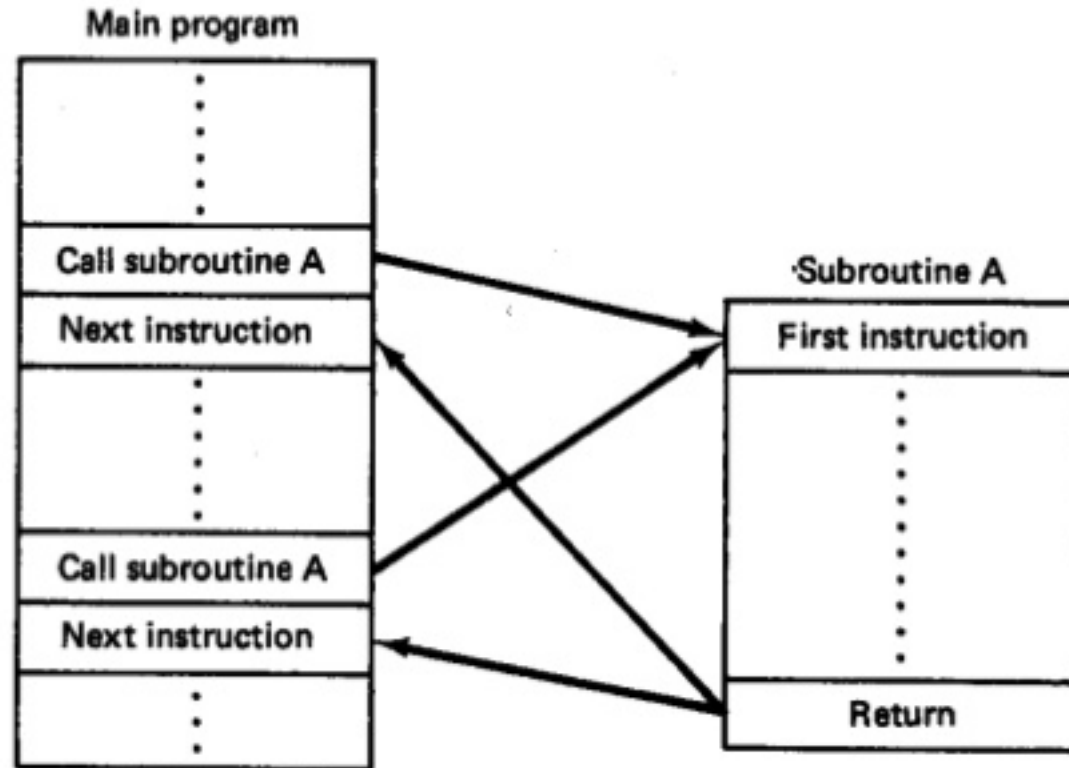


Pemrograman Dasar

Pertemuan VIII

Subrutin

- Sebuah program yang besar biasanya disusun atas sejumlah bagian yang lebih kecil → subrutin / subprogram
- Tujuan penggunaan subrutin :
 - Untuk memudahkan pengelolaan atau pengembangan program
 - Untuk mengurangi jumlah kode → pada program yang besar, seringkali potongan kode yang sama diketik berulang-ulang / digunakan berkali-kali.



