

PROCEDURE DAN FUNCTION PADA ALGORITMA

Saniman dan Muhammad Fathoni

ABSTRAK

Sebuah program yang besar dapat dipecah-pecah menjadi bagian-bagian program yang lebih kecil. Penggalan program ini disebut *modul* atau *subprogram* atau *rutin* atau *prosedur* atau *fungsi*. Teknik pemrograman ini dinamakan teknik pemrograman modular atau teknik pemrograman prosedural. Fungsi merupakan potongan program yang tidak dapat berdiri sendiri, karena fungsi memberikan hasil berupa suatu nilai dan memerlukan suatu tempat untuk menampung nilai tersebut. Ada fungsi standart (fungsi terpasang) dan ada fungsi buatan (defined function). Struktur fungsi sama dengan struktur algoritma yaitu ada *header* yang berisi nama fungsi dan spesifikasi fungsi, bagian *deklarasi*, dan *badan fungsi*. Setiap fungsi mempunyai nama yang unik serta daftar parameter formal jika diperlukan.

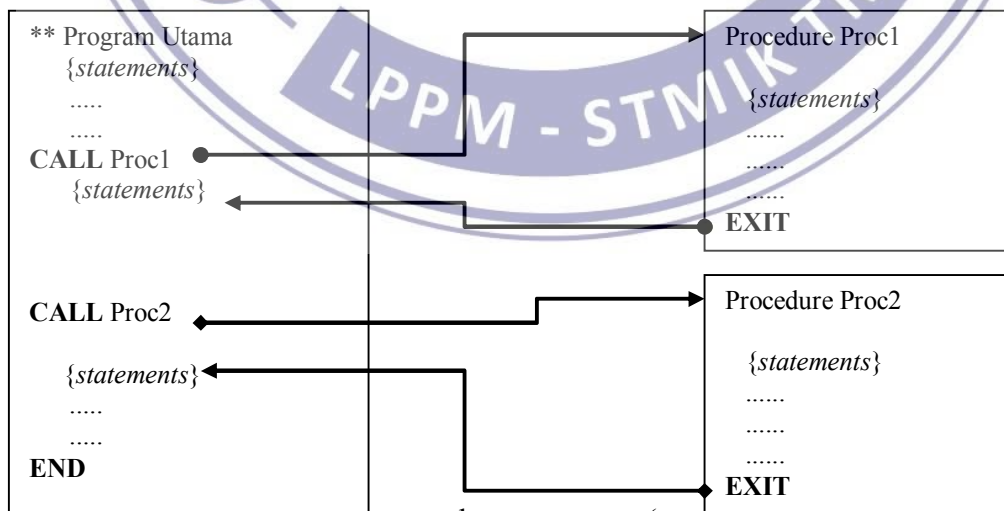
Kata Kunci: procedure, function, algoritma

A. PROCEDURE (PROSEDUR)

Sebuah program yang besar dapat dipecah-pecah menjadi bagian-bagian program yang lebih kecil. Penggalan program ini disebut *modul* atau *subprogram* atau *rutin* atau *prosedur* atau *fungsi*. Subprogram *kadangkala* cukup independen dari program utama sehingga programnya dapat dirancang tanpa mempertimbangkan konteks tempat ia digunakan. Subprogram ini dapat dieksekusi berulang-ulang sesuai keperluan. Teknik pemrograman ini dinamakan teknik pemrograman modular atau teknik pemrograman prosedural.

Modularisasi program memberikan dua keuntungan yaitu :

- Untuk menghindari penulisan teks program yang sama secara berulang. Dapat mengurangi panjangnya program.
- Kemudahan menulis dan menemukan kesalahan (*debug*) program.
- Menjadikan program lebih flexible.



