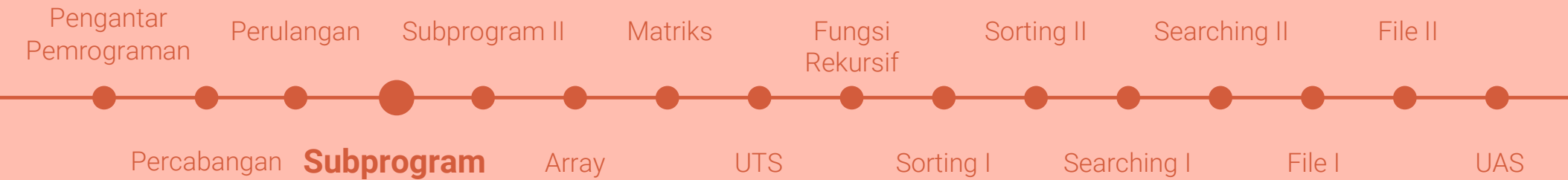


DASAR PEMROGRAMAN

Pertemuan IV





Tujuan

- Mahasiswa memahami makna dan kegunaan subprogram dalam bentuk fungsi dan prosedur
- Mahasiswa dapat menggunakan notasi fungsi dan prosedur dengan benar dan menggunakannya dalam program
- Mahasiswa dapat membuat program dengan menggunakan fungsi dan prosedur





Materi

SUBPROGRAM

FUNGSI

CAKUPAN VARIABEL

PARAMETER FORMAL DAN AKTUAL





Baca nomor kartu
Periksa PIN
Tampilkan pilihan menu jika PIN sesuai



Cari nomor rekening berdasarkan nomor kartu
Cari saldo berdasarkan nomor rekening
Tampilkan saldo



Cari nomor rekening berdasarkan nomor kartu
Cari saldo berdasarkan nomor rekening
Periksa apakah saldo > jumlah penarikan
Keluarkan uang jika saldo > jumlah penarikan



Cari nomor rekening berdasarkan nomor kartu

Cari saldo berdasarkan nomor rekening

Periksa apakah saldo > jumlah pembayaran

Bayar ke nomor tujuan jika saldo > jumlah pembayaran



Cari nomor rekening berdasarkan nomor kartu

Cari saldo berdasarkan nomor rekening

Periksa apakah saldo > jumlah transfer

Kirim ke rekening tujuan jika saldo > jumlah transfer



SUBPROGRAM



Sekumpulan perintah yang didesain untuk melakukan operasi yang sering dilakukan dalam suatu program



Baca nomor kartu
Periksa PIN
Tampilkan pilihan menu jika PIN sesuai



CEK SALDO
Tampilkan saldo



CEK SALDO
Periksa apakah saldo > jumlah penarikan
Keluarkan uang jika saldo > jumlah penarikan



CEK SALDO
Periksa apakah saldo > jumlah pembayaran
Bayar ke nomor tujuan jika saldo > jumlah pembayaran



CEK SALDO
Periksa apakah saldo > jumlah transfer
Kirim ke rekening tujuan jika saldo > jumlah transfer



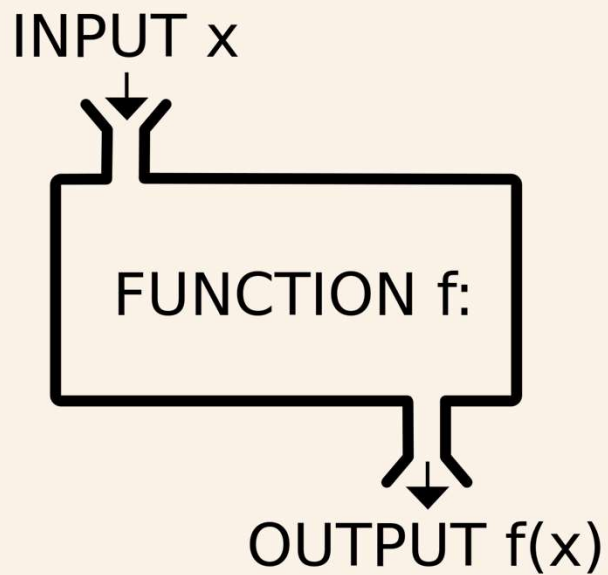
FUNGSI

Sekumpulan instruksi yang mungkin perlu input untuk melakukan pekerjaan tertentu dan mengembalikan satu nilai

PROSEDUR

Sekumpulan instruksi yang mungkin perlu input untuk melakukan pekerjaan tertentu

FUNGSI



FUNGSI

- Fungsi adalah rangkaian perintah yang diberi nama untuk melakukan komputasi.
- Fungsi menerima **argument** dan mengembalikan sebuah **hasil**.

