

PENGANTAR ALGORITMA DAN PEMROGRAMAN

Saniman dan Muhammad Fathoni

ABSTRAK

Algoritma adalah susunan langkah-langkah sistematis dan logis dalam pemecahan suatu masalah. Ada 3 cara dalam menyusun algoritma yaitu: (1) *Dengan merumuskan langkah-langkah pemecahan masalah melalui kalimat yang terseruktur (tersusun secara logis)*; (2) *Menggabungkan kalimat dengan penggalan statements yang ada di suatu bahasa pemrograman (mis : Pascal)*. Biasa disebut Pseudo code (mirip kode/perintah pemrograman); dan (3) *Menggunakan diagram alir (flowchart)*. Algoritma adalah merupakan jantung ilmu komputer atau informatika. Program adalah merupakan perwujudan atau implementasi dari algoritma. Program ditulis dalam salah satu bahasa pemrograman. Kegiatan menulis program disebut pemrograman (*programming*).

Kata Kunci: algoritma, flowchart, pseudocode, program

A. PENDAHULUAN

Sejarah kata Algoritma berasal dari nama seorang ahli matematika bangsa Arab yaitu **Abu Ja'far Muhammad ibnu Musa al-Khuwarizmi**. Al-Khuwarizmi dibaca oleh orang Barat menjadi *Algorism*. Perubahan kata *algorism* menjadi *algorithm* karena kata *algorism* sering dikelirukan dengan *arithmetic*, sehingga akhiran *-sm* berubah menjadi *-thm*. Lambat laun kata *algorithm* dipakai sebagai metode perhitungan (komputasi) secara umum, sehingga kehilangan makna aslinya. Dalam bahasa Indonesia kata *algorithm* diserap menjadi *algoritma*.

Definisi Algoritma adalah : Susunan langkah-langkah sistematis dan logis dalam pemecahan suatu masalah.

Ada 3 cara dalam menyusun algoritma yaitu :

Algoritma cara ke-1 :

Dengan merumuskan langkah-langkah pemecahan masalah melalui kalimat yang terseruktur (tersusun secara logis).

Contoh:

Ada persoalan sbb : Dua buah gelas misalkan gelas A berisi air warna merah dan gelas B berisi air warna biru. Permasalahannya adalah bagaimana mempertukarkan isi kedua gelas A dan B sehingga gelas A berisi air warna biru dan gelas B berisi air warna merah. Buatlah algoritmanya.

Jawab:

Langkah-langkahnya sbb:

1. Sediakan satu gelas kosong misalkan C
2. Tuangkan isi gelas A ke gelas C
3. Tuangkan isi gelas B ke gelas A
4. Tuangkan isi gelas C ke gelas B
5. Selesai

```
{Program : URUT.PAS}
PROGRAM URUT_DATA;
USES CRT;
VAR D,X,Z : BYTE;
```

```

T      : LONGINT;
R      : REAL;
DATA   : ARRAY[1..5] OF BYTE;

BEGIN
  CLRSCR;
  T:=0;
  FOR D:=1 TO 5 DO
    BEGIN
      Write('DATA KE ',D,' : '); ReadLn(DATA[d]);
      T:=T+Data[d];
    END;
  WriteLn;
  Write('Data sebelum diurut : ');
  FOR D:= 1 TO 5 DO
    Write(Data[d]:3);

  { -- URUT DATA --- }
  FOR D:= 1 TO 4 DO
    BEGIN
      FOR X:= D+1 TO 5 DO
        BEGIN
          IF DATA[d] > DATA[x] THEN
            BEGIN
              Z:=DATA[d];
              DATA[d] := DATA[x];
              DATA[x] := Z;
            END;
        END;
    END;
  WriteLn;
  WriteLn;
  Write('DATA SETELAH DIURUT : ');
  FOR D:=1 TO 5 DO
    WRITE(DATA[d]:3);

  WRITELN;
  WRITELN('TOTAL : ',T);
  R:=T/D;
  WRITELN('RATA : ',R:10:2);
  READLN;
END.

