**DAFTAR ISI**

**BAB II : DASAR-DASAR ALGORITMA**

**A. Dasar-Dasar Algoritma**

 1. Proses

2. Aksi

3. Instruksi

**B. Struktur Penulisan Algoritma**

1. Kepala Algoritma

2. Deklarasi

3. Deskripsi

**C. Komentar (Comment) Algoritma**

**D. Penamaan/Pengenal (Identifier)**

**E. Struktur dasar algoritma dan pemograman**

1. Runtunan
2. Pemilihan
3. Perulangan
4. flowchart

**F. Variabel**

**G. Konstanta**

**H. Ekspresi**

1. operator aritmatika

2. operator perbandingan

3. operator string

4. Logika

5. Tingkatan-tingkatan dalam operator

**I. Penugasan (Assigment)**

**J. Input dan Output**

1. Input

2. Output

**BAB II : DASAR-DASAR ALGORITMA**

**A. Dasar – Dasar Algoritma**

 **1. Proses**

 Dikerjakan oleh pemroses (CPU)

 **2. Aksi**

 Mengerjakan suatu tindakan (Aksi), maka sejumlah Operasi sesuai dengan Aksi yang dikerjakan oleh Pemroses

 **3. Instruksi**

 Algoritma disusun oleh sederatan Langkah Instruksi yang Logis



****

**B. Struktur Penulisan Algoritma**

Mulai dari bab ini dan seterusnya, penulisan algoritma hendaknya menggunakan format pseudocode agar nantinya mudah diterjemahkan ke dalam bahasa pemrograman Pascal dalam mata kuliah Pemrograman pada semester berikutnya. Penulisan algoritma dalam format pseudocode ini mengikuti aturan-aturan yang dipakai dalam buku tulisan Rinaldi Munir yang berjudul “Algoritma dan Pemrograman”. Dalam buku tersebut, struktur penulisan algoritma dibagi menjadi 3 bagian yaitu bagian kepala, bagian deklarasi, dan bagian deskripsi.

Struktur lengkapnya dapat dilihat di bawah ini.

**01 Algoritma NAMA\_ALGORITMA**

**02 {Penjelasan tentang algoritma, yang berisi uraian singkat mengenai apa yang dilakukan oleh algoritma}**

**03 DEKLARASI**

**04 {semua nama yang dipakai, meliputi nama tipe, nama tetapan, nama variabel, nama prosedur, dan nama fungsi didefinisikan di sini}**

**05**

**06 DESKRIPSI:**

**07 {semua langkah/aksi algoritma dituliskan di sini}**

**1. Kepala Algoritma**

Kepala algoritma adalah bagian yang terdiri dari nama algoritma dan penjelasan algoritma. Aturan pemberian nama algoritma mengacu pada aturan pemberian nama pengenal yang akan dibahas pada bagian berikutnya. Nama algoritma hendaknya singkat, namun mencerminkan isi algoritma secara keseluruhan. Bagian penjelasan algoritma berisi penjelasan mengenai hal-hal yang dilakukan oleh algoritma secara singkat. Contoh kepala algoritma dapat dilihat di bawah ini.

**01 Algoritma LUAS\_SEGITIGA**

**02 {Menghitung luas segitiga berdasarkan panjang alas dan tinggi segitiga yang diinput user. Luas segitiga dapat diperoleh dari rumus Luas = 0,5 x alas x tinggi}**

**01 Algoritma Menentukan\_Bilangan\_Prima**

**02 {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}**

**2. Deklarasi**

Bagian deklarasi berisikan semua nama pengenal yang dipakai di dalam algoritma. Nama tersebut dapat berupa nama tetapan (konstanta), nama peubah (variabel), nama tipe, nama prosedur, dan nama fungsi. Contohnya dapat dilihat di bawah ini.

**01 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}**

**3. Deskripsi**

Bagian deskripsi merupakan inti dari algoritma. Bagian ini berisi uraian langkah-langkah penyelesaian masalah. Langkah-langkah ini dituliskan dengan notasi-notasi yang akan dijelaskan pada bab-bab selanjutnya.

**C. Komentar (Comment) Algoritma**

Penulisan algoritma yang baik selalu disertai pemberian komentar. Komentar merupakan suatu penjelasan mengenai suatu hal yang tertulis dalam algoritma dan komentar bukanlah merupakan bagian dari langkah-langkah penyelesaian suatu algoritma.

