktur atau aturan notasi bahasa C# :

1. Menggunakan *Statement*, hal ini berada di bagian paling atas dari program. Bagian ini mendeklarasikan file – file library yang diperlukan dalam eksekusi sebuah program. Tetapi, jika memerlukan library lain, maka perlu ditambahkan secara manual seperti:


```

Using System;
Using System.Collection.Generic;
Using System.Linq;
Using System.Text;

```
2. *Namespace* statement, hal ini berupa deklarasi nama project yang dibuat.
3. *Class* statement, hal ini merupakan salah satu bagian penting dalam bahasa pemrograman C#. Awal dan akhir satu class ditandai dengan tanda kurung kurawal ({...}).

4. Main method, hal ini salah satu jenis method khusus, dimana merupakan statement yang pertama kali dieksekusi oleh .NET framework pada saat eksekusi program.
5. Code comment, hal ini berupa sebuah keterangan berbentuk tulisan atau tanda tangan apapun yang berguna sebagai dokumentasi atau catatan yang ditambahkan pada program tetapi tidak dieksekusi sebagai bagian program. Keterangan atau tulisan tersebut disebut sebagai comment dengan menggunakan beberapa pilihan, yaitu :
 - a. Menggunakan tanda //
 - b. Menggunakan tanda /*...*/
 - c. Menggunakan tanda ///
6. Console.WriteLine(), hal ini digunakan untuk menampilkan teks pada console berupa teks “Hello, World!” guna membuat sebuah karakter yang dianggap sebagai teks dengan tanda (“ ”).
7. Console.ReadLine, hal ini merupakan method untuk membaca teks yang dimaksud oleh user dan proses pembacaan tersebut dihentikan jika tombol ENTER ditekan.

Berikut merupakan sebuah hasil dari notasi bahasa C# :

Bahasa C++

Bahasa C++ atau yang sering disebut bahasa CPP (C Plus Plus) merupakan hal yang biasanya tidak dapat lepas dari Notasi bahasa C, sebagai bahasa pendahulunya, sehingga notasi ini disebut dengan istilah perluasan dari notasi C dengan tambahan fasilitas kelas (Class). Penciptaan notasi C++ sendiri diciptakan sekitar satu dekade setelah notasi bahasa awal, notasi C. Penciptanya bernama Bjarne Stroustrup dari Laboratorium Bell, AT&T, pada tahun 1983. Notasi C++ cukup kompatibel dan sesuai dengan bahasa pendahulunya. Pada awal mulanya Notasi C++ disebut “a better C”. Nama C++ sendiri diberikan oleh Rick Mascitti pada tahun 1983, yang berasal dari operator increment pada bahasa C.

Notasi C++ ini memiliki beberapa karakteristik seperti pada program Notasi C++ yaitu berupa sekumpulan fungsi yang menyeluruh. Bahkan bahasa program utamanya juga terlihat berbentuk fungsi, yaitu fungsi main(). Kode Notasi bahasa C++ bersifat case sensitive, yang artinya dapat membedakan antara huruf kapital dengan huruf kecil. Oleh karena itu seperti yang kita bahas pada bagian notasi bahasa C, kedua notasi ini merupakan notasi bahasa yang sangat populer dalam dunia pengembangan perangkat lunak. Kedua bahasa ini digolongkan ke dalam bahasa tingkat menengah (middle level language), namun juga menggolongkannya ke dalam bahasa tingkat tinggi. Bahasa C++ ini juga memiliki keistimewaan yaitu adalah bahasa ini mendukung pemrograman yang berarah pada objek atau yang

lebih sering disebut dengan sebuah padanan kata yang terdapat pada istilah Object Oriented Programming (OOP).

Bahasa C++ memiliki kerangka yang berbentuk seperti berikut :

```
#include <iostream.h>
//Prototype fungsi
tipe_data nama_fungsi(parameter1,parameter2,...);
//Fungsi utama
void main()
{
    statemen_yang_akan_dilakukan;
    ...
    return 0;
}
//Implementasi fungsi
tipe_data nama_fungsi(parameter1,parameter2,...)
{
    statemen_yang_akan_dilakukan;
    ...
}
```

Contoh :

```
#include<iostream.h>
int main()
{
    cout <<"Selamat menggunakan C++ dalam
Program Algoritma";
    return 0;
}
```

Struktur Dasar Algoritma Pemrograman

Seperti yang telah kita bahas sebelumnya bahwa Algoritma merupakan hal yang berisikan urutan suatu penyelesaian dari masalah. Nah, pada bagian tersebut merupakan suatu struktur

yang menjadi dasar pemecahan dari suatu masalah tersebut sekaligus menjadi pernyataan dari pemecahan masalahnya. Langkah-langkah itu dapat disusun dengan menggunakan beberapa cara, yaitu sebagai berikut :

