STRUKTUR DASAR PENULISAN ALGORITMA PEMROGRAMAN

Nur Sofia

Program Studi Pendidikan Matematika

Fakultas Ilmu Tarbiyah Dan Keguruan Universitas Islam Negeri Sumatera Utara

Email nursofia19628@gmail.com

Abstrak

Algoritma merupakan akar dari sebuah sistem yang terbentuk dalam dunia pemrograman atau susunan langkah – langkah sistematis dan logis dalam pemecahan masalah, pengetahuan tentang Struktur Dasar Algoritma Pemrograman nantinya sangat berguna dalam melakukan suatu pemrograman., terutama bagi orang yang sering merancang atau yang menggunakan algoritma maka stuktur dasar penulisan algoritma pemrograman ini perlu di ketahui dan juga di kuasai Melalui serangkaian langkah-langkah secara logis dan sistematis, algoritma ini juga mampu menyelesaikan suatu permasalahan yaitu masalah yang dapat disusun secara sistematis dan logis. Struktur Dasar Algoritma Pemrograman nantinya sangat berguna dalam melakukan pemrograman dalam algoritma. algoritma berisi langkah-langkah penyelesaian sebuah masalah., langkah – langkah tersebut di sebut dengan struktur dasar, Srtuktur tersebut terbagi atas tiga bagian yaitu Langkah-langkah itu dapat diproses secara beruntun lurus ke bawah. struktur atau di namakan struktur runtunan(*squenstial*), atau dapat berpindah ke langkah tertentu jika memenuhi syarat tertentu atau yang di sebut struktur pemilihan atau seleksi(*selection*), atau dapat berulang sesuai counter yang dikehendaki atau yang di sebut dengan struktur perulangan(*looping*).

Kata Kunci : struktur dasar, algoritma, pemrograman

PENDAHULUAN

Apabila kita mendengar kata “ algoritma ” sehingga kita bakal langsung beranggapan pada arah istilah matematika yang tentu saja apabila hendak menerapkan proses hitung– hitungan dalam matematika. namun, sesungguhnya algoritma itu memiliki makna yang lebih dari pada itu ataupun bahkan berbeda. Definisi dari Algoritma itu sendiri ialah merupakan urutan langkah- langkah pada penyelesaian permasalahan yang disusun secara sistematis serta logis. atau bisa juga di artikan bahwa algoritma itu adalah suatu urutan penyelesaian suatu permasalahan yang di susun secara sistematis dengan menggunakan Bahasa yang logis untuk menyelesaikan suatu permasalahan suatu kasus. dari penjabaran tersebut bahwa algoritma itu memiliki arti yaitu jika langkah langkah suatu permasalahan yang hendak dipecahkan dan berbeda dengan definisi matematika yang bisa menyelesaikan soal dari suatu buku, maka algoritma yaitu suatu pemecahan masalah dengan cara menggunakan rumus atau computer, Dalam ilmu matematika dan komputer, pengertian algoritma yakni prosedur dari sebagian langkah demi langkah untuk suatu penghitungan. algoritma juga dapat di pakai sebagai penghitungan, penalaran otomatis dan juga sebagai pemrosesan informasi. adapun pengertian dari algoritma adalah suatu tata atau pun metode yang sangat efisien apabila di gunakan sebagai suatu pemecahan masalah yang telah di jabarkan dengan baik untuk menghitung suatu fungsi.

Berikutnya apabila “ Algoritma ” ditambahkan dengan adanya “ Struktur Informasi atau data ” akan menjadi sesuatu “ Program ” yang biasa digunakan dalam program komputer. Sehingga bisa ditarik suatu kesimpulan jika “ Program ” adalah kumpulan dari sebagian statment yang ada pada komputer, sementara itu prosedur dan tahapan-tahapan sistematisnya yang terdapat di dalam program tersebut merupakan algoritma.

Algoritma Pemrograman merupakan urutan langkah- langkah dalam menuntaskan permasalahan memakai Bahasa pemrograman serta disusun secara sistematis serta logis. Kata logis ialah kata kunci dalam algoritma. Langkah- langkah dalam algoritma harus logis serta mesti bisa didetetapkan nilainya, apakah

salah ataupun benar. Supaya program bisa dilaksanakan oleh komputer, program tersebut wajib ditulis dalam suatu bahasa yang dipahami oleh komputer. Bahasa komputer yang di gunakan untuk menulis suatu program computer yaitu di yang namanya Bahasa pemrograman. Urutan langkah- langkah yang sistematis untuk menyelesaikan suatu permasalahan dinamakan algoritma.

