



ISA-105

Algorithm & Pemrograman

Sofia Umaroh, S.Pd. M.T.
Sistem Informasi ITENAS

Today's materials

- Fungsi dan Prosedur dalam Bahasa C
- Variabel Global
- Variabel Lokal
- Tabulasi

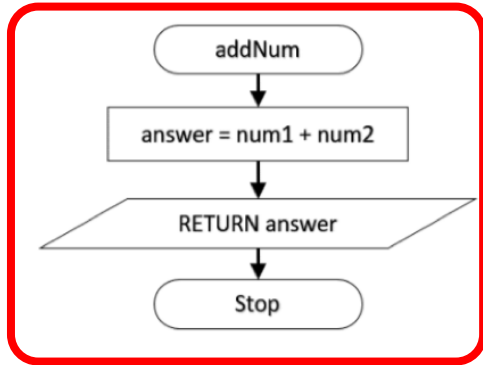
1

**Sub Program:
Fungsi & Prosedur**

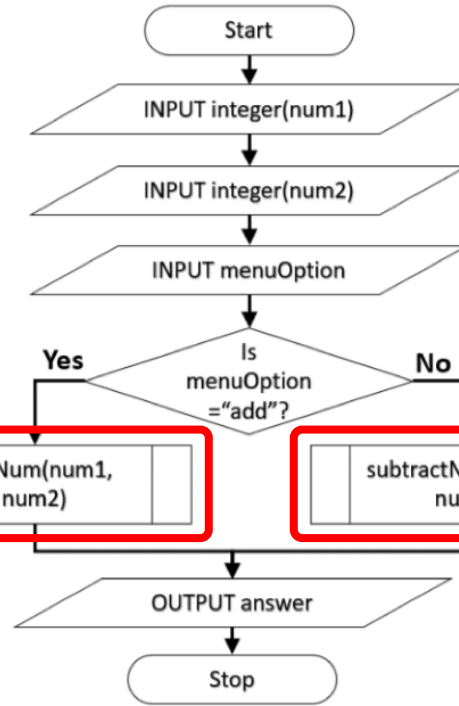
Definisi Sub-Program

- Sub-Program adalah suatu program terpisah dalam blok sendiri yang berfungsi sebagai sub-program (modul program)
- Sub-Program merupakan sebuah program kecil untuk memproses sebagian dari pekerjaan program utama.
- Dengan menggunakan Sub-Program, program bisa disusun secara lebih terstruktur (lebih modular) dan lebih efektif