1. Struktur Runtunan Aksi (Sequence):

Pada struktur runtunan ini biasanya digunakan untuk melakukan program yang di dalamnya terdapat pernyataannya sequential atau urutan yang terurut. Runtunan Aksi (Sequence) biasanya memuat satu pernyataan atau lebih dari. Setiap pernyataan yang ada nantinya akan dieksekusi dengan cara berurutan atau terurut dengan aturan yang telah ditetapkan, pengeksekusian tersebut yaitu merupakan sebuah instruksi atau pernyataan yang akan dijalankan setelah instruksi sebelumnya selesai dijalankan. Sebagai contoh, runtunan dapat ditemui pada algoritma Tukar_isi_bejana berikut ini. Hal itu dapat diselesaikan dengan langkah – langkah seperti berikut :

Langkah 1

Tuangkan isi bejana A ke dalam bejana C

Langkah 2

Tuangkan isi bejana B ke dalam bejana A

Langkah 3

Tuangkan isi bejana C ke dalam bejana B

2. Struktur Pemilihan Aksi (Selection):

Struktur Pemilihan Aksi atau Selection merupakan langkah yang digunakan untuk melakukan suatu program dengan menggunakan penyeleksian kondisi.

Namun, adakalanya bila sebuah aksi struktur pemilihan ini dibuat setelah kondisi tertentu yang sebelumnya terpenuhi. Misalnya, terdapat sebuah kendaraan seperti bus yang berada di perempatan atau persimpangan jalan yang disana terdapat lampu lalu lintas. Yang aturannya sudah pasti harus ditaati demi kenyamanan berkendara sesuai warna lampu merah, hijau, dan kuning. Kondisi ini dapat ditulis melalui notasi sebagai berikut :

*Jika nilai ujian > 50, maka
"keterangan = D"*

Notasi tersebut dapat diubah ke bentuk pernyataan pemilihan aksi dengan notasi sebagai berikut :

*if kondisi then
aksi*

maka, untuk pernyataan seperti contoh di atas dapat ditulis seperti :

*if nilai ujian > 50 then
"Keterangan = D"*

Strategi pemilihan kata if (jika) tersebut memberikan suatu kata berupa pilihan dengan keadaan bila kondisi atau persyaratannya tersebut terpenuhi. Selain itu strategi pemilihan kata lain ini juga paling banyak digunakan diantaranya adalah penyeleksian satu dari dua aksi yang sesuai dengan kondisi bentuk struktur yang diperlukan. Berdasarkan contoh yang telah dipaparkan tadi maka

didapatkan bentuk struktur seperti di bawah ini :

*if kondisi then
aksi 1
else
aksi 2*

Contohnya, maka ditentukan bilangan yang lebih terbesar dari dua buah bilangan bulat yaitu x & y. Pemilihan struktur selanjutnya adalah sebagai berikut :

*if y < x then
tertulis x bilangan yang lebih besar
lain
tertulis y bilangan yang lebih besar*

Apabila terdapat pilihan kondisi yang bisa dilaksanakan lebih banyak, sehingga dapat menggunakan cara pemilihan nested if atau yang dinamakan bersarang. Untuk contoh maka dapat diperhatikan seperti yang dipaparkan di bawah ini :

*if traffic light menunjukkan bagian merah then
stop
lain
if traffic light menunjukkan bagian kuning then
be care full
lain
jalan terus*

Contoh lain dari pemilihan nested if adalah dengan memilih bilangan yang lebih dari besar dari tiga buah bilangan dari x, y, z, dengan notasi berikut :

*if y < x lalu
if z < x lalu
tertulis x bilangan yang lebih besar*

lain

tertulis z bilangan lebih besar

lain

if $z < y$ lalu

tertulis y bilangan lebih besar

lain

tertulis z bilangan lebih besar

3. Struktur Perulangan Aksi (Iteration):

Struktur Perulangan merupakan langkah yang biasanya digunakan untuk melakukan suatu program dengan pernyataannya yang nantinya akan dilakukan berulang. Pada tipe perulangan aksi ini juga terdapat suatu kondisi yaitu apabila terdapat suatu peristiwa dimana programmer harus memprogram teks dengan kata yang berulang atau teks dilakukan lebih dari 1 kali dalam sebuah algoritma pemrograman, solusinya kita bisa menggunakan struktur ini. Contohnya, saat hendak membuat paragraf mengenai “Teknik Informatika” sebanyak 8 kali. Maka algoritma pemrogramannya dapat dibuat dengan menggunakan bentuk struktur pengulangan yaitu fordo seperti yang dipaparkan berikut ini :

untuk i dari 1 until 8 lakukan

tertulis Teknik Informatika

end for

Notasi i disini adalah pencacahan sebuah pengulangan yang nantinya akan melakukan iteration tersebut mulai dari pengulangan 1 sampai dengan pengulan 8. Selanjutnya,

Komputer akan melakukan program iteration sebanyak yang telah diatur tersebut sebelumnya. Umumnya, struktur iteration dapat dibuat dengan notasi sebelumnya dengan 1 sebagai a dan 8 sebagai b.