```

Algoritma cara ke-2 :

Menggabungkan kalimat dengan penggalan statements yang ada di suatu bahasa pemrograman (mis : Pascal). Biasa disebut Pseudo code (mirip kode/perintah pemrograman)

Contoh : Algoritma untuk konversi bilangan berbasis-10 menjadi bilangan berbasis-2.

1. **INPUT** a { a adalah bilangan berbasis 10}
 - Hit = 1 { hit adalah indeks untuk menyimpan sisa hasil bagi }
2. **DO WHILE** a>0
 - sb = sisa bagi a dengan 2
 - bil(hit) = sb

hit = hit + 1

a = hasil pembagian a dengan 2

ENDDO

3. **DO WHILE** hit > 0

cetak bil(hit);










hit = hit + 1

ENDDO

Algoritma cara ke-3 :

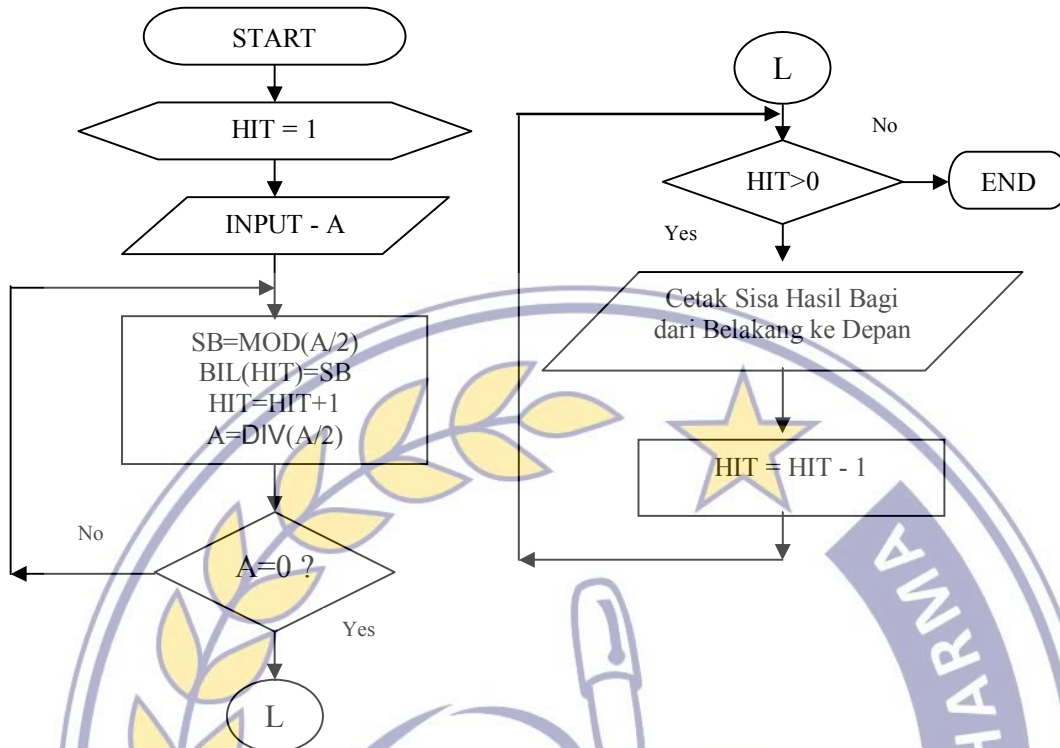
Menggunakan diagram alir (flowchart).

Lambang lambang flowchart secara umum

No	Lambang	Keterangan
1		Terminal (Start atau Finish) : Sebagai lambang untuk mengawali flowchart dan mengakhiri flowchart.
2		Input atau Output : Sebagai lambang untuk memasukkan data atau mengeluarkan data atau hasil proses data
3		Proses : Sebagai lambang proses terhadap data maupun formulasi matematis lainnya.
4		Predefined proses : Merupakan suatu Prosedur atau Rutin atau Sub Program baik yang terpisah dari program utama maupaun yang terpadu dalam satu program.
5		Preparasi : Pemberian harga awal terhadap suatu pengenalan (indentifier) yang akan digunakan
6		Decision (Pengambilan keputusan berdasarkan hasil pengujian ekspresi logis apakah hasilnya True atau False)
7		Connector dalam satu halaman
8		Connector ke halaman lain
9		Line : Sebagai lambang arah / arus dari aliran program

Gambar 1.1 : Lambang-lambang flowchart.

