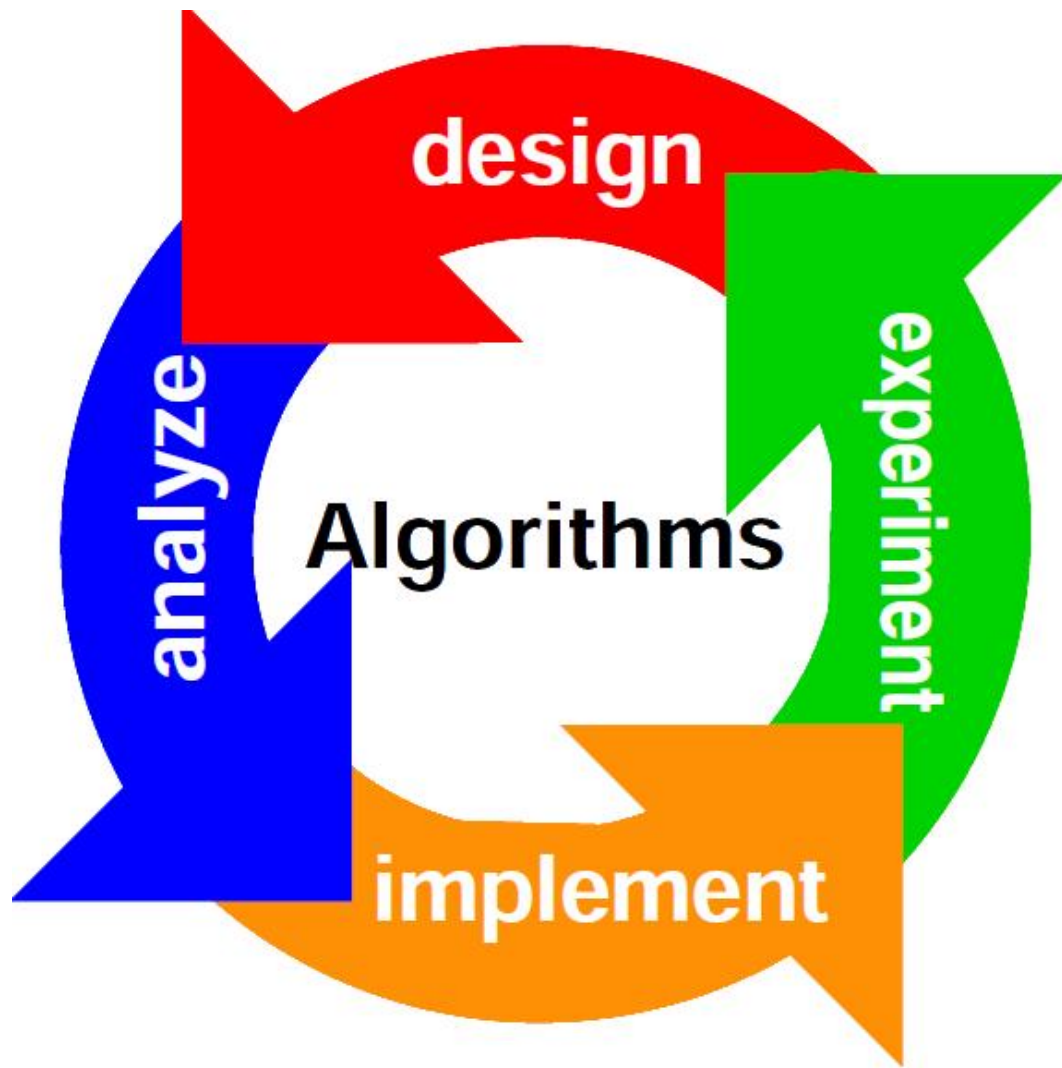


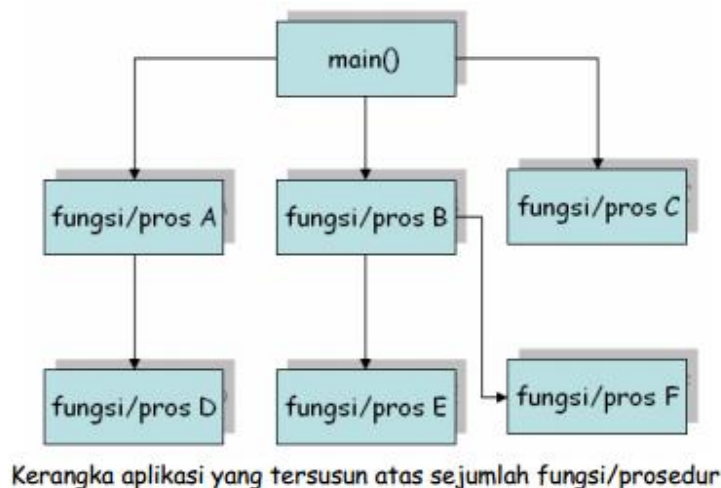
# ALGORITMA DAN PEMROGRAMAN 2



TEKNIK PERANGKAT LUNAK  
FT - UNPAM

## PROSEDUR dan FUNGSI

Baik prosedur maupun fungsi merupakan suatu modul sub program yang mengerjakan tugas/aktivitas yang spesifik. Perbedaan diantara keduanya adalah sebuah fungsi akan mengembalikan suatu nilai tertentu ke modul/atau fungsi lain/prosedur lain yang memanggilnya. Tujuan utama dari prosedur atau fungsi adalah untuk membuat suatu aplikasi agar dapat dipecah menjadi sebuah bagian yang dapat dikelola dengan lebih mudah.



### 1. Pendefinisian prosedur

- menuliskan nama prosedur
- mendeklarasikan nama-nama konstanta, variable dan tipe
- menjabarkan rangkaian aksi yang dilakukan

Setiap prosedur mempunyai nama yang unik Prosedur terdiri dari:

- bagian judul(header) , terdiri atas nama prosedur dan deklarasi parameter (jika ada)
- bagian deklarasi, untuk mengumumkan nama-nama
- bagian algoritma/deskripsi, disebut badan prosedur

Parameter adalah nama-nama variabel yang dideklarasikan pada bagian header.

Parameter :

- aktual (argumen) : parameter yang disertakan pada saat pemanggilan prosedur
- formal : parameter yang dideklarasikan pada bagian header prosedur