Contoh sebelumnya juga dapat berartinya, tindakan yang diprogram yaitu akan sebanyak hitungan pencacahan yang melakukan pengulangan yaitu pengulangan dari x sampai y sebanyak $x - y + 1$ kali. Dengan struktur iteration yang kedua yaitu ulangi - hingga. Ulangi berarti “repeat”, sedangkan hingga berarti “until”. Sehingga terdapat formasi notasi umum sebagai berikut :

ulangi

aksi

sampai kondisi

Formasi di atas memiliki arti yaitu tindakan yang nantinya akan melakukan pengulangan hingga atau sampai keadaan (syarat) sudah terpenuhi. Contoh lainnya, menurut data sejumlah mahasiswa yang melakukan pengambilan pada mata kuliah Algoritma Pemrograman I, akan dibuat perbedaan data alamat mahasiswa dengan Nomor Induk Mahasiswa 009 dari sebelumnya. Maka, bila ditelusuri algoritma yang mungkin dengan menggunakan struktur pengulangan yaitu:

pada data mahasiswa posisi I

jika data mahasiswa I memiliki mempunyai Nomor 009 lalu

ubahlah data alamat mahasiswa tersebut

lain

lihat data mahasiswa pada posisi II

*jika data mahasiswa kedua memiliki NIM 009
lalu*

*ubahlah data alamat mahasiswa tersebut
lain*

... begitu pula selanjutnya

Dari Algoritma Pemrograman yang dicontohkan tersebut, dapat dilihat sebuah pemeriksaan atau penyeleksian tentang apakah data mahasiswa pada posisi I, II memiliki Nomor Induk 009 yang diprogram selanjutnya dengan mengulangi formasi. Algoritma pemrograman harus berhenti ketika selesai mengerjakan sejumlah urutan notasi yang terbatas. Bila terjadi kondisi ini yang dihadapi programmer, dapat menggunakan penyelesaian notasi ulangi–sampai ini seperti yang telah dibahas sebelumnya mengenai struktur ini yaitu dengan bentuk Algoritma Pemrogramannya adalah sebagai berikut :

*jika data mahasiswa I NIM 009 lalu
ubahlah data alamat mahasiswanya
lalu
lihatlah data mahasiswa pada posisi selanjutnya
sampai datanya dengan NIM 009 telah ditemukan
atau semua data telah dilakukan pemeriksaan*

Selanjutnya terdapat notasi pengulangan dengan struktur pengulangan selama-lakukan. selama dengan berupa “while”, sedangkan lakukan dengan berupa “do”. Struktur ini dapat digunakan dengan notasi umum yaitu berupa :

*selama kondisi lakukan
tindakan
sampai selama yang ditentukan*

Contoh di atas berarti, while dengan keadaan atau dengan syarat melakukan pengulangan yang masih dapat terpenuhi atau benar sesuai aturan yang telah ditetapkan Algoritma Pemrograman, maka didapati aksi nantinya akan dilakukan seperti di atas. Hal itu didapati perbedaannya dengan struktur repeat-until, jika pada struktur repeat-until ini terdapat kondisi pengulangan yang nantinya akan dievaluasi atau program melakukan sebuah pengecekan after tindakan sebelumnya diprogram, sementara pada struktur selama-laksanakan ini terdapat situasi pengulangan yang berbeda maka akan dicek sebelum tindakan mengalami pemrograman atau pada bagian awalnya mengalami sebuah pengulangan program.

Pembahasan

Maka, dapat setelah apa yang telah dipaparkan dan yang dijelaskan tentang banyaknya Notasi Bahasa dan Struktur Dasar dari Algoritma Pemrograman di atas, setelah mempelajarinya saya menjadi lebih mengetahui mengenai notasi bahasa yang biasanya paling banyak digunakan oleh para programmer Algoritma. Walaupun dengan beberapa notasi bahasa yang berbeda beda tersebut pastinya tidak memiliki persamaan jika dibuat oleh masing masing programmer, tapi dengan notasi yang benar pasti komputer bisa melakukan program Algoritma tersebut dengan benar.

Dengan kondisi perkembangan zaman sekarang yang penuh dengan kecanggihan teknologi dan informasi apalagi menggunakan

elektronik, maka hal ini menuntut masyarakat untuk terus belajar mengikuti arus tersebut. Terlebih lagi di dunia pendidikan, sehingga kita harus terus menerus mengikuti kecanggihan zaman tersebut. Namun, selebih dari itu masih banyak lagi notasi bahasa pemrograman yang digunakan di luar sana seiring berkembangnya zaman.