Contoh: flowchart konversi bilangan basis 10 menjadi bilangan basis-2 :



Gambar 1.2 : Flowchart konversi bil desimal ke biner.

Buatlah program menggunakan Turbo Pascal untuk konversi bil basis 10 ke bil basis 2 sesuai flowchart diatas. Simpan program dengan nama Biner.Pas

```

{ Nama Program : Biner.Pas }
Program Konversi_Biner;
Uses Crt;
Var Hit,A,SB : Integer;
    Bil:Array[1..100] of Integer;

Begin
  ClrScr;
  Write('Masukan bilangan basis 10 : '); ReadLn(A);
  Hit:=1;
  Repeat
    SB := A Mod 2;
    Bil[Hit]:=SB;
    Hit:=Hit+1;
    A := A Div 2;
  Until A=0;
  Writeln;
  Repeat
    Write(Bil[Hit]:2);
    Hit:=Hit-1;
  
```



```
Until Hit=0;  
ReadLn;  
End.
```

a. Algoritma merupakan jantung Ilmu Informatika

Algoritma adalah merupakan jantung ilmu komputer atau informatika. Banyak cabang ilmu komputer yang diacu dalam terminologi algoritma. Namun jangan beranggapan algoritma selalu identik dengan ilmu komputer saja. Cara membuat kue atau masakan yang dinyatakan dalam resep masakan, itu juga merupakan algoritma. Ibu-ibu yang mencoba resep masakan tersebut akan membaca satu persatu langkah pembuatannya, lalu mengerjakan proses (melakukan aksi) sesuai yang ia baca. Secara umum, pihak yang mengerjakan proses disebut pemroses (processor). Pemroses dapat berupa manusia, komputer, robot, alat mekanik, alat elektronik dll. Melaksanakan algoritma berarti mengerjakan langkah-langkah yang tertulis dalam algoritma tersebut.

b. Mekanisme Pelaksanaan Algoritma

Agar algoritma dapat dilaksanakan dalam komputer maka algoritma harus di ubah ke notasi bahasa pemrograman sehingga disebut program. Jadi program adalah merupakan perwujudan atau implementasi dari algoritma. Program ditulis dalam salah satu bahasa pemrograman. Kegiatan menulis program disebut pemrograman(programming). Orang yang menulis program disebut pemrogram (programmer). Tiap langkah di dalam program disebut pernyataan atau instruksi. Jadi program adalah : "Sederetan instruksi yang sistematis dan logis yang menggunakan sintaks tertentu untuk menyelesaikan permasalahan".

Secara garis besar komputer tersusun atas 4 komponen utama yaitu :

1. Input device (piranti masukan)
2. Output device (piranti keluaran)
3. Unit pemroses utama (Central Processing Unit)
4. Memory (piranti penyimpanan sementara)

Mekanisme kerja ke-empat komponen tersebut dapat dijelaskan sbb :