Komentar ditulis di antara tanda baca **kurung kurawal buka** dan **kurung kurawal tutup**. “ { } “. Pada bagian kepala algoritma, penjelasan algoritma merupakan suatu komentar. Contoh lainnya dapat Anda lihat dalam contoh bagian deklarasi di atas. Pemberian komentar bukanlah suatu keharusan, namun algoritma yang disertai komentar-komentar yang tepat akan memudahkan pembaca lainnya untuk mengerti algoritma yang Anda buat.

**D. Penamaan/Pengenal (Identifier)**

Dalam algoritma, ada beberapa hal yang harus diberi nama atau pengenal. Hal-hal tersebut meliputi: nama algoritma, nama tetapan (konstanta), nama peubah (variabel), nama tipe, nama prosedur, dan nama fungsi. Pemberian nama harus mengikuti aturan-aturan sebagai berikut:

1. hanya boleh terdiri dari huruf, angka, atau garis bawah

2. tidak boleh dimulai dengan angka

3. tidak membedakan huruf kapital maupun huruf kecil (non case-sensitive)

4. panjang tidak dibatasi

5. harus unik, artinya tidak boleh ada nama pengenal yang sama untuk hal yang berbeda

6. hendaknya mencerminkan kegunaannya

Pemberian nama pengenal ini biasanya mengikuti salah satu dari 2 gaya berikut:

1. pemisah antar-kata menggunakan tanda garis bawah. Contoh: Luas\_Segitiga

2. pemisah antar-kata menggunakan huruf kapital Contoh: LuasSegitiga

Beberapa contoh penamaan yang valid maupun tidak valid:

**01 Nama\_4\_Pegawai {valid}**

**02 2\_luas {salah, karena diawali angka}**

**03 ^LuasSegitiga {salah, karena terdapat tanda ^}**

**04 Luas\_Segitiga\_&\_Lingk {salah, karena terdapat tanda &}**

**05 Hati2 {valid}**

**E. Struktur dasar algoritma dan pemograman**

1. Runtunan (**Sequence**)

Terdiri dari satu atau lebih instruksi. Tiap instruksi dikerjakan urut sesuai urutan penulisan

1. Pemilihan (**Selection**)

Instruksi dikerjakan bila telah memenuhi kondisi tertentu

**If** kondisi **Then** aksi1 (**kondisi=true**) **else** aksi2 (**kondisi=False**)

1. Perulangan (**Repetition**)

Mengerjakan instruksi yang berulang.

Repet aksi until kondisi (true=selesai; false=ulang)

**F. Variabel**

Variabel merupakan sesuatu yang digunakan sebagai tempat penyimpanan nilai dalam memori yang isinya dapat diubah. Contoh:

**01 k,l : integer**

**02 c : char**

**03 kalimat : string**

**04 nilai1, nilai2, rata2 : real**

**F. Konstanta**

Konstanta merupakan suatu nilai yang telah ditetapkan di awal pembuatan algoritma dan nilainya tidak dapat diubah oleh proses dalam algoritma. Cara mendefinisikan konstanta adalah dengan menambahkan kata kunci **const** diawal nama konstanta dan **diletakkan di bagian deklarasi**. Contoh:

**01 Deklarasi:**

**02 const phi = 3.14**

**03 const k = 0.5**

**04 const password = „SeCReT‟**

**G. Ekspresi**

Ekspresi adalah serangkaian perhitungan nilai yang menghasilkan suatu nilai yang diinginkan. Ekspresi terdiri dari **operand** dan **operator**. Operand adalah nilai-nilai yang akan dihitung. Operator adalah lambang operasi yang dipakai dalam perhitungan. Contoh: 6 + 3, angka 6 dan 3 adalah operand, sedangkan tanda + merupakan operator. Operand dapat berupa **tetapan** (konstanta), **peubah** (variabel), atau **hasil dari suatu fungsi**. Operator dapat berupa operator **unary**, yaitu operator yang hanya memerlukan 1 operand, dan operator **binary**, yaitu operator yang memerlukan 2 operand. Beberapa jenis operator yang dipakai dalam algoritma adalah:

**1. operator aritmatika**

Lambang Operator

Jenis

Tipe Operand1

Tipe Operand2

Tipe Hasil

+, –

unary

Integer Real

Integer Real

+, –, \*

Binary

Integer Integer Real Real

Integer Real Integer Real

Integer Real Real Real

Div

Binary

integer

integer

integer

/

Binary

Integer Integer Real Real

Integer Real Integer Real

Real Real Real Real

Mod

Binary

Integer

Integer

Integer

**2. operator perbandingan**

Lambang Operator

Jenis

Tipe Operand1

Tipe Operand2

Tipe Hasil

< <= > >= = <>

Binary Binary Binary Binary Binary Binary Binary Integer Integer Real Char Char String Boolean

Integer Real Real Char String String Boolean Boolean Boolean Boolean Boolean Boolean Boolean Boolean

Keterangan:

1. perbandingan char mengacu pada urutan karakter dalam tabel ASCII.

 Contoh: „a‟ > „A‟ akan menghasilkan true

2. perbandingan string, dibandingkan per karakter yang berkesesuaian mulai dari karakter pertama.

 Contoh: „abc‟ > „aBc‟ akan menghasilkan true.

3. perbandingan boolean, false diidentikkan dengan angka 0, sedangkan true diidentikkan dengan angka 1. Jadi true > false akan menghasilkan true.

**3. operator string**

Yaitu lambang + yang merupakan operasi penyambungan (concatenation) Contoh: „anak‟ + „ku‟ akan menghasilkan „anakku‟

**4. Logika**

Lambang Operator

Jenis

Tipe Operand1

Tipe Operand2

Tipe Hasil

not

Unary

Boolean

Boolean

and, or, xor

Binary

Boolean

Boolean

Boolean

a

not a

True False

False True

a

b

a and b

a or b

a xor b

true true false false

true false true false

true false false false

true true true false

false true true false

**5. Tingkatan-tingkatan dalam operator**

Precedence

operator

Keterangan

First (High)

NOT + –

Unary operator

Second

\* / div mod

Perkalian dan pembagian

AND

Boolean

Third

+ –

Penjumlahan

OR XOR

Boolean

Fourth (Low)

= <> < > >= <=

Relasional

**H. Penugasan (Assigment)**

Penugasan atau **Assignment** merupakan pemberian nilai ke variabel **secara langsung**. Notasi yang digunakan adalah . Nilai yang dapat diberikan adalah **tetapan**, **peubah**, **ekspresi**, maupun **nilai yang dihasilkan oleh fungsi**. Syarat penugasan adalah nilai yang diberikan **harus sesuai** dengan tipe variabel. Apabila tipenya tidak sama, maka berlaku tipe yang lebih luas dapat menampung tipe yang lebih sempit. Tipe **integer** dapat ditampung oleh tipe **real**, sebaliknya tipe **real tidak dapat ditampung** oleh tipe **integer**. Begitu pula dengan **string** dan **char**, char dapat diberikan ke string, namun tidak sebaliknya.

Contohnya:

**01 Deklarasi:**

**02 a,b : integer**

**03 c,d : real**

**04 nama1, nama2 : string**

**05 huruf : char**

**06 ketemu : boolean**

**07**

**08 Deskripsi:**

**09 a** 🡨**3 {boleh}**

**10 b** 🡨**a + 4 {boleh}**

**11 c** 🡨**3 {boleh}**

**12 a** 🡨**0.5 {tidak boleh karena a tidak dapat menampung real}**

**13 a** 🡨**b + c {tidak boleh karena b+c bertipe real}**

**14 huruf** 🡨**„?‟ {boleh}**

**15 nama1 „Ani‟ 🡨{boleh}**

**16 nama2** 🡨**nama1 + huruf {boleh}**

**17 nama1** 🡨**„a‟ {boleh}**

**18 huruf** 🡨**nama1 {tidak boleh karena nama1 bertipe string}**

**19 ketemu**🡨**true {boleh}**

**20 ketemu** 🡨**„true‟ {tidak boleh karena „true‟ bertipe string}**

**I. Input dan Output**

Menurut Donald E. Knuth (1973, p4-6), algoritma memiliki 5 ciri pokok, yaitu:

1. Finiteness Algoritma harus selalu berakhir setelah melakukan sejumlah langkah berhingga.

2. Definiteness Setiap langkah dari suatu algoritma harus terdefinisi secara tepat (logis).

3. Input Suatu algoritma dapat tidak memiliki input, ataupun memiliki satu atau lebih dari satu input.

4. Output Suatu algoritma harus memiliki paling sedikit satu output.

5. Effectiveness Algoritma yang dibuat diharapkan efektif, artinya setiap langkah yang hendak dilaksanakan dalam algoritma haruslah sederhana sehingga dapat secara prinsip dilakukan dengan tepat dan dalam waktu yang masuk akal apabila dilakukan secara manual oleh manusia dengan menggunakan pensil dan kertas.

