

Fungsi dalam C

Tim PHKI Modul Dasar Pemrograman
Fakultas Ilmu Komputer
UDINUS Semarang



Dasar Pemrograman
Fakultas Ilmu Komputer - UDINUS

Deklarasi Fungsi Dalam C

- Ada 2 cara deklarasi fungsi dalam bahasa C.
 - Menuliskan fungsi “pendukung” diatas fungsi main
 - Deklarasi fungsi “pendukung” di awal program



Menuliskan diatas Fungsi Main

Berikut tata penulisan fungsi “pendukung” diatas fungsi main

```
#include<header_file_1>
#include<header_file_2>
....
#include<header_file_n>
//fungsi "pendukung_1" berada diatas fungsi main
<type_fungsi> <nama_fungsi_1>([parameter1, parameter2,...]){
    statement_yang_akan_dieksekusi;
    ...
}
//fungsi "pendukung_2" berada diatas fungsi main
<type_fungsi> <nama_fungsi_2>([parameter1, parameter2,...]){
    statement_yang_akan_dieksekusi;
    ...
}
....
//fungsi "pendukung_n" berada diatas fungsi main
<type_fungsi> <nama_fungsi_n>([parameter1, parameter2,...]){
    statement_yang_akan_dieksekusi;
    ...
}
//fungsi main berada dipaling bawah
<type_fungsi> <fungsi_main>([parameter1, parameter2,...]){
    statement yang akan dieksekusi;
    ...
}
```

Fungsi Pendukung



Fungsi main



Menuliskan diatas Fungsi Main

Contoh:

```
#include<stdio.h>
```

```
int KarakterToInteger(char x){  
    /*KAMUS KarakterToInteger*/  
    int x_int;  
    /* ALGORITMA KarakterToInteger*/  
    switch (x){  
        _____  
    return x_int;  
}
```

Fungsi Pendukung

Fungsi main

```
int main() { //fungsi main  
    /* KAMUS main*/  
    char x;  
    int x_int;  
    /* ALGORITMA main*/  
    printf ("\n\tmasukan karakter [0-9]=");  
    scanf ("%s", &x);  
    x_int=KarakterToInteger(x);  
    printf ("\tx berhasil dikonversi menjadi int dengan nilai = %d\n", x_int);  
    return 0;  
}
```



Deklarasi fungsi diawal

Berikut tata penulisan fungsi “pendukung” di awal program

```
#include<header_file_1>
....
#include<header file n>
//deklarasi fungsi diawal program
<type_fungsi> <nama_fungsi_1>([parameter1, parameter2,...]);
....
<type_fungsi> <nama_fungsi_n>([parameter1, parameter2,...]);

//fungsi main [dapat] berada dipaling atas atau dimana saja
<type_fungsi> <fungsi_main>([parameter1, parameter2,...]){
    statement_yang_akan_dieksekusi;
    ...
}

//fungsi "pendukung 1" diletakan dimana saja
<type_fungsi> <nama_fungsi_1>([parameter1, parameter2,...]){
    statement_yang_akan_dieksekusi;
    ...
}

....

//fungsi "pendukung_n" diletakan dimana saja
<type_fungsi> <nama_fungsi_n>([parameter1, parameter2,...]){
    statement_yang_akan_dieksekusi;
    ...
}
```

Deklarasi diawal program

Ingat : “Jika sudah di deklarasi Fungsi pendukung dapat diletakan dimana saja.”

Deklarasi fungsi diawal

Contoh:

```
#include<stdio.h>
#include<string.h>
/* KAMUS global*/
//deklarasi variable global
int saldo=1000000;
//deklarasi fungsi
void kurangiSaldo(int jmltarik);
void sisaSaldo();

int main() {

void kurangiSaldo(int jmltarik){
    saldo=saldo-jmltarik;
    printf ("Anda melakukan penarikan sebesar %d\n", jmltarik);
    sisaSaldo();
}

void sisaSaldo(){
    printf ("Sisa saldo Anda = %d\n", saldo);
    printf ("Terima kasih atas kunjungan Anda\n");
}
```

Deklarasi Fungsi
Pendukung diawal
program

Fungsi main



Lingkup Variable

- Variable Global
 - Adalah variable yang didklarasikan diluar fungsi, baik fungsi utama, maupun fungsi pendukung.
- Variable Lokal
 - Adalah variable yang dideklarasasi dalam fungsi, jadi hanya dapat diakses oleh fungsi itu sendiri.