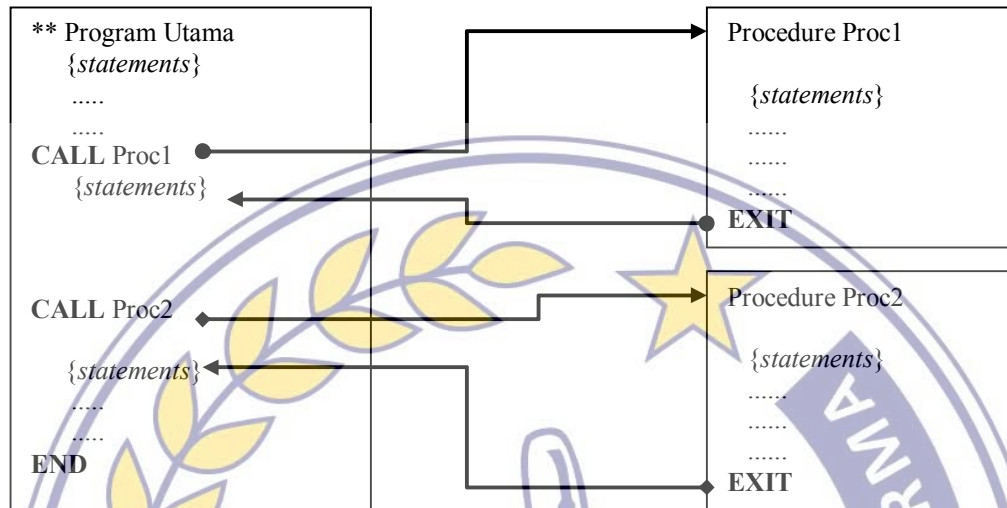
- Program dimasukan kedalam memori komputer.
- Setiap instruksi yang ada di memori di dikirim ke CPU untuk dieksekusi.
- CPU mengerjakan operasi-operasi yang bersesuaian dengan instruksi tsb.
 - Bila operasi memerlukan data maka data dibaca dari piranti masukan. Data yang dimasukan disimpan dimemori lalu dikirim ke CPU untuk operasi yang memerlukan operasi tadi.
 - Bila proses menghasilkan keluaran atau informasi, maka keluaran disimpan ke memori, lalu memori menuliskan keluaran ke piranti keluaran (mis : screen atau printer)

Perbedaan belajar Memprogram dengan Belajar Bahasa Pemrograman

- Belajar memprogram adalah belajar tentang metodologi pemecahan masalah, kemudian menuangkannya kedalam suatu notasi yang mudah dipahami. Belajar algoritma sama dengan belajar memprogram.
- Belajar bahasa pemrograman adalah belajar memakai suatu bahasa, aturan dan tata bahasanya, instruksi-instruksinya, cara pengoreasian *compiler*-nya. Belajar bahasa program contohnya adalah belajar Turbo- Pascal.

c. Pemrograman Prosedural.

Bahasa pemrograman yang menerapkan konsep prosedural didalam penulisan programnya misalnya adalah T.Pascal. Teknik pemrograman prosedural adalah : *program di bagi menjadi beberapa bagian program yang lebih kecil yang disebut modul atau sub program atau rutin atau procedure atau function*. Bagian-bagian program kecil ini nantinya di kontrol atau dikendalikan dari program utama (main program). Setiap sub program dapat di panggil berkali-kali dari program utama. Perhatikan skema konsep pemrograman prosedural berikut ini.



Gambar 1.3 : Diagram konsep prosedural.

Program Utama memanggil sub program **Procedure Proc1** kemudian melaksanakan semua statements yang ada di sub program **Procedure Proc1** hingga selesai (disini digambarkan dengan **Exit**). Kemudian program keluar dari sub program dan kembali ke perintah (statements) dibawah pemanggil. Demikian juga dengan prosedur-prosedur yang lain.

Contoh :

```

{ Program: Modular1.Pas}
Program Prosedure1;
Uses Crt;
Label Ulang;
Var Pilihan : integer;
{ ----- }
Procedure Pros_Persegi;
Var P,L,Luas : Integer;
Begin
  ClrScr;
  WriteLn('Menghitung Luas Persegi Panjang');
  Write('Masukkan Panjang : '); ReadLn(P);
  Write('Masukkan Lebar : '); ReadLn(L);
  Luas := P * L;
  Write('Luas adalah : ',Luas:10);
  ReadLn;
End;
{ ----- }
Procedure Pros_Lingkaran;

```

```

Var R, Luas : Real;
Begin
  ClrScr;
  WriteLn('Menghitung Luas Lingkaran');
  Write('Masukkan Panjang Jari-jari: '); ReadLn(R);
  Luas := Pi * R * R;
  Write('Luas adalah      : ',Luas:10:5);
  ReadLn;
End;

{ ----- }
Procedure Pros_Kubus;
Var S, Volume : Integer;
Begin
  ClrScr;
  WriteLn('Menghitung Volume Kubus');
  Write('Masukkan Panjang Sisi kubus: '); ReadLn(S);
  Volume := S * Sqr(S);
  Write('Luas adalah      : ',Volume:10);
  ReadLn;
End;
{ ----- Bag Utama Program ----- }
Begin
  ulang:
  ClrScr;
  WriteLn('Ini modul utama ');
  WriteLn('1. Luas Persegi panjang ');
  WriteLn('2. Luas Lingkaran ');
  WriteLn('3. Volume Kubus ');
  WriteLn('4. Selesai ');
  WriteLn;
  Write('Pilihan anda : '); ReadLn(Pilihan);
  Case Pilihan of
    1 : Pros_Persegi;
    2 : Pros_Lingkaran;
    3 : Pros_Kubus;
  Else
    Exit;
  End;
  GoTo ulang;
  ReadLn;
End.