Supaya program bisa dilaksanakan oleh komputer, program tersebut mesti ditulis dalam suatu bahasa yang dimengerti oleh komputer. Bahasa komputer yang di pakai untuk menulis program tersebut dinamakan Bahasa pemrograman. Urutan langkah- langkah yang sistematis untuk menyelesaikan sesuatu permasalahan dinamakan algoritma.

Dalam sesuatu algoritma langkah- langkah buat penyelesaian perkaranya terdapat sebagian metode ataupun langkah- langkahnya, ada pula Langkah- langkah itu bisa diproses secara beruntun lurus ke bawah(struktur runtunan), ataupun bisa berpindah ke langkah tertentu bila penuhi syarat tertentu(struktur pemilihan), ataupun bisa berulang sesuai counter yang dikehendaki(struktur perulangan). Ketiga jenis langkah-langkah tersebut membentuk konstruktif suatu algoritma.di dalam algoritma pemrograman juga terdapat suatu struktur yang menjadi dasar dari sebuah pemikiran algoritma pemrograman itu sendiri. Struktur dasar ini pastinya juga disusun dengan terurut guna menyusun Algoritma Pemrograman yang benar sesuai dengan kaidahnya.

Algoritma dapat dituliskan dengan banyak metode, mulai dari mengenakan bahasa alami maupun biasa yang dapat digunakan sehari- hari, simbol grafik bagan alir(flowchart), hingga memakai bahasa pemrograman semacam bahasa C maupun C++. Penyusunan algoritma tidak terikat pada suatu ketentuan tertentu, namun harus jelas maksudnya dari sesuatu penyusunan algoritma untuk tiap langkah- langkah algoritmanya. Notasi algoritma bukan merupakan notasi bahasa pemrograman, tetapi notasi ini dapat diterjemahkan kedalam bermacam bahasa pemrograman. Algoritma dapat dinotasikan dalam beberapa cara ataupun metode.

METTODE

Metode yang digunakan pada artikel ini adalah dengan studi literatur yaitu dengan dengan mengunjungi beberapa halaman atau artikel dari world wide web dan juga dari buku juga jurnal sebagai referensi, yang dilakukan dengan cara pemaparan yaitu terkait dengan judul struktur dasar dan notasi penulisan algoritma pemrograman. Teknik untuk analisis data apa yang dilakukan adalah dengan melakukan metode penyaringan (apply filter) terkait dengan judul yaitu struktur dasar dan notasi penulisan algoritma pemrograman. Karakteristik pada judulnya, setidaknya terdapat beberapa kata yang ada kata struktur dasar dan juga notasi penulisan algoritma pemrograman. Menurut penulis dengan cara yang seperti itu maka penulis dapat lebih mudah dalam menyaring artikel dan juga dapat mencocokkan pokok bahasan yang menjadi inti permasalahannya yang sedang di bahas, yaitu penulis dapat menyajikan mulai dari pengertian algoritma pemrograman sampai bagaimana struktur dasar dari algoritma tersebut dan juga bagaimana cara penulisannya.

PEMBAHASAN

Istilah algoritma berasal dari kata “algorithm” yang merujuk pada seseorang pakar matematika muslim yang hidup di abad ke- 19 bernama Ibnu Al- Khwarizmi. Pada perkembangannya, konsep pemikiran ia dalam menguraikan serta memecahkan perkara secara logis serta matematis diterapkan sebagai suatu prosedur algoritma dalam proses suatu kerja komputer.

Algoritma merupakan urutan langkah- langkah yang logis dalam penyelesaian permasalahan yang disusun secara sistematis. Referensi lain juga mengatakan algoritma merupakan sesuatu prosedur yang menggambarkan urutan langkah- langkah yang berintegrasi, ataupun sesuatu prosedur khusus yang digunakan dalam buat menyelesaikan sesuatu permasalahan yang nyata.

Algoritma berisi langkah-langkah penyelesaian sebuah masalah, langkah langkah tersebut di sebut dengan struktur dasar. Algoritma memiliki struktur untuk dapat memproses instruksi dalam sebuah program. Srtuktur tersebut terbagi atas tiga bagian yaitu Langkah- langkah itu bisa diproses secara beruntun lurus ke bawah

atau yang di namakan struktur runtutan (squential),atau juga dapat berpindah dari langkah satu ke langkah tertentu jika memenuhi ketentuan atau juga syarat tertentu,yang selanjutnya yaitu struktur pemilihan atau pilih (selection),dan yang terakhir yatu struktur perulangan (looping) atau dapat di sebut sebagai struktur dapat berulang sesuai counter yang di kehendaknya. Berikut ulasan atau pembahasan lengkapnya.