Flowchart Sub Program

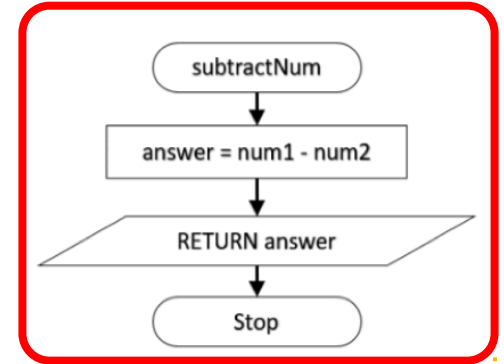


Sub Program: **addNum**



Program utama

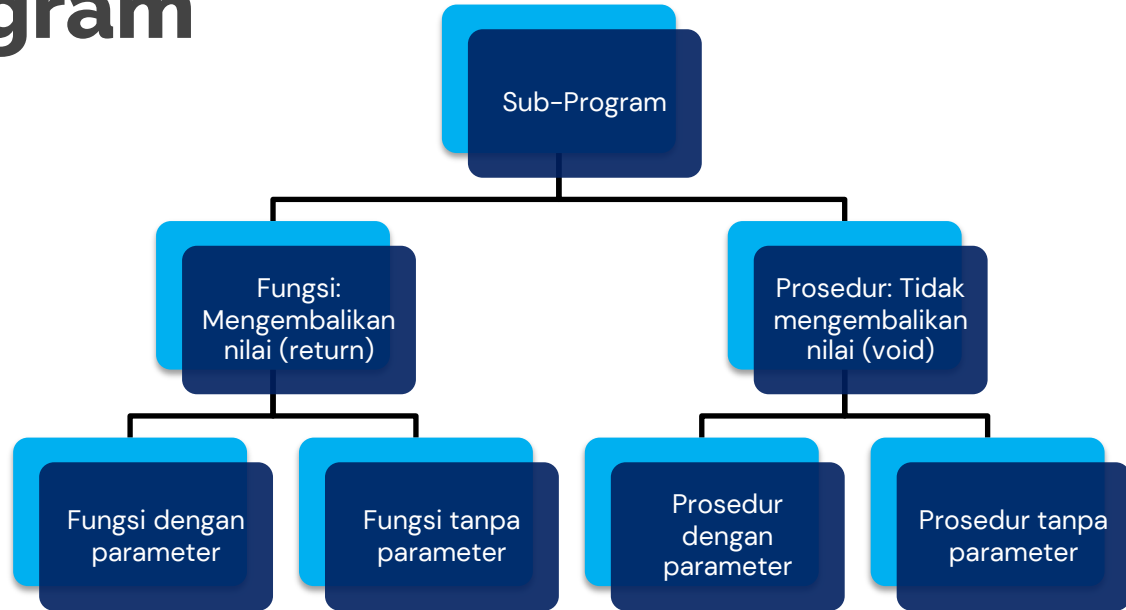
Sub Program: **subtractNum**



Jenis Sub-Program

- Sub-Program terbagi menjadi 2:
 - Prosedur, sub-program yang tidak mengembalikan nilai (*void*)
 - Fungsi, sub-program yang mengembalikan nilai (*return*)

Peta Sub-Program



Variabel Global dan Lokal

- **Variabel Lokal:** variabel yang dideklarasikan dalam suatu fungsi, dan hanya bisa diakses atau dikenali dari dalam fungsi itu sendiri
- **Variabel Global:** variabel yang dideklarasikan di luar blok fungsi, dan bisa diakses atau dikenali dari fungsi manapun

1

Sub-Program:
Prosedur

Definisi Prosedur

- Prosedur diawali dengan kata kunci **Procedure (pascal)** atau **void** (Bahasa C) saat deklarasi prosedur.
- Prosedur dipanggil dan digunakan di dalam blok program yang lainnya dengan menyebutkan nama prosedurnya.
- Prosedur banyak digunakan pada program yang terstruktur, karena :
 1. Merupakan penerapan konsep program modular, yaitu memecah program yang rumit menjadi program bagian yang lebih sederhana
 2. Untuk instruksi yang sering dilakukan berulang-ulang, cukup dituliskan sekali saja dalam prosedur dan dapat dipanggil atau dipergunakan sewaktu-waktu bila diperlukan.

Deklarasi Prosedur

- Sebagaimana halnya sebuah program, procedure juga memiliki header dan block.
- Prosedur tidak mengembalikan nilai (tidak ada return)
- Jika kita menggunakan procedure dalam suatu program, maka procedure tersebut harus dituliskan pada bagian deklarasi.
- Bentuk Umum header suatu procedure adalah :

```
procedure nama_prosedur;  
var  
    {deklarasi variable jika diperlukan}  
begin  
    {statement procedure}  
end;
```

Contoh Prosedur

- Misal akan dibuat suatu procedure untuk mencetak kalimat **Hello world** setiap kali prosedur dipanggil

Pascal	Bahasa C
<pre>{header} procedure hello; begin writeln('hello world'); end.</pre>	<pre>/* header */ void hello(){ printf("Hello world"); };</pre>

Memanggil Prosedur

- Prosedur harus dipanggil di program utama untuk dapat dijalankan
- Berikut contoh program utama yang memanggil prosedur **hello**

Pascal	Bahasa C
<pre>program contoh_1; procedure hello; begin writeln('Hello world'); end. {Program utama} begin hello; hello; end.</pre> <p>Memanggil prosedur hello</p>	<pre>/* Program utama */ int main(){ hello(); hello(); return 0; } /* Implementasi void */ void hello(){ printf("Hello world\n"); };</pre> <p>Memanggil prosedur hello</p>
<p>Output: Hello world Hello world</p>	<p>Output: Hello world Hello world</p>

Contoh Prosedur dengan variabel

- Misal akan dibuat suatu procedure untuk menentukan bilangan bulat terbesar diantara tiga bilangan bulat, maka procedure tersebut adalah sebagai berikut :

Pascal	Bahasa C
<pre>{header} procedure maksimum; var max : integer; begin if a > b then max := a else max := b; if c > max then max := c; writeln(max); end.</pre>	<pre>/* header */ void maksimum(){ int max; if (a > b) max = a else max = b; if (c > max) max = c; printf("%d",max); };</pre>

Memanggil Prosedur

- Prosedur harus dipanggil di program utama untuk dapat dijalankan
- Berikut contoh program utama yang memanggil prosedur

Memanggil
prosedur
maksimum ←

Pascal	Bahasa C
<pre>program contoh_1; var a,b,c : integer; procedure maksimum; var max : integer; begin if a>b then max := a else max := b; if c>max then max := c; writeln(max); end; {Program utama} begin readln(a,b,c); maksimum; end.</pre>	<pre>Int main(){ int a,b,c; maksimum(); → Memanggil return 0; prosedur maksimum } /* Implementasi void */ void maksimum(){ int max; if (a > b) max = a else max = b; if (c > max) max = c; printf("%d",max); };</pre>

Variabel Lokal dan Global

Variabel Lokal

- Variabel di dalam suatu sub-program **bersifat lokal**, artinya hanya dapat digunakan pada sub-program bersangkutan saja
- Pada contoh prosedur maksimal, **variable max adalah var local**, hanya dapat digunakan pada prosedur maksimum saja
- Tidak ada return