Gambar Pengguna Digital tahun 2021

Dari data di atas dapat dilihat bahwa penggunaan digital merupakan hal yang paling banyak digunakan oleh masyarakat sehingga jika Algoritma Pemrograman ini dilakukan maka merupakan sangat tepat dan disarankan untuk dilakukan terlebih lagi di zaman kecanggihan teknologi sekarang.

KESIMPULAN

Algoritma adalah urutan atau langkah - langkah yang logis mengenai penyelesaian yang masalahnya disusun secara sistematis dan logis tanpa menghilangkan makna kebenarannya dengan dapat disusun dengan kata yang tidak baku sehingga Algoritma ini dapat disusun dengan kata yang lebih santai seperti yang digunakan dalam kegiatan sehari hari.

Terdapat perbedaan antara kata “Algoritma” dengan kata “Program” yaitu jika “Algoritma”

ditambahkan dengan adanya “Struktur Data” akan menjadi suatu “Program” yang biasa digunakan dalam program komputer. Sehingga “Program” adalah kumpulan beberapa pernyataan yang ada pada komputer, sedangkan metode dan tahapan sistematis di dalam program tersebut adalah algoritma.

Dalam membuat Algoritma Pemrograman membutuhkan notasi bahasa dan juga struktur dasar yang menjadi ketetapanannya sehingga Algoritma Pemrograman dapat dinilai benar dan jauh dari kata kesalahan.

Dari sekian banyak notasi bahasa yang digunakan dalam bahasa pemrograman seperti notasi dengan menggunakan kalimat deskriptif, notasi bahasa Flow Chart, notasi bahasa Pseudo code, notasi bahasa Pascal, notasi bahasa C, C#, dan C++. Namun, masih banyak lagi notasi bahasa yang digunakan di luar sana.

Struktur Dasar Algoritma Pemrograman merupakan hal yang menjadi dasar dari pemrograman itu sendiri. Struktur dasar Algoritma ini yaitu runtunan, pemilihan, dan perulangan dari masing masing aksi atau bisa juga dengan cara kombinasi.

DAFTAR PUSTAKA

- Abidin Jaenal, dkk. 2021. Pelatihan Dasar - Dasar Algoritma dan Pemrograman untuk Membangkitkan Minat Siswa - Siswi SMK Pada Dunia Pemrograman. Dalam Jurnal Teknologi Sosial untuk Komunikasi. Vol. 2, No. 2. 54 – 62
- Allen, dkk. 2020. Pengaruh Penerapan Algoritma

- Terhadap Pembelajaran Pemrograman Komputer. Dalam Jurnal Inovasi Pendidikan Matematika. Vol 2, No 2
- Annisa, dkk. 2020. Prediksi Nilai Akademik Mahasiswa Menggunakan Algoritma Naive Bayes. Dalam Jurnal Sains dan Teknologi Vol. 9, No. 1
- Andi & Nurjannah. 2021. Pengembangan Modul Pembelajaran Interaktif (Mobelin) untuk Meningkatkan Mutu Pembelajaran Algoritma dan Pemrograman. Dalam Jurnal Benjamin Mutu. Vol. 7, No. 2. 145 - 146
- Anonim. 2015. Pemrograman Algoritma Struktur Data. Dalam Jurnal UMB. 2 – 15
- Dwi, Novian & Harits. 2020. Pengembangan Sistem Kuis Algoritma Pemrograman Berbasis Web. Dalam Jurnal Belantika Pendidikan Vol.3 (2), 66-74
- Gun. 2017. Pembelajaran Dasar Algoritma dan Pemrograman Menggunakan El-Goritma Berbasis Web. Dalam Jurnal Teknik Mesin. Vol. 06
- Hadiono, Kristophorus & Ridwan S. 2005. Program Database MySQL dengan Bahasa Pascal di Linux. Penerbit Andi. 17 – 19
- Hakim, Lukman. 2018. Bahasa Pemrograman (C# dan Emgucv). Penerbit Budi Utama. 11 – 19
- Heriyanto, Tedi & Eko B. 2005. Tool Pemrograman Bahasa C/C++ di GNU/Linux. Penerbit Andi. 5 – 10
- Mulyono, Budi. 2012. Algoritma Pemrograman I. UNSRI. 1 – 20
- Permana, Budi. 2007. Algoritma Pemrograman Dalam Bahasa Pascal. 2 – 5
- Prestiliano, Jasson. 2005. Petualangan Bahasa C (Implementasi Pemrograman Bahasa C dalam Berbagai Bidang). Penerbit Andi. 2 – 9
- Yuhendra. 2013. Algoritma dan Pemrograman (Aplikasi Bahasa Pemrograman PASCAL). ITP. 14 – 50