Struktur dasar algoritma

1. Struktur runtutan (squential / sequence)

Struktur dasar Runtutan merupakan struktur algoritma yang paling bawah atau pun mendasar yang berisi rangkaian instruksi yang diproses secara sekuensial, satu persatu, mulai dari instruksi awal hingga instruksi terakhir serta juga suatu runtutan terdiri dari satu ataupun lebih pernyataan. Tiap pernyataan akan dieksekusi secara berurutan sesuai dengan urutan penulisannya, yaitu suatu instruksi(pernyataan) yang hendak dijalankan setelah instruksi sebelumnya selesai dijalankan. ataupun dengan kata lain suatu algoritma dibentuk dengan langkah- langkah(instruksi/ perintah) dikerjakan secara berurutan dari atas ke bawah sesuai dengan urutan penulisannya, tidak boleh melompati satu langkah perintah pun. Misalnya dalam suatu kasus algoritma yaitu ada 20 langkah, maka seluruh langkah tersebut harus dikerjakan secara berurutan mulai dari langkah 1 sampai dengan pada langkah 20 tanpa melewati satu langkah pun.

Cara kerja sekuensial juga nantinya berlaku dalam bahasa pemrograman, pada saat instruksi bahasa pemrograman yang kita tulis diproses oleh komputer, sehingga komputer akan memproses serta menterjemahkan baris demi baris intruksi- instruksi bahasa pemrograman tersebut secara beruntun mulai dari awal sampai akhir diawali dari instruksi pada baris awal sampai baris akhir.

Dengan struktur runtutan ini, untuk membuat suatu algoritma nya maka kita harus dapat menganalisa dan juga menentukan intruksi mana yang harus ditulis

lebih awal dan dan mana yang di tulis di akhir sampai seterusnya tanpa melewati satu langkah pun.

Menurut pendapat Goldschlager dkk dalam bukunya yaitu Algoritma dan Pemrograman yang ditulis Rinaldi Munir dkk, pada dasarnya algoritma ialah runtunan (sequence) satu maupun lebih instruksi, yang berarti bahwa:

- Masing- masing instruksi dikerjakan satu persatu;
- Masing- masing instruksi dikerjakan cuma sekali, tidak ada yang diulang;
- Urutan instruksi yang dikerjakan pemroses sama dengan urutan instruksi sebagaimana yang tertulis di dalam suatu teks algoritmanya;
- Instruksi terakhir menggambarkan akhir algoritma.

Contoh struktur algoritma runtutan: yaitu menjumlahkan 2 bilangan

Langkah 1

Masukkan nilai bilangan kesatu (Bil 1)

Masukkan nilai bilangan kedua (Bil 2)

Hitung Hasil = Bil 1 + Bil 2

Cetak hasil penjumlahan (Hasil)

2. Struktur pemilihan atau seleksi(*selection*)

Struktur pemilihan ialah suatu struktur program yang menerapkan suatu proses pengujian untuk mengambil sebuah keputusan apakah suatu hasil keputusan tersebut dapat di proses atau tidak. Pada struktur pemilihan tidak setiap baris program hendak dikerjakan, Baris program yang dikerjakan cuma yang memenuhi syarat saja. instruksi akan diproses ataupun tidak. Pengujian keadaan ini dilakukan untuk memilih salah satu dari sekian banyak alternatif yang ada.

Pada kenyataannya banyak suatu algoritma setidaknya akan menggunakan proses seleksi (*selection*) pada intruksi-intruksi pada saat dalam proses algoritma, instruksi seleksi yang digunakan apabila menemukan/memiliki suatu kasus bercabang dua atau lebih maka alternatif untuk penyelesaian keputusannya menggunakan struktur pemilihan.

Pada struktur pemilihan maupun percabangan, terdapat pemeriksaan keadaan/ ketentuan yang wajib dipenuhi, yang kemudian akan memilih perintah apa yang bakal dicoba bila ketentuan tersebut dipenuhi. Maka suatu Perintah tersebut tidak lagi dikerjakan secara beruntun seperti pada struktur runtutan, namun di kerjakan berdasarkan ketentuan yang wajib dipenuhi. Berikut ini adalah Notasi algoritma untuk struktur pemilihan menggunakan 2 konstruksi yaitu IF– THEN dan CASE berikut ini:

1. Konstruksi IF – THEN

Konstruksi IF– THEN dipisah lagi menjadi 3 yakni:

a. Satu aksi

Ialah Apabila suatu keadaan yang dipilih terpenuhi maka aksi ataupun statement yang akan mengikuti“ then” yang hendak diproses. Dan jika sebaliknya hingga tidak diminta aksi apapun.

Kontribusi ini berbentuk :

If <kondisi> then

aksi

End if

Contoh :

Tuliskan suatu algoritma untuk menampilkan penjelasan“ bilangan bulat positif” bila bilangan bulat yang diinputkan yaitu bilangan bulat positif.