Bentuk Subrutin

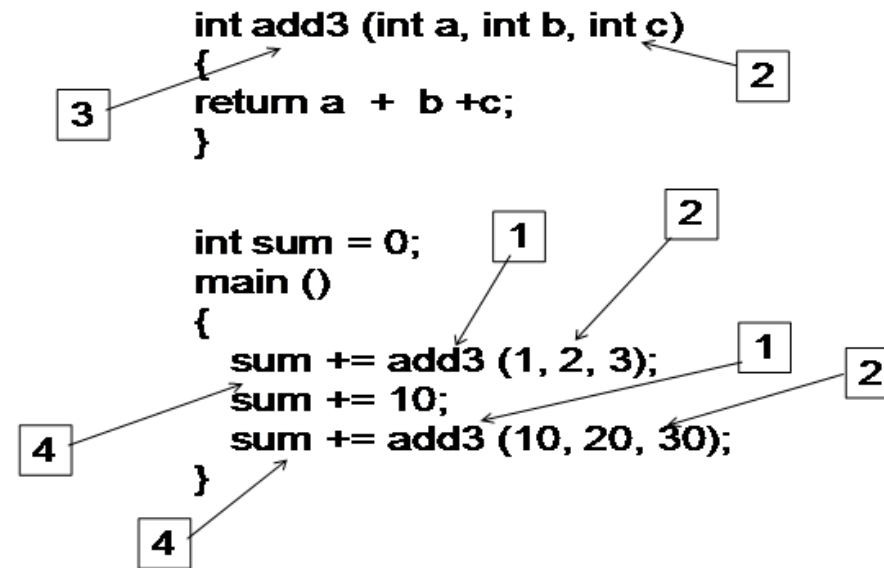
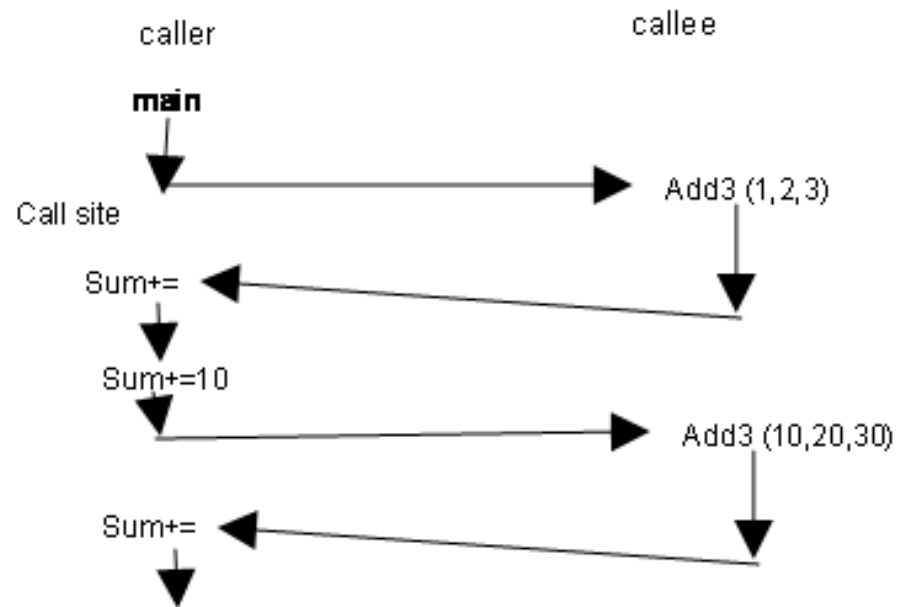
```
SUBROUTIN namaSubrutin (daftarParameter)  
    Pernyataan1  
    Pernyataan2  
    ...  
    PernyataanN  
AKHIR-SUBROUTIN
```

- SUBROUTIN *namaSubrutin* : disebut sebagai judul subrutin
- Subrutin dapat memberikan nilai balik maupun tidak.
- Nilai balik : nilai yang diberikan ke pemanggilnya
- Contoh :

```
    SUBROUTIN perolehKelilingKotak (panjang, lebar)  
        keliling  $\leftarrow 2 \times (\textit{panjang} + \textit{lebar})$   
        NILAI-BALIK keliling  
    AKHIR-SUBROUTIN
```

- Contoh pemanggilannya :
 hasil \leftarrow *perolehKelilingKotak* (2,3)
 Maka nanti *hasil* akan bernilai 10

Contoh Lain Subrutin



Contoh Program I

- Menghitung keliling kotak
- SUBROUTIN *perolehKelilingKotak (panjang, lebar)*
keliling $\leftarrow 2 \times (\text{panjang} + \text{lebar})$
NILAI-BALIK *keliling*
AKHIR-SUBROUTIN

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

```
double hitung_keliling_kotak (double panjang, double lebar)
{
    double keliling;
    keliling = 2 * (panjang + lebar);
    return keliling;
}
```

```
int main ()
{
    double keliling, panjang, lebar;
    panjang = 56.5;
    lebar = 2;
    keliling = hitung_keliling_kotak (panjang, lebar);
    printf("keliling = %lf\n", keliling);
    return 0
} //kelkotak.c
```

Contoh Program II

- Berikut adalah contoh **fungsi tanpa nilai balik**

- `#include <stdio.h>`

```
void infoPerusahaan()  
{  
    printf("Teknik Informatika UPN Veteran Jatim");  
}
```

```
int main ()  
{  
    infoPerusahaan();  
    return 0;  
}
```

Contoh Program III

- Contoh aplikasi mencari bilangan terkecil

- Algoritma :
SUBROUTIN terkecil(x,y)
 JIKA $x < y$ MAKA
 $\text{min} \leftarrow x$
 SEBALIKNYA
 $\text{min} \leftarrow y$
 AKHIR-JIKA

 NILAI-BALIK min
AKHIR-SUBROUTIN

```
#include <stdio.h>

double terkecil (double x, double y)
{
    double min;
    if (x < y)  min = x;
    else  min = y
    return min;
}

int main ()
{
    double a, b, c;
    a = 35.6;
    b = 78.5;
    c = terkecil (a,b);
    printf("Terkecil antara %lf dan %lf : %lf \n", a, b, c);
    return 0;
}
```

Contoh Program IV

- Contoh aplikasi mencari bilangan terkecil dari sebuah array

• Algoritma :
SUBROUTIN terkecil(X,jum)
 min \leftarrow X[0]
 UNTUK i=1 S/D jum – 1
 JIKA X[i] < min MAKA
 min \leftarrow X[i]
 AKHIR-JIKA
 AKHIR-UNTUK