BUILT-IN FUNCTION

`cin`
`cout`
`pow()`
`sqrt()`
`ceil()`
`floor()`

Fungsi bawaan yang sudah disediakan pada bahasa pemrograman tertentu.

Untuk menggunakan fungsi bawaan, harus mendeklarasikan file header dimana fungsi tersebut disimpan pada bagian pre-processing statement.



MEMBUAT FUNGSI

- Kita bisa membuat fungsi baru selain fungsi yang sudah ada pada built-in function.
- Saat program semakin kompleks, fungsi baru untuk melakukan pekerjaan tertentu sering dibutuhkan.
- Fungsi memecah suatu pekerjaan yang kompleks menjadi sub-tasks.



MEMBUAT FUNGSI

- 1 Deklarasikan fungsi
 - Tentukan nama fungsi
 - Tentukan formal parameter
 - Tentukan tipe data dari hasil
- 2 Buat realisasi fungsi
 - Buat algoritma untuk melakukan hal tertentu
- 3 Panggil fungsi di main function atau fungsi lainnya
 - Panggil fungsi menggunakan actual parameter

$$f(x) = x^2 + 3x - 5$$

Declaration of Pre-processor Statements

Global Declaration

Main Function

User Defined Function

```
#include <iostream>

using namespace std;
```

Declaration of Pre-processor Statements

```
int fxkuadrat(int x);
```

Global Declaration

```
int main()
{
    int x, p, hasil;

    x = 3;
    hasil = fxkuadrat(x);
    p = 10 + fxkuadrat(10);
    cout << hasil << " " << p << " " << fxkuadrat(10) << endl;

    return 0;
}
```

Main Function

```
int fxkuadrat(int x)
{
    return (x*x+3*x-5);
}
```

User Defined Function

```

#include <iostream>

using namespace std;

int fxkuadrat(int x); 1

int main()
{
    int x, p, hasil;

    x = 3;
    hasil = fxkuadrat(x);
    p = 10 + fxkuadrat(10); 3
    cout << hasil << " " << p << " " << fxkuadrat(10) << endl;

    return 0;
}

int fxkuadrat(int x) 2
{
    return (x*x+3*x-5);
}

```

1

Definisikan fungsi

- Tentukan nama fungsi
- Tentukan formal parameter
- Tentukan tipe data dari hasil

2

Buat realisasi fungsi

- Buat algoritma untuk melakukan hal tertentu

3

Panggil fungsi di main function atau fungsi lainnya

- Panggil fungsi menggunakan actual parameter



MENDEFINISIKAN FUNGSI

Fungsi harus didefinisikan terlebih dahulu sebelum dipanggil di tempat lain.

Sebuah fungsi bisa tidak membutuhkan argument atau bisa memiliki banyak argument.

Fungsi yang memiliki banyak argument dipisahkan dengan koma.

SINTAKS MENDEFINISIKAN FUNGSI

```
tipe_data_hasil nama_fungsi(tipe_data1  
parameter_formal1, tipe_data2 parameter_formal2, ... ,  
tipe_dataN parameter_formalN);
```

```
int fxkuadrat(int x);
```



MEREALISASIKAN FUNGSI

Realisasi fungsi berisi algoritma untuk melakukan pekerjaan tertentu.

Fungsi **harus** mengembalikan sebuah nilai. Jika tidak, subprogram tersebut dikenal sebagai prosedur (dibahas pertemuan berikutnya)

SINTAKS MEREALISASIKAN FUNGSI

```
 tipe_data_hasil nama_fungsi(tipe_data1 parameter_formal1, tipe_data2
parameter_formal2, ... , tipe_dataN parameter_formalN)
{
    // algoritma untuk melakukan sesuatu

    return (hasil/variabel_hasil);
}
```

```
int fxkuadrat(int x)
{
    return (x*x+3*x-5);
}
```

```
int fxkuadrat(int x)
{
    int hasil;

    hasil = x*x+3*x-5;

    return (hasil);
}
```




MEMANGGIL FUNGSI

Fungsi bisa dipanggil sebanyak apapun dibutuhkan.