Contoh notasi algoritmik sebuah prosedur:

Notasi algoritmik	Kode bahasa C++
<pre> <u>procedure</u> Luas { K. Awal   : nilai panjang dan lebar diberikan} { K. Akhir   : luas segi empat tercetak} DEKLARASI     panjang      : <u>integer</u>     lebar         : <u>integer</u>     HitungLuas    : <u>integer</u> DESKRIPSI:     HitungLuas ← panjang * lebar     <u>write</u>(HitungLuas) </pre>	<pre> void Luas(void) {     int panjang, lebar, HitungLuas;     cin&gt;&gt;panjang;     cin&gt;&gt;lebar;     HitungLuas = panjang*lebar;     cout&lt;&lt;HitungLuas; } </pre>

## 2. Pemanggilan Prosedur

Prosedur bukan program yang berdiri sendiri, jadi tidak dapat dieksekusi secara langsung. Prosedur diakses dengan cara memanggil namanya dari program pemanggil. Prosedur bisa memiliki parameter atau tidak. Jika tidak memiliki parameter cukup dengan menuliskan namanya. Ketika sebuah prosedur dipanggil, kendali program secara otomatis pindah ke prosedur tersebut. Seluruh instruksi dalam badan prosedur tersebut akan dilaksanakan. Setelah semua instruksi selesai dilaksanakan, kendali program berpindah secara otomatis kembali ke instruksi sesudah pemanggilan prosedur dalam program pemanggil.