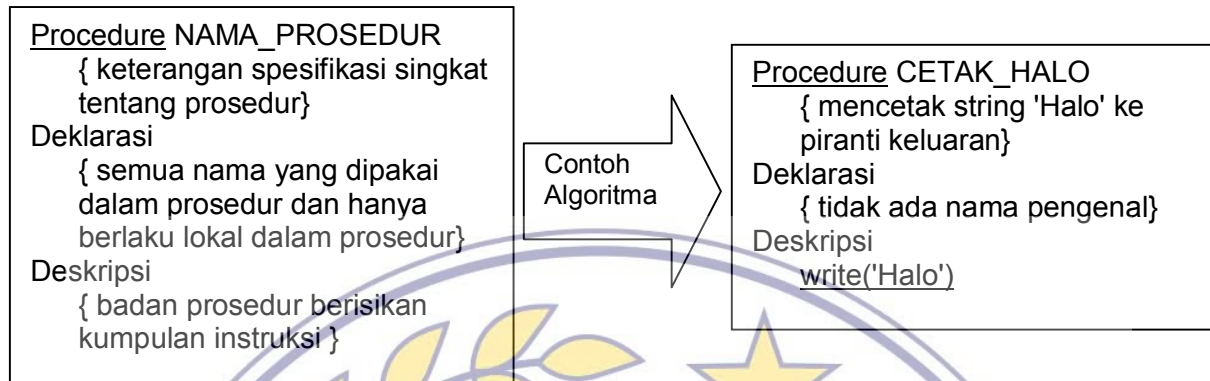
Ada dua
Contoh p

Gambar 1. Mekanisme penggunaan modul dari program utama

uan pemakaian.

1. Mendefinisikan prosedur

Pada dasarnya struktur prosedur sama dengan struktur algoritma yang sudah kita kenal, yaitu ada judul(header), deklarasi (keterangan) dan deskripsi (uraian).



Gambar 2. Struktur Prosedur

2. Pemanggilan prosedur

Prosedur bukan program yang berdiri sendiri, ia tidak dapat dieksekusi secara langsung. Isi prosedur hanya dapat diakses dengan cara memanggil namanya dari program pemanggilanya (program utama atau modul lain). Bentuk umum cara pemanggilan prosedur adalah sbb :

```
NAMA_PROSEDUR;
```

Ketika nama prosedur dipanggil maka kendali program berpindah secara otomatis ke prosedur yang dipanggil tersebut. Setelah isi prosedur selesai dieksekusi, kendali program kembali ke instruksi sesudah pemanggil prosedur.

3. Nama global dan nama lokal

Nama-nama indentifier yang dideklarasikan di dalam bagian deklarasi prosedur, hanya dikenal di dalam badan prosedur itu saja. Nama-nama indentifier tersebut dikatakan bersifat "lokal" dan hanya dapat dikenali didalam prosedur itu sendiri, sedangkan nama-nama yang dideklarasikan di program utama bersifat "global" dan dapat dikenali didalam prosedur.

4. Parameter

Kebanyakan program memerlukan pertukaran informasi antara prosedur/fungsi dan titik dimana ia dipanggil. Penggunaan parameter menawarkan mekanisme pertukaran informasi tersebut. Tiap item data ditransfer antara parameter aktual dan parameter formal yang bersesuaian. Parameter aktual adalah parameter yang disertakan pada waktu pemanggilan prosedur, sedangkan parameter formal adalah parameter yang dideklarasikan didalam header prosedur itu sendiri. Ketika prosedur itu dipanggil, parameter aktual menggantikan parameter formal. Tiap parameter aktual berpasangan dengan parameter formal yang bersesuaian.

contoh :

```
procedure SATU(input x,y : integer)
{ cth procedure dgn parameter formal berjenis parameter masukan,
```

K.Awal : nilai x dan nilai y sudah terdefenisi

K.Akhir : nilai x dan y masing-masing dinaikkan satu, lalu dicetak ke piranti keluaran }

DEKLARASI:

{ TIDAK ADA}

DESKRIPSI:

$x \leftarrow x + 1$

$y \leftarrow y + 1$

write(x)

write(y)

5. Algoritma PQR

{ cth program utama yg memanggil prosedur SATU }

DEKLARASI:

a, b : real

procedure SATU(input x, y : integer)

{cth prosedur dgn parameter formal berjenis parameter masukan}

DESKRIPSI:

SATU(4,10)

read(a, b)

SATU(a, b)

SATU(a+2, 20)

- Masukan
- Keluaran
- Masukan / keluaran
- Prosedur tanpa parameter
- Prosedur dengan parameter
- Parameter masukan dan parameter keluaran

B. FUNCTION (FUNGSI)

Fungsi merupakan potongan program yang tidak dapat berdiri sendiri, karena fungsi memberikan hasil berupa suatu nilai dan memerlukan suatu tempat untuk menampung nilai tersebut. Ada fungsi standart (fungsi terpasang) dan ada fungsi buatan (defined function).