Seperti yang sudah kita diketahui bahwa bilangan positif ialah bilangan yang nilainya lebih besar sama dengan nol (≥ 0). Jadi bila bilangan bulat yang diinputkan= 0 maka akan muncul penjelasan“ bilangan bulat positif”, namun bila < 0 tidak diminta aksi apapun. adapun struktur penyelesaiannya ialah:

```
if bil_bulat  $\geq$  0 then  
write “bilangan bulat positif”  
end if
```

b. Dua aksi

Yaitu Apabila keadaan terpenuhi sehingga aksi ataupun statement 1 yang akan diproses, jika tidak maka statement 2 yang bakal diproses.

Konstruksi ini berbentuk :

```
If < kondisi >  
then < statement 1 >  
else < statement 2 >
```

contoh:

Tuliskan algoritma untuk menunjukkan penjelasan“ bilangan genap” ataupun“ bilangan ganjil”. Bila bilangan bulat positif yang diinputkan habis dibagi 2, sehingga akan muncul“ bilangan genap”, tetapi bila sebaliknya muncul“ bilangan ganjil”.

Dalam dunia pemrograman diketahui operator modulus yang disingkat mod. Operator tersebut berperan guna mencari sisa pembagian.

struktur penyelesaiannya ialah:

```
if (bil mod 2) = 0 then
```


write “bilangan genap”

else

write “bilangan ganjil”

c. Tiga aksi atau lebih

Konstruksi IF– THEN pada 3 aksi ataupun lebih, bisa pula disebut IF Bersarang ataupun IF Bertingkat. Dimana ada perintah IF di dalam IF(nested if). Ilustrasi konstruksi tersebut bisa dilihat pada gambar di bawah ini:

Gambar 3.1. Konstruksi IF – THEN Bersarang

- Bila keadaan 1 terpenuhi sehingga cek kondisi 2: bila keadaan 2 terpenuhi sehingga kerjakan aksi ataupun statement 1, sedangkan bila keadaan 2 tidak terpenuhi maka kerjakan aksi ataupun statement 2.
- Apabila keadaan 1 tidak terpenuhi maka kerjakan statement 3

Contoh :

Coba Tuliskan suatu algoritma yang di gunakan untuk menghitung total pembayaran suatu barang pada suatu psar swalayan, dengan menggunakn syarat yaitu sebagai berikut:

- Bila total pembelian=100000, sehingga mendapatkan diskon dengan ketentuan ataupun kondisi sebagai berikut:
 - Bila pembeli mempunyai kartu anggota, sehingga mendapatkan diskon 20%,
 - bila bukan anggota ataupun member sehingga mendapatkan diskon 10%
- Bila total pembelian < 100000, sehingga tidak mendapatkan diskon (diskon=0)

Berikut adalah strukturnya :

If beli >= 100000 then

{ If pembeli = member then

*diskon = beli * 0.20*

else if

*diskon = beli * 0.10 }*

else diskon = 0

2. Konstruksi CASE

Konstruksi CASE digunakan untuk permasalahan dengan 2 permasalahan ataupun lebih. Konstruksi ini bisa menyederhanakan perintah if- then- else yang bertingkat- tingkat maupun bersarang untuk permasalahan tertentu. Statement CASE dibagi menjadi 2 struktur:

a.CASE – OF

b.CASE – OF – ELSE

Coba kita perhatikan perbandingan ataupun perbedaan dari struktur CASE– OF dengan CASE– OF– ELSE pada gambar di bawah ini:

Dari gambar di atas maka penjelasannya dapat dilihat pada penjelasan di bawah :

- Dalam suatu konstruksi case– of, variabel yang akan diinputkan dengan nama ‘ grade’ akan dibaca(dicek): bila ‘ A’ maka dicetak keterangan ‘ Sangat Baik’, bila ‘ B’ maka dicetak keterangan ‘ Baik’, serta seterusnya hingga ‘ E’, kemudian berakhir.
- Dalam suatu konstruksi case– of– else, variabel yang akan diinputkan dengan nama ‘ opsi’ akan dibaca bila tanda seperti ini (= 1), maka dilakukan statement 1, dan jika 2 dilakukan aksi ataupun statement 2. tetapi jika opsi yang diinputkan tidak hanya 1 serta 2, maka akan dilakukan statement 3.