Agar nama prosedur dikenal oleh program pemanggil, maka dalam program pemanggil harus dideklarasikan prototipe prosedur tersebut. Contoh pemanggilan prosedur dari contoh sebelumnya:

Notasi algoritmik	Kode bahasa C++
<pre> <u>ALGORITMA</u> HitungLuas DEKLARASI     <u>procedure</u> Luas DESKRIPSI:     Luas  <u>procedure</u> Luas DEKLARASI     panjang      : <u>integer</u>     lebar         : <u>integer</u>     HitungLuas    : <u>integer</u> DESKRIPSI:     Read(panjang,lebar)     HitungLuas ← panjang * lebar     <u>write</u>(HitungLuas) </pre>	<pre> #include &lt;iostream.h&gt; void Luas(); void main() {     Luas(); }  void Luas() {     int panjang, lebar, HitungLuas;     cin&gt;&gt;panjang;     cin&gt;&gt;lebar;     HitungLuas = panjang*lebar;     cout&lt;&lt;HitungLuas; } </pre>

### 3. Lingkup variabel

Lingkup variabel menjelaskan bagaimana sifat suatu variabel didalam suatu prosedur/fungsi.

1. Variabel otomatis/lokal
2. Variabel eksternal/global
3. Variabel statis

4.

#### 3.1. Variabel otomatis/lokal

Variabel otomatis bersifat lokal terhadap sebuah prosedur/fungsi. Variabel seperti ini tidak dikenal diluar prosedur/fungsi yang mendeklarasikannya.

<pre>#include&lt;iostream.h&gt; void prosX(void); //prototipe prosedur void main(void) {     int bilX=50;     cout&lt;&lt;"Sebelum memanggil prosedur"&lt;&lt;endl;     cout&lt;&lt;"bilX = "&lt;&lt;bilX&lt;&lt;endl;     prosX();     cout&lt;&lt;"Setelah memanggil prosedur"&lt;&lt;endl;     cout&lt;&lt;"bilX = "&lt;&lt;bilX&lt;&lt;endl; }  void prosX(void) {     int bilX=100;     cout&lt;&lt;"Dalam prosedur"&lt;&lt;endl;     cout&lt;&lt;"bilX = "&lt;&lt;bilX&lt;&lt;endl; }</pre>	<p>Hasilnya :</p> <p>Sebelum memanggil prosedur bilX = 50 Dalam prosedur bilX = 100 Setelah memanggil prosedur bilX = 50</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------

#### 3.2. Variabel eksternal

Didefinisikan diluar prosedur/fungsi, sehingga dikenal oleh seluruh prosedur/fungsi.

<pre>#include&lt;iostream.h&gt; void ubahBilX(void); //prototipe prosedur int bilX; void main(void) {     bilX=50;     cout&lt;&lt;"Sebelum memanggil prosedur"&lt;&lt;endl;     cout&lt;&lt;"bilX = "&lt;&lt;bilX&lt;&lt;endl;     ubahBilX();     cout&lt;&lt;"Setelah memanggil prosedur"&lt;&lt;endl;     cout&lt;&lt;"bilX = "&lt;&lt;bilX&lt;&lt;endl; }  void ubahBilX(void) {     cout&lt;&lt;"Dalam prosedur"&lt;&lt;endl;     cout&lt;&lt;"bilX = "&lt;&lt;bilX&lt;&lt;endl;     bilX=100; }</pre>	<p>Hasilnya :</p> <p>Sebelum memanggil prosedur bilX = 50</p> <p>Dalam prosedur bilX = 50</p> <p>Setelah memanggil prosedur bilX = 100</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------

### 3.3. Variabel Statis

Sifat :

- Variabel hanya bisa diakses oleh prosedur/fungsi yang mendeklarasikannya
- Variabel tidak hilang saat eksekusi prosedur/fungsi berakhir
- Inisialisasi dalam deklarasi hanya dilakukan sekali selama aplikasi berjalan