```

B. ATURAN PENULISAN TEKS ALGORITMA

Agar algoritma mudah ditranslasikan ke dalam notasi bahasa pemrograman, maka sebaiknya notasi algoritmik tersebut berkoresponden dengan notasi bahasa pemrograman secara umum. Misalkan kita menulis perintah :

tulis nilai x dan y

dalam notasi algoritmik ditulis menjadi: write(x, y)

Tabel notasi algoritmik

No.	Notasi biasa	Notasi	Notasi
-----	--------------	--------	--------

		algoritmik	T.Pascal
1	masukan nilai x	readln(x)	ReadLn(x);
2	isikan nilai 5 kedalam x	$x \leftarrow 5$	X:=5;
3	isikan nilai x kedalam min	$\text{min} \leftarrow x$	Min:=x;
4	tambahkan nilai 1 ke X itulah X	$x \leftarrow x + 1$	X:=x+1;
5	tulis nilai x dan y	write(x,y)	Write(x,y);
6	jika a lebih besar dari b maka	if a>b then	If a>b Then

Aturan diatas tidak baku, hanya penyesuaian dengan bahasa pemrograman Turbo Pascal.

Pada dasarnya teks algoritma tersusun dari 3 bagian (blok) yaitu :

1. Kepala Algoritma atau judul (header)
 2. Bagian deklarasi.
 3. Bagian deskripsi (uraian) algoritma.
- **Header** atau kepala algoritma bagian yang terdiri dari nama algoritma dan penjelasan singkat (spesifikasi) tentang algoritma tersebut. Penjelasan di apit tanda kurung kurawal { - }.
 - **Deklarasi** merupakan bagian untuk mendefinisikan semua nama yang dipakai dalam algoritma. Nama dapat berupa nama peubah, tetapan, tipe, prosedur, fungsi, dll.
 - **Deskripsi** merupakan bagian yang menjelaskan atau menguraikan langkah-langkah penyelesaian masalah. Uraian ditulis baris perbaris sesuai urutan yang harus dikerjakan secara sistematis.

Gambaran sebuah algoritma seperti berikut :

```

Algoritma NAMA ALGORITMA
{ penjelasan tentang algoritma yang berisi uraian singkat
mengenai apa yang dilakukan oleh algoritma }
DEKLARASI :
{ semua nama yang dipakai meliputi nama peubah, tetapan,
tipe, prosedur, fungsi, label didefinisikan di bagian ini }
DESKRIPSI :
{ semua langkah/aksi algoritma ditulis dibagian ini }

```

C. TIPE, NAMA, DAN NILAI PENGENAL

Pada umumnya, program komputer bekerja dengan memanipulasi objek(data) di dalam memori. Objek yang akan diprogram bermacam-macam jenis atau tipenya, misalnya nilai numerik, karakter, string dan rekaman (record). Suatu tipe menyatakan pola penyajian data dalam komputer. Tipe data dapat dikelompokkan menjadi atas dua macam: tipe data dasar dan tipe tipe bentukan. Tipe dasar adalah tipe yang dapat langsung dipakai, sedangkan tipe bentukan dibentuk dari tipe dasar. Suatu tipe diacu dari namanya. Nilai-nilai yang dicakup oleh tipe tersebut dinyatakan di dalam ranah (domain) nilai.