Contoh-contoh fungsi dan prosedur standart antara lain :

Fungsi Numerik :

1. ABS(n)

Memberikan nilai mutlak (harga positif) dari bilangan n.

N : Real atau Integer;

Hasil : sesuai dengan tipe-n

2. FRAC(n)

Memperoleh bilangan pecahan dari suatu bilangan pecahan (real).

N : Real

Hasil : Real

3. INT(n)

Memperoleh bilangan bulat dari suatu bilangan pecahan (real).

N : Real

Hasil : Real

4. TRUNC(n)
Memperoleh bilangan bulat dari suatu bilangan pecahan (real).
N : Real
Hasil : LongInt
5. ROUND(n)
Membulatkan bilangan pecahan (real) atau pembulatan.
N : Real
Hasil : LongInt
6. EXP(n)
Memperoleh nilai exponential (e^x (e =bilangan natural)).
N : Real
Hasil : Real
7. LN(n)
Memperoleh nilai logaritma natural.
N : Real atau Integer
Hasil : Real
8. SQR(n)
Memperoleh nilai kwadrat
N : Real atau Integer
Hasil : sesuai tipe argumen - n
9. SQRT(n)
Memperoleh nilai akar kwadrat (\sqrt{n})
N : Real atau Integer
Hasil : Real
10. PI
Memperoleh nilai konstanta bernilai 3.14
Hasil : Real
11. SIN(n)
Memperoleh nilai sinus - n
N : Real
Hasil : Real
12. COS(n)
Memperoleh nilai cosinus - n
N : Real
Hasil : Real
13. ARCTAN(n)
Memperoleh nilai arcus tangen - n
N : Real
Hasil : Real
14. ODD(n)
Menentukan apakah n bilangan ganjil atau tidak (True/False)
N : Word
Hasil : Bolean
15. RANDOM(n)
Memperoleh bilangan acak
N : Word
Hasil : Real $\rightarrow 0 \leq \text{hasil} < n$
16. RANDOM
Menentukan bilangan acak


$$360^\circ = 2\pi \text{ radian}$$
$$1^\circ = \pi/360^\circ = \pi/180^\circ$$

Hasil : Real $\rightarrow 0 \leq \text{hasil} < 1$

Prosedur dan fungsi Library standart untuk string:

1. CHR(n)
Memperoleh karakter dengan nomor ASCII-n
N : Byte
Hasil : Char
2. ORD(kar)
Memperoleh nomor ASCII dari karakter *kar*
kar : Char
Hasil : Byte
3. CONCATE(string1, string2, string3, ...)
Menggabungkan string1, string2, string3, ...
string1, string2, string3, ... : String
Hasil : String
4. LENGTH(st)
Memperoleh jumlah karakter dari - st
St : String
Hasil : Byte
5. POS(st1, st2)
Menentukan posisi *st1* didalam *st2*
St1 : Char atau String
Hasil : Byte
6. UPCASE(kar)
Mengubah karakter huruf kecil menjadi huruf besar (berlaku hanya 1 karakter awal)
kar : Char
Hasil : Char
7. COPY(st, p, n)
Menyalin isi *st* dimulai posisi *p* sebanyak *n* karakter
st : String
p,n : Byte
Hasil : String
8. READKEY
Meminta penekanan 1 tombol. Tombol yang diberikan tidak tampil di layar
Hasil : Char
9. DELETE(st, p, n)
Menghapus isi *st* dimulai posisi *p* sebanyak *n* karakter
st : String
p,n : Byte
Hasil : String
10. INSERT(st1, st2, p)
Menyisip *st1* kedalam *st2* pada posisi *p*. Hasilnya disimpam di dalam *st2*.
st1, st2 : String
p : Byte
Hasil : String
11. STR(Bil[:k[:d]], BilSt)
Mengubah bilangan numerik *Bil* menjadi bilangan string dengan format tertentu. Hasil disimpan dalam *BilSt*.
k, d : Byte dimana K=lebar kolom, D=lebar desimal

BilSt : String

12. VAL(BilSt, Hasil, ValErr)

Mengubah bilangan string menjadi bilangan numerik.