Pemanggilan berbagai fungsi bisa dilakukan secara kombinasi, atau sebagai argument untuk fungsi lain

KAPAN HARUS MEMBUAT FUNGSI

REUSE

Perlu menggunakan bagian tertentu dalam program di beberapa tempat.

REUSE

Mengurangi waktu yang dibutuhkan untuk melakukan perubahan dalam program.

DECOMPOSITION

Memecah program yang kompleks menjadi bagian-bagian kecil yang dilakukan di fungsi terpisah.

SHARING

Memungkinkan untuk digunakan orang lain hanya dengan mengetahui nama fungsi, input, dan output.

CAKUPAN VARIABEL



CAKUPAN VARIABEL

- Cakupan variable adalah area dalam program dimana variable tersebut dikenali dan bisa digunakan.
- Jika sebuah variable di definisikan dalam sebuah fungsi, variable tersebut hanya dikenali dan bisa digunakan dalam fungsi tersebut (variable **local**)
- Jika sebuah variable didefinisikan diluar semua fungsi, variable tersebut bisa digunakan baik di dalam maupun di luar fungsi (variable **global**)

VARIABEL LOKAL

- Didefinisikan di dalam sebuah fungsi.
- Variabel yang didefinisikan pada fungsi yang berbeda tidak saling dikenali walaupun nama variabel sama.

```
#include <iostream>

using namespace std;

int variabel_lokal();

int main()
{
    cout << "nilai x di fungsi utama = " << x << endl;

    return 0;
}
```

```
int variabel_lokal()
{
    int hasil, x = 1;
    hasil = x*x+3*x-5;
    cout << "nilai x di dalam fungsi variabel lokal = " << x ;

    return (hasil);
}
```

File	Line	Message
=== Build: Debug in test (compiler: GNU GCC Compiler) ===		
C:\Users\mered...		In function 'int main()':
C:\Users\mered...	10	error: 'x' was not declared in this scope
=== Build failed: 1 error(s), 0 warning(s) (0 minute(s), 3 second(s)) ===		


```
#include <iostream>

using namespace std;

int variabel_lokal();

int main()
{
    int x = 2;

    cout << "nilai x di fungsi utama = " << x << endl;

    return 0;
}

int variabel_lokal()
{
    int hasil, x = 1;
    hasil = x*x+3*x-5;
    cout << "nilai x di dalam fungsi variabel lokal = " << x ;

    return (hasil);
}
```

nilai x di fungsi utama = 2

Process returned 0 (0x0) ex
Press any key to continue.

```

#include <iostream>

using namespace std;

int variabel_lokal();

int main()
{
    int x = 2, y;

    cout << "nilai x di fungsi utama = " << x << endl;
    y = variabel_lokal();

    return 0;
}

int variabel_lokal()
{
    int hasil, x = 1;
    hasil = x*x+3*x-5;
    cout << "nilai x di dalam fungsi variabel lokal = " << x ;

    return (hasil);
}

```

```

nilai x di fungsi utama = 2
nilai x di dalam fungsi variabel lokal = 1
Process returned 0 (0x0)   execution time :
Press any key to continue.

```

VARIABEL GLOBAL

- Didefinisikan di luar semua fungsi, di bagian global declaration pada struktur program C++.
- Variabel global bisa diakses dari fungsi manapun.
- Jika variabel lokal dan global memiliki nama yang sama, yang akan digunakan adalah variabel lokal.

```

#include <iostream>

using namespace std;

int x = 3;
int variabel_lokal();

int main()
{
    int y;

    cout << "nilai x di fungsi utama = " << x << endl;
    y = variabel_lokal();

    return 0;
}

int variabel_lokal()
{
    int hasil;
    float x = 1.5;
    hasil = x*x+3*x-5;
    cout << "nilai x di dalam fungsi variabel lokal = " << x ;

    return (hasil);
}

```

```

nilai x di fungsi utama = 3
nilai x di dalam fungsi variabel lokal = 1.5
Process returned 0 (0x0)   execution time : 0.
Press any key to continue.

```

```

#include <iostream>

using namespace std;

int x = 3;
int variabel_lokal();

int main()
{
    int y;
    x = x + 3;
    cout << "nilai x di fungsi utama = " << x << endl;
    y = variabel_lokal();

    return 0;
}

int variabel_lokal()
{
    int hasil;
    x = x + 1;
    hasil = x*x+3*x-5;
    cout << "nilai x di dalam fungsi variabel lokal = " << x ;

    return (hasil);
}

```

```

nilai x di fungsi utama = 6
nilai x di dalam fungsi variabel lokal = 7
Process returned 0 (0x0)   execution time :
Press any key to continue.