Tipe Dasar

- Bilangan Logika
- Bil Bulat
- Bil Riil
- Karakter

Tipe	Nama Tipe	Ranah Nilai		Tetapan	Operasi
Bilangan Logika	boolean	True atau False		True dan False	not, and, or, xor
Bilangan Bulat	integer	1.byte	0..255	tidak mengandung titik desimal	1. aritmatika (+ - * div mod) 2. perbandingan (<, <=, >, >=, =, <>)
		2.shortInt	-128..127		
		3.word	0..65535		
		4.integer	-32768..32767		
		5.LongInt	-2147483648 ..2147483647		
Bilangan Riil	real	1. real	$2.9 \times 10^{-39} .. 1.7 \times 10^{38}$	harus ditulis dengan titik desimal	1. aritmatika (+ - * /) 2. perbandingan (<, <=, >, >=, <>)
		2. single	$1.5 \times 10^{-45} .. 3.4 \times 10^{38}$		
		3. double	$5.0 \times 10^{-324} .. 1.7 \times 10^{308}$		
		4.extended	$3.4 \times 10^{-4932} .. 1.1 \times 10^{4932}$		
Karakter	char	- semua alfabet - semua angka - tanda baca - operator aritmatik - karakter khusus		harus diapit tanda petik tunggal	perbandingan (=, <>, <, >, >=)

Tipe Bentukan

- String.
- Tipe dasar yang diberi nama tipe baru.
- Rekaman (*record*).

Tipe	Nama Tipe	Ranah Nilai	Tetapan	Operasi
String	string	deretan karakter yang didefinisikan pada ranah karakter	harus diapit petik tunggal	1. Penyambungan (+) 2. Perbandingan (=, <>, <, >, >=, <=) hasil operasi berupa bilangan logic (true/false)

Tipe dasar yang diberi nama tipe baru :

Kita dapat memberi nama baru untuk tipe dasar dengan kata kunci type ranah nilai, tetapan dan operasi sama seperti tipe dasar aslinya.

Contoh :

type BilBulat : integer;

BilBulat adalah tipe bilangan bulat yang sama dengan tipe integer.

Rekaman

Rekaman disusun oleh satu atau lebih *field*. Tiap *field* menyimpan data dari tipe dasar tertentu atau tipe bentukan yang sudah didefinisikan. Nama rekaman ditentukan oleh pemrogram sendiri.

Contoh :

NilMhs adalah nama tipe terstruktur yang menyatakan nilai ujian seorang mahasiswa untuk suatu mata kuliah (MK) yang diambil. Data setiap mahasiswa adalah NIM (Nomor Induk Mahasiswa), Nama Mahasiswa, Kode Mata Kuliah, Nilai mata kuliah.

NIM	NamaMhs	KodeMK	Nilai
-----	---------	--------	-------

Cara menuliskan tipe NilMhs :

```

type NilMhs : record
  < NIM      : string { Nomor Induk Mahasiswa}
     NamaMhs : string
     KodeMK  : string {kode mata kuliah}
     Nilai   : integer {nilai M.Kuliah }
  >
    
```

Jika dideklarasikan M adalah peubah bertipe NilMhs, maka cara mengacu tiap *field* pada rekaman M adalah :

M.NIM
 M>NamaMhs
 M.KodeMK
 M.Nilai

Tipe NilMhs didefinisikan sebagai berikut :

Nama Tipe : NilMhs
 Ranah Nilai : sesuai ranah masing-masing *field*
 Contoh tetapan: <'2007001', 'Shulthan Sany', 'FI123',85>
 <'2007002', 'Sundari Zahra', 'MA222',78>
 Operasi : - operasi aritmatik bilangan bulat terhadap Nilai
 - operasi *string* terhadap NIM, NamaMhs, KodeMK

Nama

Setiap objek diberi nama agar objek tersebut mudah diidentifikasi dari objek lainnya. Dalam algoritma **nama** dipakai sebagai pengidentifikasi "sesuatu" dan pemrogram mengacu "sesuatu" itu melalui namanya. Karena itu tiap nama haruslah unik (tidak sama). Didalam algoritma "sesuatu" yang diberi nama itu dapat berupa :

1. Label
 Label merupakan pengenalan baris. Suatu baris dalam program diberi nama/pengenal. Label ini diberikan pada baris tertentu biasanya untuk instruksi percabangan.
2. Peubah (*variabel*)
 Variabel adalah alamat memory yang diberi nama sebagai tempat penyimpanan data/informasi dan isi variabel dapat diubah.
3. Tetapan (*constant*)
 Sama dengan variabel tetapi isinya tetap (tidak bisa diubah) selama pelaksanaan program.
4. Tipe bentukan. Nama tipe bentukan diberikan oleh pemrogram.
5. Fungsi (*function*).