BilSt : String (angka)

Hasil : Real atau Integer

ValErr : Word

```
{Program Kon1.Pas}
Program Konversil;
Uses Crt;

Var a : string;
    b : real;
    err: integer;
Begin
  ClrScr;
  a := '1234.5';
  val(a,b,err);
  if err<>0 then
    write('salah')
  else
    write(b+2:10:2);

  Readln;
End.
```

Hasilnya 1236.5

```
{Program Kon2.Pas}
Program Konversi2;
Uses Crt;

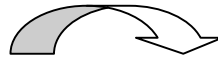
Var a : string;
    b : real;
    err: integer;
Begin
  ClrScr;
  a := 'abc';
  val(a,b,err);
  if err<>0 then
    write('salah')
  else
    write(b+2:10:2);

  Readln;
End.
```

Hasilnya : salah

C. USER DEFINED FUNCTION

Seperti halnya prosedur, struktur fungsi sama dengan struktur algoritma yaitu ada **header** yang berisi nama fungsi dan spesifikasi fungsi, bagian **deklarasi**, dan **badan fungsi**. Setiap fungsi mempunyai nama yang unik serta daftar parameter formal (jika ada).



<p><u>Function</u> Nama_Fungsi(dtr parameter formal) { keterangan spesifikasi singkat tentang fungsi} Deklarasi { semua nama yang dipakai dalam fungsi dan hanya berlaku lokal dalam fungsi} Deskripsi { badan fungsi berisikan kumpulan instruksi }</p>	<p><u>Function</u> F(<u>input</u> x:real) of real; { mengembalikan nilai $F(x)=2x^2+5x-8$ } Deklarasi { tidak ada } Deskripsi <u>return</u> $2*x*x+5x-8$</p>
---	--

Kata cadangan FUNCTION mengawali bagian deklarasi fungsi diikuti oleh identifier yang merupakan nama dari fungsinya dan secara optional dapat diikuti oleh kumpulan parameter, tipe dari fungsinya dan diakhiri dengan titik koma.

Contoh deklarasi fungsi :

```
FUNCTION Pangkat(X, Y : real) : Real;
```

Pemanggilan Fungsi

Fungsi diakses dengan cara memanggil namanya dari program pemanggil diikuti dengan daftar parameter aktual (bila ada).

Contoh pemanggilan fungsi (menggunakan notasi algoritmik):

```
peubah ← Nama_Fungsi(daftar parameter aktual)
```

atau

```
Write(Nama_Fungsi(daftar parameter aktual))
```

Contoh dalam program :

```
{Nama Program : Fungsil.Pas }
Program Faktorial;
Uses Crt;

Var f,x : Integer;

Function Fak(x:integer) :integer;
Var f:integer;
    u:byte;
Begin
    f:=1;
    for u:=1 to x do
        f:=f*u;
    fak:=f;
End;

Begin { -- blok utama program -- }
    ClrScr;
```

pemanggilan fungsi Fak dgn parameter aktual x

```
write('Masukan berapa faktorial : ');
readLn(x);
write(x,' faktorial adalah ',Fak(x));
readLn;
End.
```

```
-----
{ Nama Program : Fungsi2.Pas}
Program Contoh_Fungsi;
Uses Crt;
Var
  X, Y, Z : Integer;

{ ----- Bag. Prosedur ----- }
Procedure Inp_Bil(A, B : Integer);
Begin
  Write('Input Bilangan I : ');ReadLn(X);
  Write('Input Bilangan II : ');ReadLn(Y);
End;

{ ----- Bag. Fungsi ----- }
Function Tam_Bil(A, B : Integer) : Integer;
Begin
  Tam_Bil:=A+B;
End;

{ ----- Program Utama ----- }
Begin
  Clrscr;
  Inp_Bil(X, Y);
  Write('Hasil dari      : ',X,' + ',Y,' = ',Tam_Bil(X, Y));
  ReadLn;
End.
```

pemanggilan fungsi
Tam_Bil dgn parameter
aktual x dan y

D. 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.