```

PARAMETER FORMAL DAN AKTUAL

PARAMETER FORMAL

Daftar parameter yang terdiri dari tipe data dan nama variabel yang digunakan saat mendefinisikan fungsi.

PARAMETER AKTUAL

Parameter dalam bentuk Variable/Ekspresi/Nilai yang digunakan saat pemanggilan fungsi.

```
#include <iostream>
```

```
using namespace std;
```

```
int fxkuadrat(int x);
```

PARAMETER FORMAL

```
int main()
```

```
{
```

```
    int x, p, y, hasil;
```

```
    x = 3;
```

```
    hasil = fxkuadrat(x);
```

PARAMETER AKTUAL

```
    p = 10 + fxkuadrat(10);
```

```
    cout << hasil << " " << p << " " << fxkuadrat(x+1) << endl;
```

```
    y = fxkuadrat(fxkuadrat(x)+3);
```

```
    return 0;
```

```
}
```

```
int fxkuadrat(int x)
```

PARAMETER FORMAL

```
{
```

```
    int hasil;
```

```
    hasil = x*x+3*x-5;
```

```
    return (hasil);
```

```
}
```


PARAMETER FORMAL DAN AKTUAL

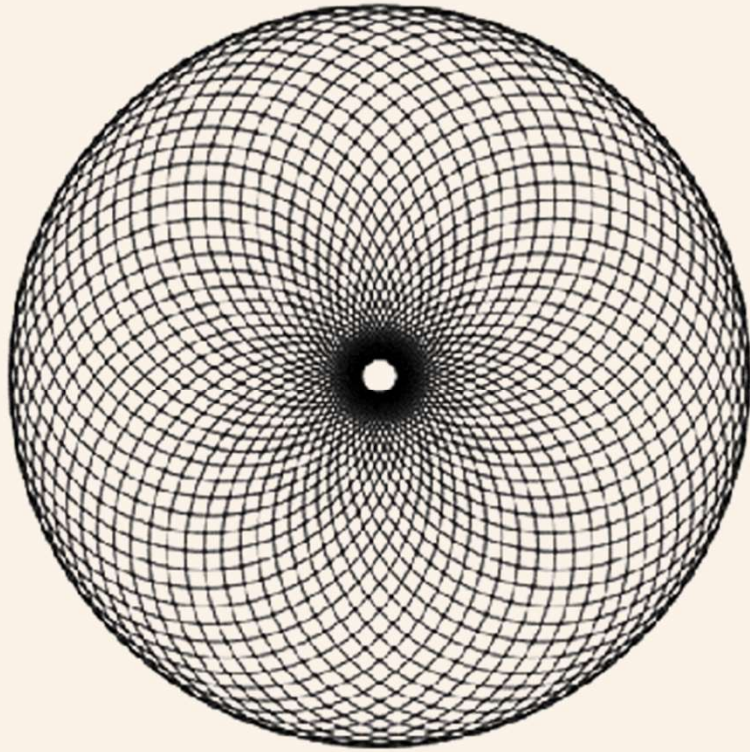
- Setiap parameter aktual berasosiasi dengan parameter formal sesuai urutannya.
- Tipe data parameter actual harus sesuai dengan tipe data formal parameter yang berasosiasi

```
int fxkuadrat(int x);
```

```
hasil = fxkuadrat(x);  
p = 10 + fxkuadrat(10);  
y = fxkuadrat(fxkuadrat(x)+3);
```

LATIHAN SOAL

LATIHAN 1



Buatlah sebuah fungsi yang digunakan untuk menerima sebuah bilangan riil (float) yang merupakan jari-jari sebuah lingkaran dan menghasilkan luas lingkaran berdasarkan jari-jari tersebut

LATIHAN 2

Bagaimana jika Anda harus membuat 2 buah fungsi dengan masukan jari-jari:

- Menghitung keliling lingkaran
- Menghitung luas lingkaran

Bagaimana agar konstanta π tidak perlu didefinisikan berulang-ulang di setiap fungsi?

LATIHAN 3

Tuliskan fungsi MAX2, yang menerima masukan dua buah bilangan integer dan menghasilkan bilangan terbesar

Contoh: MAX2(1,2) 2

Tuliskan fungsi MAX3 yang memanfaatkan fungsi MAX2. Fungsi MAX3 menerima input 3 bilangan integer dan menghasilkan bilangan terbesar

Contoh: MAX3(10,2,3) 10