6. Prosedur (*procedure*).

Nilai

- Pengisian nilai ke dalam nama peubah

Nilai adalah besaran dari tipe data yang sudah dikenal. Nilai dapat berupa isi yang disimpan oleh nama peubah (variabel) atau nama tetapan(constanata), atau hasil dari perhitungan atau nilai yang dikirim oleh fungsi lain.

Cara mengisi nilai ke dalam nama peubah dapat dilakukan sbb :

- Pengisian nilai secara langsung.

```
Nama := 'Budi Santoso';
```

```
Gaji := 2500000;
```

```
Tunj := Gaji * 0.1;
```

- Pembacaan nilai dari piranti masukan (*input device*)

```
ReadLn (Nama);
```

```
ReadLn (Gaji);
```

- Ekspresi

Transformasi nilai(data) menjadi keluaran(informasi) melalui suatu proses perhitungan dinyatakan dalam suatu *ekspresi*. Ekspresi terdiri dari operand dan operator. Operand adalah nilai yang dioperasikan, operator adalah lambang untuk operasi. Ada tiga macam ekspresi yaitu :

- Ekspresi Aritmatik menggunakan operator aritmatika.

```
c:=a*b;
```

```
a:=a+1;
```

```
d:=(a+b)/c*0.1;
```

Operator yang digunakan (+, -, *, /, div, mod, sqr, sqrt)

- Ekspresi Relational (ekspresi perbandingan yang menghasilkan nilai boolean)

```
a>b;
```

```
a>=d;
```

```
c=d;
```

```
e<f;
```

Operator yang digunakan (<, <=, >, >=, =, <>)

- Ekspresi Relational menggunakan operator (not, and, or, xor). Ekspresi ini menghasilkan nilai boolean (True atau False).

Kondisi 1	Kondisi 2	AND	OR	XOR
T	T	T	T	F
T	F	F	T	T
F	T	F	T	T
T	T	F	F	F

Not -T	F
Not - F	T

- Ekspresi String

Ekspresi string adalah ekspresi dengan operator "+" (operator penyambungan/*concatenation*).

```
a:='Budi';
```

```
b:='Joko';
```



```
write(a+b);           { BudiJoko}
```

- Menuliskan Nilai kepiranti keluaran (output device)

Nilai yang disimpan dimemori dapat ditampilkan ke piranti keluaran (misalnya layar peraga). Instruksi penulisan nilai dilakukan dengan notasi Write

```
Nama := 'Budi Santoso';  
Gaji := 2500000;  
Tunj := Gaji * 0.1;  
WriteLn>Nama);  
WriteLn>Gaji);  
WriteLn>Tunj);
```

D. STRUKTUR DASAR ALGORITMA

1. Proses, Instruksi dan Aksi

Tiap langkah instruksi mengerjakan suatu tindakan (aksi). Bila suatu aksi dilaksanakan, maka sejumlah operasi yang bersesuaian dengan aksi itu dikerjakan oleh pemroses. Efek dari pengerjaan suatu aksi dapat diamati dengan membandingkan keadaan pada saat aksi belum dimulai t_0 dan keadaan setelah aksi selesai dikerjakan t_1 .

t_0 : keadaan sebelum aksi dikerjakan
aksi
 t_1 : keaaan setelah aksi dikerjakan

2. Struktur Dasar Algoritma

Konstruksi algoritma dibangun atas 3 konstruksi dasar yaitu :

- Runtunan aksi
- Pemilihan aksi
- Pengulangan aksi

3. Runtunan Aksi (Squence)

Proses pelaksanaan instruksi dikerjakan beruntun dari urutan instruksi pertama, instruksi kedua, instruksi ketiga dan seterusnya hingga berakhir atau berhenti karna ada kesalahan instruksi. Urutan instruksi menentukan keadaan akhir dari algoritma. Bila urutannya diubah maka kemungkinan hasil akhir juga berubah.