Pascal	Bahasa C
<pre>procedure maksimum; var max : integer; {lokal} begin if a > b then max := a else max := b; if c > max then max := c; writeln(max); end.</pre>	<pre>void maksimum(){ int max; //var lokal if (a > b) max = a else max = b; if (c > max) max = c; printf("%d",max); };</pre>

Variabel Global

- Agar variable dapat diakses oleh semua sub-program, **variable harus global**
- Misalnya variable A dan B ingin agar dapat diakses oleh prosedur satu dan program utama, maka deklarasikan **var A dan B di global**
- Var C dan D hanya dapat diakses di prosedur satu (var lokal)

<pre>Var max : A,B; {var global} Procedure satu Var C,D:integer {lokal} Begin ... End</pre>	<pre>#include <stdio.h> int A,B; // var global /* Program Utama */ int main(){ ... return 0; }</pre>
<pre>{program utama} BEGIN ... END.</pre>	<pre>void satu(){ int C,D; //lokal ... }</pre>

**Bagaimana Jika Prosedur
Memerlukan Variable Yang
Dikirimkan Dari Program Utama?**

Prosedur Dengan Parameter

Jenis Prosedur (void)

- Void dengan parameter
 - **void nama(parameter1, parameter2, ...){**
 - **//statement**
 - **}**
- Fungsi tanpa parameter
 - **void nama(){**
 - **//statement**
 - **}**

Prosedur dengan Parameter

- Parameter adalah **variable yang dikirim/dilewatkan** pada sub-program
 - contoh prosedur max dengan 3 parameter:
 - **procedure max(a,b,c:integer);**
- Parameter yang dikirim dari program utama ke prosedur disebut **actual parameter**
 - Contoh: **max (8, 10, 3);** [nilai 8, 10, dan 3 adalah **actual parameter** dikirim dari program utama]
 - Nilai 8 akan menggantikan var a pada prosedur max
- Parameter yang dituliskan pada header prosedur disebut **formal parameter**.
 - Contoh: **max (a, b, c: integer);** [var a, b, c pada header max adalah **formal parameter**]

Contoh Void dengan parameter

- Void dengan parameter
 - `void nama(parameter1, parameter2, ...){`
 - `//statement`
 - `}`

- Contoh:

```
• void addition(int a, int b){ 2 variabel  
• int r; parameter formal  
• r = a+b; variabel lokal  
• printf("jumlah = %d",r);  
• }
```

Statement

prosedur mencetak ke layar hasil jumlah, **tidak mengembalikan nilai**

Contoh Void tanpa parameter

- Void tanpa parameter
 - `void nama(){`
 - `//statement`
 - `}`

- Contoh:

- `void addition(){`
- `int a, b; → variabel lokal`
- `scanf("%d",&a);`
- `scanf("%d",&b);`
- `printf("jumlah = %d",a+b);`
- `}`

→ prosedur mencetak ke layar hasil jumlah; **tidak mengembalikan nilai**

Prosedur dengan Parameter

Pascal	Bahasa C
<pre>program contoh_1; var a,b,c : integer; procedure maksimum (a,b,c: integer); var max : integer; begin if a>b then max := a else max := b; if c>max then max := c; writeln(max); end; {Program utama} begin readln(a,b,c); maksimum(a,b,c); end.</pre> <p>a,b,c: integer → formal parameter</p> <p>a,b,c → Actual parameter</p>	<pre>Int main(){ int a,b,c; maksimum(a,b,c); return 0; } /* Implementasi void */ void maksimum(int a,b,c){ int max; if (a > b) max = a else max = b; if (c > max) max = c; printf("%d",max); };</pre> <p>a,b,c → Actual parameter</p> <p>int a,b,c → formal parameter</p> <p>max → Actual parameter</p>

Latihan: tentukan output!

Pascal	Bahasa C
<pre>procedure hitung(a,b : integer); var c : integer; begin c := a * b; writeln('nilai c = ',c) end; var x,y : integer; {program utama} begin readln(x,y); hitung(x,y); end.</pre>	<pre>Int main(){ int x,y scanf("%d",&x); scanf("%d",&y); hitung(x,y); } /* Implementasi void */ void hitung(int a,b){ int c; C = a * b printf("%d",max); };</pre>

Latihan: tentukan outputnya manual!

```
Var
    a,b,c,d,e,f : integer {var global}

procedure bc (c,d: integer)
begin
    a <- a + 2
    b <- b + 4
    c <- c + 3
    e <- d + 1
    writeln("bc: ",a,b,c,d,e)
end;

{program utama}
begin
    a <- 1
    b <- 3
    c <- 2
    d <- 4
    e <- 5
```

```
...

for f = 1 to 3 do
begin
    b <- b + f + 2
    c <- c + f + 4
    bc(e,f)
    write ("G2: ",a, b, c, d,
e)
{endfor}
End.
```