Contoh lain case – of dan case – of – else :

a. Konversi angka menjadi nama hari memakai konstruksi case – of

Case angka of

1 : write ('Senin')

2 : write ('Selasa')

3 : write ('Rabu')

4 : write ('Kamis')

5 : write ('Jumat')

6 : write ('Sabtu')

7 : write ('Minggu')

End;

b. Konversi grade (dalam bentuk huruf) menjadi suatu keterangan pencapaian menggunakan konstruksi case –of :

case (grade) of

'A' :

Begin

writeln('Sangat Memuaskan');

writeln('Pertahankan!');

end;

'B' :

Begin

writeln('Memuaskan');

writeln('Tingkatkan');

end;

'C','D' :

Begin

writeln('Cukup Baik');

writeln('Belajar lebih giat lagi ya');

end;

'E' :

Begin

Writeln ('Maaf, anda tidak lolos');

3. struktur perulangan(looping).

Perulangan ialah struktur dimana ada proses pengulangan perintah yang sama sebanyak n kali. Struktur ini ialah salah satu kelebihan yang dipunyai oleh mesin komputer. Sebagai contoh untuk mencetak suatu teks“ Algoritma Pemrograman” sebanyak 100 kali pada layar monitor, cuma dibutuhkan beberapa baris perintah meggunakan metode ataupun

struktur perulangan tersebut. Tanpa mesti menuliskan perintah yang sama sebanyak 100 kali.

Struktur pengulangan disebut juga dengan kalang(loop), dan juga bagian algoritma yang diulang(aksi) maka dapat dinamakan badan kalang(loop body). algoritma sendiri untuk menanggulangi permasalahan pengulangan data, mempunyai intruksi tertentu, dengan intruksi tersebut pengulangan akan lebih gampang ditulis secara pendek serta instan dari pada mesti di tulis satu- persatu.

Algoritma menggunakan suatu struktur perulangan banyak dijumpai dalam suatu kehidupan sehari- hari, contohnya saja untuk mencetak suatu surat undangan yang jumlahnya banyak sampai ratusan dengan format yang sama, menunjukkan beberapa deret bilangan ganjil ataupun bilangan genap, ataupun deret bilangan prima, ataupun deret bilangan dengan rumus deret tertentu, mengambil beberapa uang di ATM dengan pecahan tertentu. Ada 2 tipe perulangan yang diketahui dalam algoritma, ialah:

1. Unconditional looping ialah perulangan yang tidak menyertakan keadaan tertentu sebagai ketentuan terbentuknya perulangan perintah, contoh perulangan dengan FOR- DO.
2. Conditional looping ialah perulangan dengan terdapatnya keadaan ataupun ketentuan yang wajib dipenuhi, contoh perulangan dengan WHILE- DO serta REPEAT- UNTIL

a. Konstruksi perintah FOR – DO

Konstruksi ini menetapkan jumlah perulangan saat sebelum perulangan tersebut dicoba. Jumlah perulangan sudah didefinisikan pada awal deklarasi perintah ini. Berikut format penyusunan perintahnya dalam bentuk notasi pseudo- code:

for [variabel pengulang] ← nilai_awal to nilai_akhir do

statement 1

statement 2

...
end for

contoh perintah FOR – DO untuk mencetak teks“ Algoritma Pemrograman” sebanyak 100 kali, dimana i merupakan variable pengulangnya:

for i $\leftarrow$ 1 to 100 do

write (“Algoritma Pemrograman”)

end for

Konstruksi FOR- DO juga bisa digunakan buat perulangan secara menurun (descending), yaitu memakai down to.

Contoh mencetak deret bilangan 5 4 3 2 1, sehingga format penulisan perintahnya sebagai berikut :

For x $\leftarrow$ 5 down to 1 do

Write (x)

End for

Variabel pengulang x sekalian sebagai nilai yang hendak dicetak, perhatikan perintah pada baris kedua di atas ialah write(x).

b. Konstruksi perintah WHILE - DO

Konstruksi ini juga melaksanakan suatu pengecekan kondisi perulangan terlebih dulu saat sebelum melanjutkan ke perintah selanjutnya, apakah telah terpenuhi(true) dan sesuai yang disyaratkan ataupun tidak. Berikut format penyusunan perintahnya:

while[kondisi] do

statement 1

statement 2

.....

end while

contoh perintah WHILE – DO untuk mencetak teks “ Algoritma Pemrograman” sebanyak 100 kali :

i = 1

while i <= 100 do

write (“Algoritma Pemrograman”)

i = i + 1

end while

Pada awal pernyataan didetapkan nilai awal variabel $i = 1$, perintah selanjutnya mengecek kondisi apakah nilai $i \leq 100$, sepanjang kondisi tersebut terpenuhi hingga akan dicetak teks yang diartikan. Variabel pengulang dinaikkan satu persatu($i = i+1$), serta kembali ke perintah pengecekan keadaan, apakah nilai i yang terbaru masih ≤ 100 , bila terpenuhi hingga teks “ Algoritma Pemrograman” akan terus dicetak, hingga dengan keadaan menjadi tidak terpenuhi ialah $i=100$, hingga perintah perulangan akan dihentikan.

c. Konstruksi perintah REPEAT – UNTIL

Konstruksi ini agak berbeda dengan WHILE– DO yang melakukan pengecekan keadaan di awal perulangan. Perulangan REPEAT– UNTIL dilakukan terlebih dulu tanpa pengecekan kondisi, serta akan dihentikan hingga kondisi jadi terpenuhi(true).