 NILAI-BALIK min
 AKHIR-SUBROUTIN

```
#include <stdio.h>

double terkecil (double x[], int jum)
{
    int i;
    double min;

    min = x[0];
    for (i=1; i < jum; i++)
        if(x[i] < min)
            min = x[i];
    return min;
}

int main ()
{
    double y;
    double data [] = {578, 67.8, 2.3, 24, 123.5};
    y = terkecil(data,5);
    printf("Terkecil : %lf\n", y);
    return 0;
} //minim.c
```


Contoh Program V

- Contoh subrutin untu membedakan bilangan genap atau ganjil
- Algoritma :
SUBRUTIN ganjil (bil)
 JIKA sisaPembagian(bil,2) = 1 MAKA
 NILAI-BALIK BENAR
 SEBALIKNYA
 NILAI-BALIK min
 AKHIR-JIKA
AKHIR-SUBRUTIN

```
#include <stdio.h>

int ganjil (int bil)
{
    if(bil % 2 == 1) return 1;
    else return 0;
}

int main ()
{
    printf("%d\n", ganjil (5));
    printf("%d\n", ganjil (7));
    printf("%d\n", ganjil (6));
} //ganjil.c
```

Contoh Program VI

- Contoh subrutin mengulang string

- Algoritma :

```
SUBROUTIN ulang (St, n)
    panjang ← jumlahKarakter(St)
    JIKA panjang = 0 MAKA
        NILAI-BALIK ""
    AKHIR-JIKA

    posisi ← 0
    StrTemp ← ""
    UNTUK i ← 0 S/D panjang – 1
        UNTUK j ← 0 S/D panjang – 1
            StrTemp[posisi] ← St[j]
            posisi ← posisi + 1
        AKHIR-UNTUK
    AKHIR-UNTUK

    StrTemp[posisi] ← 0

    NILAI-BALIK StrTemp
    AKHIR-SUBROUTIN
```

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

```
char * ulang(char st[], int n)
{
    static char strtemp[1024];
    int i, j, panjang, posisi;
    panjang = strlen(st);
    if(panjang == 0)
        return (char *) 0;
    posisi=0;
    for(i=0;i<n;i++)
        for(j=0; j<panjang; j++)
        {
            strtemp[posisi] = st[j];
            posisi++;
        }
    strtemp[posisi]=0;
    return (char *) strtemp;
}

int main ( )
{
    printf("%s\n", ulang("+", 8));
    printf("%s\n", ulang("HA", 4));
    printf("%s\n", ulang("BLA", 3));
} // ulang.c
```

Contoh Program VII

- Contoh subrutin untuk menghitung sisi miring segitiga siku-siku

• Algoritma :
SUBROUTIN *sisiMiring* (x,y)
 NILAI-BALIK akar(kuadrat(x) +
 kuadrat(y))
AKHIR-SUBROUTIN

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

```
double sisi_miring (double x, double y)
{
    return sqrt (x * x + y * y)
}
```

```
int main ()
{
    printf("%lf\n", sisi_miring(3,4));
    return 0;
} //sisimir.c
```

Call by Reference

- Pada bahasa C, pelewatan argumen ke dalam fungsi membuat nilai argumen tersebut dapat diubah dalam fungsi, atau dikenal dengan *call by reference*. Terdapat dua hal yang perlu diperhatikan :
 1. Parameter dalam definisi fungsi perlu ditulis dengan awalan tanda *. Hal serupa dikenakan pada semua akses parameter yang disebutkan dalam tubuh fungsi
 2. Awalan & perlu ditulis di depan argumen pada pemanggilan fungsi.
 3. Sebagai pengecualian, tanda * tidak perlu disebutkan jika argumennya berupa array

Contoh Program VIII

- Contoh subrutin untuk menghitung menukar nilai x dan nilai y

- Algoritma :
SUBROUTIN *tukar* (x,y)
 $z \leftarrow x$
 $x \leftarrow y$
 $y \leftarrow z$
AKHIR-SUBROUTIN

```
#include <stdio.h>

void tukar (double *x, double *y)
{
    double z;
    z = *x;
    *x = *y;
    *y = *z;
}

int main ()
{
    double a,b;
    a = 56.7;
    b = 12.3;
    tukar(&a, &b);
    printf("a = %lf, b = %lf \n", a, b)
} //tukar.c
```

Latihan Soal

1. Buatlah program dengan subrutin yang mengandung dua buah argumen array L (array integer) dan h yang menyatakan jumlah elemen dalam array L. Subrutin tersebut nantinya akan mengembalikan rata-rata dari array L.
2. Buatlah program dengan subrutin dimana argumen input berupa string dan outputnya adalah string kebalikan dari inputan. Misal input “Yogya”, maka outputnya adalah “aygoY”.