<pre>#include&lt;iostream.h&gt; void Pencacah(void); //prototipe prosedur void main(void) {     Pencacah();     Pencacah();     Pencacah(); }  void Pencacah(void) {     static int jumlah=1;     cout&lt;&lt;"Prosedur telah dipanggil = "&lt;&lt;jumlah&lt;&lt;" kali"&lt;&lt;endl;     jumlah++; }</pre>	<p>Hasilnya :</p> <p>Prosedur telah dipanggil 1 kali Prosedur telah dipanggil 2 kali Prosedur telah dipanggil 3kali</p> <p><i>Coba kata <b>static</b> dihilangkan, bagaimana hasilnya?</i></p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 4. Parameter/Argumen

Kebanyakan program memerlukan pertukaran informasi antara prosedur dan pemanggilnya. Penggunaan parameter menawarkan mekanisme pertukaran tersebut. Prosedur dengan parameter diakses dengan cara memanggil prosedur tersebut beserta parameternya.

Aturan yang harus diperhatikan dalam korespondensi satu-satu antara parameter formal dan parameter aktual

adalah:

- Jumlah parameter aktual pada pemanggilan prosedur harus sama dengan jumlah parameter formal pada deklarasi prosedurnya
- Tiap parameter aktual harus bertipe sama dengan tipe parameter formal yang bersesuaian
- Parameter aktual diekspresikan sesuai dengan jenis parameter formal (penjelasan lihat dibawah)

Berdasarkan maksud penggunaannya, terdapat tiga jenis parameter formal:

- a. parameter masukan (input parameter)
- b. parameter keluaran (output parameter)
- c. parameter masukan/keluaran (input/output parameter)

#### 4.1. Parameter masukan

Parameter masukan : parameter yang nilainya berlaku sebagai masukan untuk prosedur. Sering disebut juga

sebagai parameter nilai (value parameter atau parameter by value

Contoh

<p><b>ALGORITMA</b> HitungLuas</p> <p><b>DEKLARASI</b></p> <p>a,b : integer</p> <p><u>procedure</u> Luas(<u>input</u> panjang, lebar : <u>integer</u>)</p> <p><b>DESKRIPSI:</b></p> <p><u>read</u>(a,b)</p> <p>Luas(a,b)</p> <p><u>procedure</u> Luas(<u>input</u> p,l:<u>integer</u>)</p> <p><b>DEKLARASI</b></p> <p>luas : <u>integer</u></p> <p><b>DESKRIPSI:</b></p> <p>luas ← p * l</p> <p><u>write</u>(luas)</p>	<pre>#include&lt;iostream.h&gt; void Luas(int a,int b);//prototipe prosedur void main(void) {     int panjang=5,lebar=3;     Luas(panjang,lebar); }  void Luas(int p,int l) {     int luas;     luas=p*l;     cout&lt;&lt;"Luas = "&lt;&lt;luas&lt;&lt;endl; }</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Nilai parameter aktual diisikan kedalam parameter formal yang bersesuaian.

Perubahan nilai dalam badan

prosedur tidak mengubah nilai parameter aktual. Karena yang dipentingkan adalah nilainya, maka nama

parameter aktual boleh berbeda dengan nama parameter formal yang bersesuaian.



## 4.2. Parameter Keluaran

Parameter keluaran : parameter yang menampung keluaran yang dihasilkan oleh prosedur yang akan digunakan oleh program pemanggil.