Berikut format penulisan perintahnya :

Repeat


```

statement 1

statement 2

... ..

until [kondisi]

```

contoh perintah REPEAT- UNTIL untuk mencetak teks“ Algoritma Pemrograman” sebanyak 100 kali:

```

i = 1

repeat

write (“Algoritma Pemrograman”)

i = i + 1

until i > 100

```

Dalam struktur perulangan pula ada struktur perulangan bersarang, dimana ada perulangan di dalam perulangan, contoh ada For- Do di dalam For- Do pada permasalahan membaca data matrik. Matrik ataupun array 2 dimensi yang terdiri dari baris dan juga kolom, dimana pembacaan data diawali dari baris ke- 1 terus lanjut ke proses pembacaan data pada masing- masing kolom yang ada dalam baris ke- 1 tersebut. Perihal itu dilakukan berulang hingga 1 baris habis terbaca, lanjut ke baris ke- 2. Pada baris ke- 2 dilakukan pembacaan data pada kolom baris tersebut satu persatu, serta begitu seterusnya proses berulang hingga seluruh informasi pada seluruh baris serta kolom habis terbaca.

Struktur algoritma

Agar algoritma bisa ditulis lebih teratur sehingga hendaknya dibagi ke dalam beberapa bagian. Salah struktur yang kerap dijadikan patokan yaitu sebagai berikut:

a. **Bagian Kepala (Header)**

Bagian Kepala algoritma merupakan bagian yang terdiri dari nama algoritma serta uraian algoritma. Ketentuan pemberian nama algoritma

mengacu pada ketentuan pemberian nama pengenalan. Nama algoritma sebaiknya pendek, tetapi mencerminkan isi algoritma secara totalitas. Bagian uraian algoritma berisi uraian mengenai hal-hal yang dilakukan oleh algoritma secara pendek. Contoh kepala algoritma bisa dilihat pada gambar di bawah ini.

Algoritma LUAS_SEGITIGA

{Menghitung luas segitiga berdasarkan panjang alas dan tinggi segitiga yang diinput user. Luas segitiga dapat diperoleh dari rumus $\text{Luas} = 0,5 \times \text{alas} \times \text{tinggi}$ }

Algoritma Menentukan_Bilangan_Prima

{Menentukan apakah suatu bilangan bulat yang diinputkan oleh user merupakan bilangan prima atau komposit. Algoritma akan mencetak kata "prima" apabila bilangan tersebut adalah bilangan prima, dan sebaliknya akan mencetak kata "komposit" bila bilangan tersebut bukan bilangan prima}

b. Bagian Deklarasi

Bagian deklarasi berisikan seluruh nama pengenalan yang dipakai di dalam algoritma. Nama tersebut bisa berbentuk nama tetapan(konstanta), nama peubah(variabel), nama jenis, nama prosedur, serta nama fungsi. Contohnya bisa dilihat di pada gambar bawah ini.

DEKLARASI

```
02| {nama tetapan}
03| const Npeg = 100 {jumlah pegawai}
04| const phi = 3.14 {nilai phi}
05| {nama tipe}
06| type TTitik : record {tipe koordinat bidang kartesius}
07| < x,
08| y : integer
09| >
10| {nama peubah}
11| c : char {karakter yang dibaca}
12| Q : TTitik {titik dalam koordinat kartesius}
13| ketemu : boolean {keadaan hasil pencarian}
14|
15| function IsPrima(input x:integer) □ boolean
16| {mengembalikan nilai true bila x adalah prima, atau false
  bila x adalah komposit}
17|
18| procedure Tukar(input/output a,b : integer)
19| {mempertukarkan isi variabel a dan b}
```

c. Bagian Deskripsi

Bagian ini memuat langkah- langkah penyelesaian permasalahan, termasuk sebagian perintah yaitu seperti dalam membaca data, tampilkan, ulangi, serta sebagainya. Berikut ini merupakan contoh suatu algoritma yang mengikuti struktur tersebut diatas:

Algoritma Nilai_Rata (menghitung nilai rata sejumlah bilangan yang dimasukkan lewat keyboard)
Deklarasi integer x, N, k, jumlah; real nilai_rata; Deskripsi { masukkan jumlah data } read (N); k $\leftarrow$ 1; jumlah $\leftarrow$ 0; while (k <= N) do { baca data } read (x); jumlah $\leftarrow$ jumlah + x; k $\leftarrow$ k + 1; endwhile { hitung nilai rata } nilai_rata $\leftarrow$ jumlah / N; write (nilai_rata);