Perhatikan ilustrasi berikut. Misalkan runtunan instruksi yang dilambangkan dengan A1, A2, A3, A4 dan A5 yang disusun sbb :

A1
A2
A3
A4
A5

Mula-mula pemroses melaksanakan instruksi A1. Selesai instruksi A1 dilanjutkan pelaksanaan instruksi A2. Intruksi A3 akan dilaksanakan jika instruksi A2 selesai dikerjakan dan seterusnya hingga A5. Pelaksanaan instruksi akan berurut kecuali jika ada perintah percabangan. Pengaruh urutan instruksi terhadap hasil adalah apabila urutan diubah maka hasil akhir kemungkinan juga berubah.

Contoh : Kita ambil persoalan seperti pada bab-I

Ada persoalan sbb : Dua buah gelas misalkan gelas A berisi air warna merah dan gelas B berisi air warna biru. Permasalahannya adalah bagaimana mempertukarkan isi kedua gelas A dan B sehingga gelas A berisi air warna biru dan gelas B berisi air warna merah. Buatlah algoritmanya.

Jawab :

<p>Langkah-langkahnya sbb:</p> <ol style="list-style-type: none"> 1. Sediakan satu gelas kosong misalkan C 2. Tuangkan isi gelas A ke gelas C 3. Tuangkan isi gelas B ke gelas A 4. Tuangkan isi gelas C ke gelas B 5. Selesai <p>Hasil akhir adalah gelas A berisi larutan biru dan gelas B berisi larutan merah. Keadaan hasil akhir ini adalah jawaban yang sesuai dengan keinginan algoritma.</p>	<p>Jika urutan diubah</p>	<p>Langkah-langkahnya sbb:</p> <ol style="list-style-type: none"> 1. Sediakan satu gelas kosong misalkan C 2. Tuangkan isi gelas B ke gelas A 3. Tuangkan isi gelas A ke gelas C 4. Tuangkan isi gelas C ke gelas B 5. Selesai <p>Hasil akhir adalah gelas A kosong dan gelas B berisi larutan campuran merah dan biru. Keadaan hasil akhir ini adalah jawaban yang tidak sesuai dengan keinginan algoritma.</p>
--	---------------------------	---

Contoh A

```
{Program : Lagu.Pas}
Program LaguIndonesiaRaya;
Uses Crt;

Begin
  ClrScr;
  WriteLn('Indonesia tanah
airku');
  WriteLn('Tanah tumpah
darahku');
  WriteLn('Disanalah aku
berdiri');
  WriteLn('Jadi pandu
ibuku');
  ReadLn;
End.
```

----- hasil-----

```
Indonesia tanah airku
Tanah tumpah darahku
Disanalah aku berdiri
Jadi pandu ibuku
```

Contoh B

```
{Program : Lagu.Pas}
Program LaguIndonesiaRaya;
Uses Crt;

Begin
  ClrScr;
  WriteLn('Disanalah aku
berdiri');
  WriteLn('Tanah tumpah
darahku');
  WriteLn('Indonesia tanah
airku');
  WriteLn('Jadi pandu
ibuku');
  ReadLn;
End.
```

----- hasil-----

```
Disanalah aku berdiri
Tanah tumpah darahku
Indonesia tanah airku
Jadi pandu ibuku
```

E. DAFTAR PUSTAKA

- Budi Sutedjo, S.Kom., MM dan Michael AN, S.Kom., 2004, **Algoritma dan Teknik Pemrograman Konsep, Implementasi dan Aplikasi**, Yogyakarta: ANDI.
- Fathul Wahid, 2004, **Dasar-Dasar Algoritma dan Pemrograman**, Yogyakarta: ANDI.
- Heri Sismoro, 2005, **Pengantar Logika Informatika, Algoritma dan Pemrograman Komputer**, Yogyakarta: ANDI.
- Jogiyanto H.M., 1997, **Teori dan Aplikasi Program Komputer Bahasa Pascal Jilid 1**, Yogyakarta: ANDI Offset.
- Yulikuspartono, S.Kom., 2004, **Pengantar Logika dan Algoritma**, Yogyakarta: ANDI.