<p><b>ALGORITMA</b> HitungLuas</p> <p><b>DEKLARASI</b></p> <p>a,b,L : <u>integer</u></p> <p><u>procedure</u> Luas(<u>input</u> p,l : <u>integer</u>, <u>output</u> luas : <u>integer</u>)</p> <p><b>DESKRIPSI:</b></p> <p><u>read</u>(a,b)</p> <p>Luas(a,b,L)</p> <p><u>write</u>(L)</p> <p><u>procedure</u> Luas(<u>input</u> p,l : <u>integer</u>, <u>output</u> luas : <u>integer</u>)</p> <p><b>DEKLARASI</b></p> <p>{tidak ada}</p> <p><b>DESKRIPSI:</b></p> <p>luas ← p * l</p>	<pre>#include&lt;iostream.h&gt; void hitung(int ,int, int * ); void main(void) {     int panjang,lebar,luas;     cout&lt;&lt;"Panjang = ";cin&gt;&gt;panjang;     cout&lt;&lt;"Lebar = ";cin&gt;&gt;lebar;cout&lt;&lt;endl;     hitung(panjang,lebar,&amp;luas);     cout&lt;&lt;"Luas = "&lt;&lt;luas&lt;&lt;endl; }  void hitung(int A, int B, int *luas) {     *luas=A*B; }</pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Karena nama parameter merupakan suatu lokasi di memori maka bila didalam prosedur, parameter aktual diisi suatu nilai, nilai ini akan tetap berada didalam parameter aktual meskipun prosedur selesai dilaksanakan. Jadi setelah pemanggilan, parameter aktual berisi suatu nilai yang merupakan keluaran dari prosedur tersebut.

## 4.3. Parameter Masukan/Keluaran

Parameter masukan/keluaran : parameter yang berfungsi sebagai masukan sekaligus keluaran bagi prosedur tersebut.

Pada kebanyakan aplikasi kadang informasi harus dikirim dalam dua arah sehingga prosedur harus dapat mengakomodasi baik masukan dari dan keluaran ke blok program pemanggil.  
Contoh:

<p><b>ALGORITMA SelisihXY</b></p> <p><b>DEKLARASI</b>  X,Y,Z : <u>integer</u>  <u>procedure</u> Tukar(<u>input/output</u> A,B : <u>integer</u>)</p> <p><b>DESKRIPSI:</b>  <u>read</u>(X,Y)  <u>if</u> X &lt; Y <u>then</u>      Tukar(X,Y)  <u>endif</u>  Z ← X - Y  <u>write</u>(Z)</p> <p><u>procedure</u> Tukar(<u>input/output</u> A,B : <u>integer</u>)</p> <p><b>DEKLARASI</b>  tmp : <u>integer</u></p> <p><b>DESKRIPSI:</b>  tmp ← A  A ← B  B ← tmp</p>	<pre>#include&lt;iostream.h&gt; void Tukar(int *, int *); void main(void) {     int X,Y,Z;     cout&lt;&lt;"X = ";cin&gt;&gt;X;     cout&lt;&lt;"Y = ";cin&gt;&gt;Y;     if(X&lt;Y)         Tukar(&amp;X,&amp;Y);     Z=X-Y;     cout&lt;&lt;Z&lt;&lt;endl; }  void Tukar(int *A, int *B) {     int tmp;     tmp=*A;     *A=*B;     *B=tmp; }</pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Akibat penggunaan parameter masukan/keluaran, bila parameter aktual diubah nilainya dalam badan prosedur maka sesudah pemanggilan prosedur, nilai parameter aktual di titik pemanggilan juga berubah.

## 5. Translasi notasi algoritmik Prosedur kedalam notasi bahasa C

Aturan dalam translasi :

- Pendefinisian prosedur ditulis diluar blok program utama, kecuali jika direalisasikan sebagai file include. Prosedur umumnya diletakan setelah blok main(), sedangkan deklarasi prosedur ditulis sebelum blok main() sebagai prototipe. Jika pendefinisian prosedur ditulis sebelum blok main() maka pendeklarasian prototype tidak diperlukan lagi.
- Dalam bahasa C tidak dikenal istilah prosedur. Semua modul program adalah fungsi. Prosedur adalah fungsi yang tidak mengembalikan nilai apapun. Karena itu nilai kembali untuk prosedur adalah void (artinya kosong).
- Bila prosedur tidak memiliki parameter maka tanda kurung “(“ dan “)” tetap ditulis setelah nama prosedur tersebut.
- Semua parameter dalam bahasa C adalah parameter masukan. Oleh karena itu, semua argumen parameter aktual dilewatkan sebagai “by value” , artinya nilai