**1. Input**

Input artinya meminta data yang diperlukan dari user. Sebagai contoh, dalam menghitung luas persegi panjang, tentu diperlukan data berupa besarnya panjang dan lebar bangun persegi panjang tersebut. Dengan kata lain, algoritma menentukan luas persegi panjang mempunyai 2 input berupa panjang dan lebar persegi panjang. Algoritma di buku ini menggunakan kata kunci **read** untuk menginput data. Bentuk penulisannya adalah Read(variabel1, variabel2, ..., variabeln) Data yang dapat diinputkan hanyalah data berupa integer, real, char, atau string. Sedangkan **data boolean tidak dapat** diinputkan menggunakan read. Dalam algoritma, kita tidak perlu memikirkan dari peralatan mana user menginput data, apakah dari mouse, keyboard, scanner, dan lain sebagainya. Hal itu merupakan masalah pemrograman. Pembuat algoritma juga tidak perlu memikirkan masalah tampilan saat penginputan berlangsung. Contohnya adalah

**01 Algoritma ContohPenginputan**

**02 {contoh penggunaan read untuk menginput sejumlah data dari user}**

**03 Deklarasi:**

**04 a,b,c : integer**

**05 d : real**

**06 e : char**

**07 f : string**

**08 g : boolean**

**09 Deskripsi**

**10 {menginput data bil bulat, kemudian dimasukkan ke variabel a}**

**11 Read(a)**

**12 {menginput 2 data integer, dimasukkan ke variabel b dan c}**

**13 read(b,c)**

**14 {menginput data pecahan, kemudian dimasukkan ke variabel d}**

**15 read(d)**

**16 {menginput data bulat dan pecahan}**

**17 read(a,d)**

**18 {menginput data bulat, pecahan, dan 1 karakter}**

**19 read(a,d,e)**

**20 {menginput data string}**

**21 read(f)**

**22 {berikut ini adalah** suatu kesalahan**}**

**23 read(g) {data boolean tidak dapat diinputkan}**

**2. Output**

Output artinya mencetak informasi yang dihasilkan oleh algoritma. Sebagai contoh dalam algoritma menghitung luas persegi panjang, hasil akhir yang diinginkan adalah luas persegi panjang. Dengan kata lain, algoritma tersebut memiliki satu output yaitu luas persegi panjang. Algoritma dalam buku ini menggunakan kata kunci **write** untuk mencetak suatu data. Bentuk penulisannya adalah write(data1, data2, ..., datan) dimana data dapat berupa suatu data konstan, nama konstanta, ekspresi, maupun suatu variabel. Dalam algoritma, kita tidak mempermasalahkan ke peralatan mana data tersebut akan dicetak, karena merupakan masalah pemrograman. Kita juga tidak mempermasalahkan dalam format bagaimana data tersebut dicetak. Contoh:

**01 Algoritma ContohPengoutputan**

**02 {berisi contoh-contoh penggunaan write untuk mencetak data}**

**03 deklarasi:**

**04 const k = 0.5**

**05 a : integer**

**06 b : real**

**07 c : char**

**08 d : string**

**09 e : boolean**

**10 deskripsi**

**11 read(a) {menginput sebuah data integer}**

**12 write(a) {mencetak data hasil input tadi}**

**13 write(b) {**tidak boleh**, karena variabel b belum ada isinya}**

**14 c** **„y‟**

**15 d** **„stmik‟**

**16 write(c,d) {mencetak isi variabel c dan d}**

**17 write(„Susi kuliah di „,d) {mencetak Susi kuliah di stmik}**

**18 write(5 + 6) {mencetak bilangan 11, yaitu hasil dari 5+6}**

**19 e** **2 = 1 + 3**

**20 write(e) {mencetak FALSE karena e berisikan false}**

**21 write(k) {mencetak 0.5 karena k berisi 0.5}**