Struktur dasar penulisan algoritma menggunakan notasi algoritma flowchart dan pseudo-code

1. Struktur dasar algoritma dalam bentuk flowchart

Struktur dasar dari algoritma juga bisa dinotasikan dalam wujud flowchart, Flowchart merupakan bagan- bagan yang mempunyai bentuk menggambarkan langkah- langkah penyelesaian sesuatu permasalahan, Merupakan metode penyajian dari sesuatu algoritma.

a. flowchart dengan struktur runtutan

Contoh 1: menampilkan seluruh proses dilakukan secara berentetan dari atas hingga ke bawah hingga berakhir. Tidak terdapat percabangan maupun pengulangan di dalamnya.

Misalnya : suatu permasalahan untuk mencari suatu luas dari suatu lingkaran yang Di mulai dengan simbol terminal‘ mulai’, setelah itu symbol untuk memastikan harga mula ialah $\phi = 3.14$, dilanjutkan memasukkan variabel r (jari- jari), serta dihitung dengan rumus $L = \phi * r * r$, setelah itu mencetak hasilnya(yang disimpan dalam variabel L), terakhir terminal berakhir.

Gambar Flowchart Dengan Struktur Runtutan

b. Flowchart dengan struktur pemilihan

Contoh flowchart dengan struktur pemilihan ataupun percabangan yang pasti memiliki aliran bercabang untuk memilih ataupun mengecek keadaan tertentu. contoh nya ialah flowchart yang menggambarkan permasalahan menunjukkan penjelasan“ Lulus” ataupun“ Gagal” berdasarkan nilai yang dimasukkan.maka flowchart nya yaitu sebagai berikut :

Gambar flowchart struktur pemilihan

Flowchart di atas menggambarkan alur pengecekan nilai yang diinputkan. Bila nilai yang diinputkan=70 hingga dicetak keterangan“ Lulus”, bila nilai; 70 sehingga dicetak keterangan“ Gagal”.

c. Flowchart dengan struktur perulangan

jika struktur perulangan ialah struktur dimana ada proses pengulangan perintah yang sama sebanyak n kali. Beberapa permasalahan yang bisa dituntaskan memakai struktur ini antara lain menampilkan deret bilangan.

Contohnya yaitu Flowchart menggambarkan aliran langkah-langkah untuk menampilkan deret bilangan bulat 0 1 2 3 4 5.

Maka gambarnya flowchart nya yaitu :

Gambar Flowchart struktur perulangan

2. Struktur dasar algoritma dalam bentuk pseudocode

a. Pseudocode dengan struktur runtutan

Contoh pseudocod, mencari kecepatan serta percepatan maka cara pengerjaannya yaitu seperti dibawah ini:

<i>Pseudocode cari kecepatan dan percepatan</i>	
1	Mulai
2	Deklarasi variabel: v, a, s, t
2	Input/masukkan nilai: s, a
3	$v \leftarrow s/t$
4	$a \leftarrow v/t$
5	Tampilkan nilai kecepatan: v
6	Tampilkan nilai percepatan: a
7	Selesai

b. Pseudocode dengan struktur pemilihan

Contoh pseudocode dengan kasus Ada dua angka yang akan diseleksi dengan membandingkan keduanya mana yang terbesar/sama dengan atau terkecil. Misal nilai $a = 6$ dan nilai $b = 5$, maka seleksinya adalah akan tercetak “Nilai $a \geq b$ ” dan jika sebaliknya maka keputusan yang lain adalah tercetak “Nilai $a < b$ ”.

<i>Pseudocode menentukan nilai terbesar</i>	
1	Mulai
2	Deklarasi variabel: a, b
3	Input/masukkan nilai: a, b
4	if a >= b
5	Cetak "Nilai a >= b"
6	ke langkah 9
7	else //lainnya
8	Cetak "Nilai a < b"
9	Selesai

Dalam dunia algoritma dan pemrograman akan sering diperhadapkan kondisi seleksi seperti contoh di atas, baik tipe kasusnya sama maupun berbeda.

c. Pseudocode dengan struktur perulangan

Contoh pseudocode untuk menuliskan angka 1 sampai dengan angka 5 yaitu sebagai berikut :

<i>Pseudocode cetak angka 1 - 100</i>	
1	Mulai
2	Deklarasi variabel dan inisiasi : a=0, b=100
3	a ← a + 1
4	if a > b
5	Cetak a
6	ke langkah 3
7	else //lainnya
8	Selesai