Contoh Lingkup Variable

```
vGlobalLokal.c |
7 Spesifikasi      : menampilkan variable global dari
8                   |   fungsi utama dan fungsi pendukung
9                   |   -----*/
10 #include<stdio.h>
11 /* KAMUS (Global)*/
12 int x=10; //variable Global
13 //fungsi pendukung --> ini adalah komentar
14 void TampilNilaiX () {
15     /* KAMUS (lokal-fungsi TampilNilaiX)*/
16     /* ALGORITMA (lokal-fungsi TampilNilaiX)*/
17     printf ("X dipanggil dari fungsi TampilNilaiX :%d\n", x);
18 }
19 //fungsi utama --> ini adalah komentar
20 int main() {
21     /* KAMUS (lokal-fungsi utama)*/
22     int y=10;//variable Lokal
23     /* ALGORITMA (lokal-fungsi utama)*/
24     TampilNilaiX (); // memanggil fungsi TampilNilaiX ()
25     printf ("X dipanggil dari fungsi utama :%d\n", x);
26     printf ("y merupakan variable lokal fungsi main :%d\n", y);
27     return 0;
28 }
```


Jenis Parameter dalam C

- Dalam dunia pemrograman (terutama bahasa C) terdapat 3 macam parameter, yaitu:
 - Parameter masukan
 - Parameter keluaran
 - Parameter masukan/keluaran



Parameter Masukan

- Digunakan untuk menampung data sebagai input ke fungsi
- Dengan ini, fungsi dapat menghasilkan output yang berbeda



Contoh Parameter Masukan

```
#include<stdio.h>
/*KAMUS Global*/
static double PI = 3.14;
//definisi fungsi hitungKeliling dengan parameter masukan
double hitungKeliling(int radius){
    /*KAMUS hitungKeliling*/
    double K;
    /* ALGORITMA hitungKeliling*/
    K = (2 * PI * radius);
    return K;          //Fungsi Utama
}

int main(void) {
    /* KAMUS main*/
    int r;
    double K;
    /* ALGORITMA main*/
    printf ("\n\tKALKULATOR HITUNG KELILING LINGKARAN\n");
    printf ("\tMasukan jari2 lingkaran = ");
    scanf("%d", &r);
    K=hitungKeliling(r);
    printf ("\tKeliling lingkaran = %.3f\n", K);
    return 0;
}
```



Parameter Keluaran

- Kebalikan parameter masukan
- Digunakan untuk menampung nilai yang akan dikembalikan ke proses (output)
- Banyak digunakan untuk fungsi bertipe void



Contoh Parameter Keluaran

```
#include<stdio.h>
/*KAMUS Global*/
static double PI = 3.14;
//definisi fungsi hitungKeliling dengan parameter keluaran
void hitungKeliling(int radius, double *K) {
    /*KAMUS hitungKeliling*/
    /* ALGORITMA hitungKeliling*/
    *K = (2 * PI * radius);
}
//Fungsi Utama
int main(void) {
    /* KAMUS main*/
    int r;
    double Keliling;
    /* ALGORITMA main*/
    printf ("\n\tKALKULATOR HITUNG KELILING LINGKARAN\n");
    printf ("\tMasukan jari2 lingkaran = ");
    scanf("%d", &r);
    hitungKeliling(r, &Keliling);
    printf ("\tKeliling lingkaran = %.3f\n", Keliling);
    getch();
    return 0;
}
```



Parameter Masukan/Keluaran

- Parameter ini punya 2 kegunaan sekaligus:
 - Bertindak sebagai penampung nilai masukan
 - Selanjutnya sebagai penampung nilai keluaran



Parameter Masukan/Keluaran

```
#include<stdio.h>
/*KAMUS Global*/
static double PI = 3.14;
//definisi fungsi hitungKeliling dengan parameter in/out
void hitungKeliling(double *x) {
    /*KAMUS hitungKeliling*/
    /* ALGORITMA hitungKeliling*/
    *x = 2 * PI * (*x);
}
//Fungsi Utama
int main(void) {
    /* KAMUS main*/
    int r;
    double param;
    /* ALGORITMA main*/
    printf ("\n\tKALKULATOR HITUNG KELILING LINGKARAN\n");
    printf ("\tMasukan jari2 lingkaran = ");
    scanf("%d", &r);
    param = (double) r;//typecasting int to double
    hitungKeliling(&param);
    printf ("\tKeliling lingkaran = %.3f\n", param);
    getch();
    return 0;
}
```



THANKS

- References : I Made Joni, Budi Raharjo,
Pemrograman C dan Implementasinya (2011)