Maka dari pseudocode di atas maka tahap pengerjaannya yaitu pada Tahap awal variabel a diisi 1 dan variabel b diisi 100, (Perulangan Pertama) kemudian variabel a diisi dengan $a = a + 1$ sehingga $a = 1$, karena $a = a + 1$ sama dengan $a = 0 + 1$. Langkah berikutnya a diuji dengan b apakah $a > b$ atau $1 > 100$, tentu jawabannya (Yuhendra, 2013) salah atau "Tidak", karena hasil seleksi adalah "Tidak" maka variabel a dicetak. (Perulangan Kedua) Dan kemudian kembali ke $a = a + 1$ atau a

= 1 + 1 sehingga $a = 2$, proses ini akan berulang dan mencetak angka 100 hingga sampai memenuhi kondisi $a > b$ benar atau “Ya” dan algoritma perulangan selesai.

KESIMPULAN

Algoritma merupakan urutan langkah- langkah pada penyelesaian permasalahan yang disusun secara sistematis serta logis. Kata logis ialah kata kunci. Langkah langkah tersebut wajib logis, ini berarti nilai kebenarannya mesti bisa didetetapkan, benar ataupun salah. algoritma juga ialah dasar dalam pemrograman, Sebaliknya Algoritma Pemrograman merupakan urutan langkah- langkah dalam menuntaskan permasalahan memakai bahasa pemrograman serta disusun secara sistematis serta logis.

Struktur Dasar Algoritma Pemrograman ialah tentang yang menjadi dasar dari pemrograman itu sendiri. Struktur dasar Algoritma ini ialah runtunan, pemilihan, serta perulangan dari masing masing aksi ataupun dapat pula dengan metode campuran. Jadi struktur dasar pembentukan algoritma terdapat 3, ialah:

1. Struktur Runtunan: digunakan sebagai program yang pernyataannya sequential ataupun urutan.
2. Struktur Pemilihan: digunakan buat program yang menggunakan pemilihan ataupun penyeleksian keadaan.
3. Struktur Perulangan: digunakan untuk program yang pernyataannya bakal dieksekusi secara berulang- ulang.

Dalam membuat ataupun menuliskan Algoritma Pemrograman maka memerlukan notasi bahasa serta juga struktur dasar yang jadi ketetapanannya sehingga Algoritma Pemrograman bisa dinilai benar serta jauh dari kata kesalahan. Supaya program bisa dilaksanakan oleh komputer, program tersebut mesti ditulis dalam suatu bahasa yang dimengerti oleh komputer. Bahasa komputer yang digunakan dalam menulis program dinamakan bahasa pemrograman. Urutan langkah- langkah yang sistematis untuk menyelesaikan suatu permasalahan dinamakan algoritma.

DAFTAR PUSTAKA

- Allen Margareta, a. i. (n.d.). Pengaruh penerapan algoritma terhadap pembelajaran pemrograman komputer. *inovasi pendidikan matematika*, 2(2), 126-135.
<https://jurnal.univpgri-palembang.ac.id/index.php/indiktika/article/download/4125/3805>
- A. S. Putra, “2018 *Artikel Struktur Data, Audit Dan Jaringan Komputer*,” 2018.
- dkk, A. B. (2014). *Logika Dan Algoritma*. surabaya: Politeknik Elektronika negeri surabaya.
- Edy, B. (2015). *Belajar dasar algoritma dan pemrograman*. samarinda: PEMULA.
- Maulana, G. G. (2017, maret). Pembelajaran Algoritma Dan Pemrograman EL-Goritma Berbasis web. *jurnal teknik mesin (JTM)*, 6, 69-73. Retrieved from
<https://publikasi.mercubuana.ac.id/index.php/jtm/article/download/1183/1453>
- Nuraini, R. (n.d.). Analisis Algoritma Definite Integration Menggunakan metode Desk Check. *jurnal informatika dan bisnis*. Retrieved from
<https://jurnal.kwikkiangie.ac.id/index.php/JIB/issue/view/23/30>
- Pranata, A. 2005. *Algoritma dan Pemrograman*. Yogyakarta: Penerbit Graha Ilmu.
- udayana, K. B. (2018, agustus). Penerapan komponen dan struktur algoritma pada algoritma dan pemrograman dasar. *jurnal bisnis dan teknologi politeknik NSC Surabaya*, 5(2355-8865). Retrieved from
<https://nscpolteksby.ac.id/ejournal/index.php/jbt/article/view/73/67>
- Yuhendra. (2013). *Algoritma Dan Pemrograman* . padang: Jurusan teknik informatika Fakultas Teknologi Industri Institut Teknologi Padang.
- Yunianita, U. B. (2020). *Buku Ajar Algoritma Dan Pemrograman Dalam Bahasa C++*. Sidoarjo, Jawa Timur: UMSIDA Press.