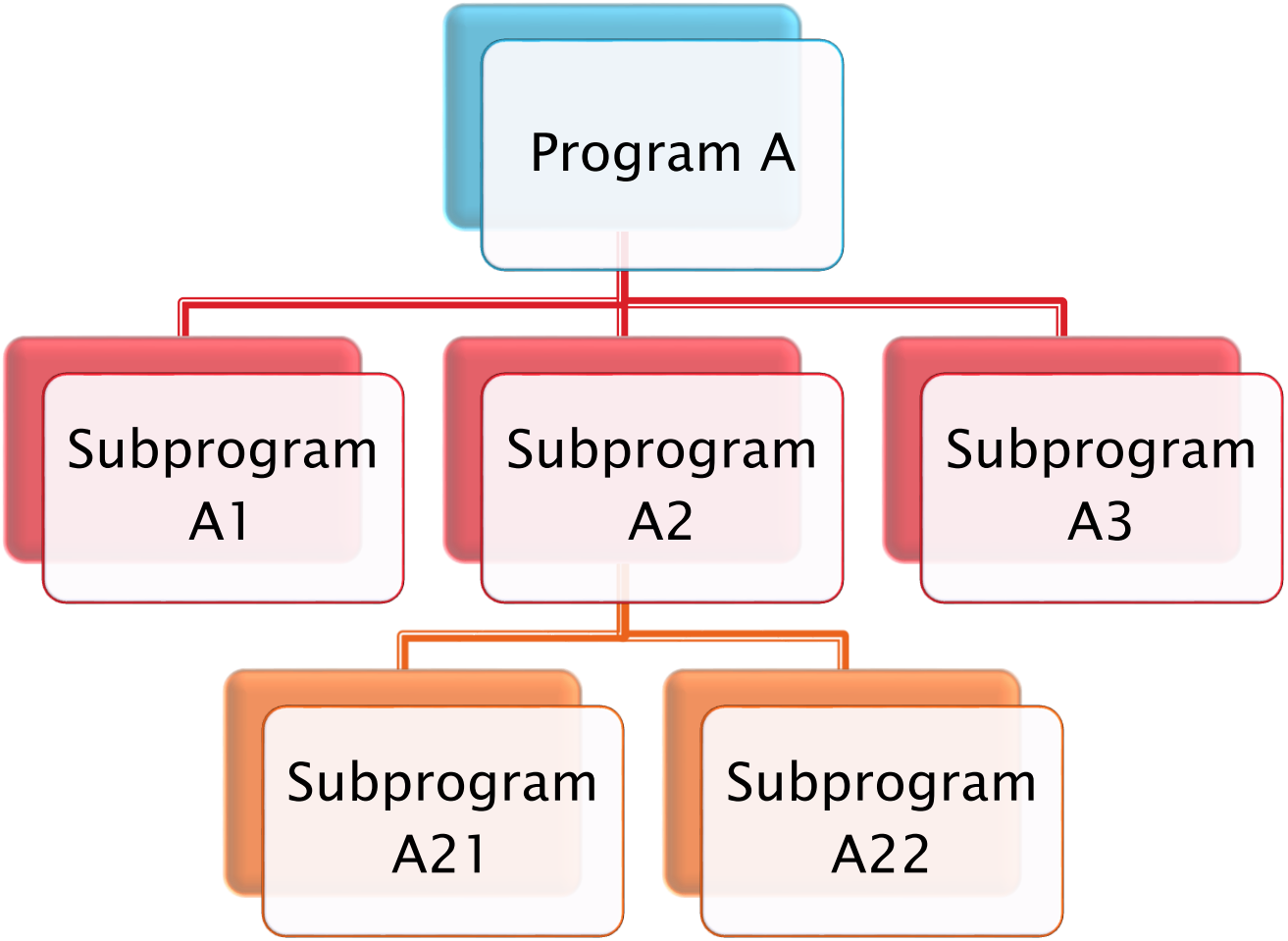
Pengantar Pemrograman Modular

Pemrograman Modular

Pemrograman modular merupakan teknik pemecahan masalah menjadi sejumlah subprogram.

- Subprogram sering juga disebut sebagai *subroutine*, modul, prosedur, atau fungsi

Ilustrasi


Program A
Subprogram A1
Subprogram A2
Subprogram A3
Subprogram A21
Subprogram A22

Contoh 1

Program Pertukaran

```
/* PROGRAM Pertukaran */
/* Mempertukarkan nilai A dan B. Nilai A dan B dibaca terlebih dulu.*/

#include <stdio.h>
main()
{   /*DEKLARASI*/
     int A, B, temp;

     /*ALGORITMA:*/
     printf("A = ?"); scanf("%d", &A);
     printf("B = ?"); scanf("%d", &B);

     /*Proses Pertukaran*/
     temp = A;
     A = B;
     B = temp;

     /*Tulis nilai A dan B setelah pertukaran*/
     printf ("A = %d \n", A);
     printf ("B = %d \n", B);
}
```

Kandidat subprogram pertama

Kandidat subprogram kedua

Kandidat subprogram ketiga

SubProgram Program Pertukaran

```
/* Subprogram pertama */
```

```
void Baca (int *A, int *B)
/* Membaca nilai A dan B */
{
    /*ALGORITMA:*/
    /*Baca nilai A dan B */
    printf("A = ?"); scanf("%d", &A);
    printf("B = ?"); scanf("%d", &B);
}
```

SubProgram Program Pertukaran

```
/* Subprogram kedua */
```

```
void Tukar(int *A, int *B)
/*Mempertukarkan nilai A dan B*/
{
    /*DEKLARASI*/
    int temp;   {peubah bantu}
    /*ALGORITMA:*/
    temp = *A;
    *A = *B;
    *B = temp;
}
```

```
/* Subprogram ketiga */
```

```
void Tulis(int A, int B)
/* Mencetak nilai A dan B */
{
    /*ALGORITMA:*/
    printf ("A = %d \n", A);
    printf ("B = %d \n", B);
}
```

Program Utama Pertukaran

```
/* PROGRAM Pertukaran */
/* Mempertukarkan nilai A dan B. Nilai
A dan B dibaca terlebih dahulu. */

#include <stdio.h>
void Baca(int *A, int *B);
void Tukar(int *A, int *B);
void Tulis(int *A, int *B);

main() /* Program Utama */
{
/* DEKLARASI */
  int A, B;
/* ALGORITMA */
  Baca(A,B); /*Baca nilai A & B*/
  Tukar(&A, &B); /*Pertukaran */
  Tulis(A, B); /*Tulis nilai A&B*/
}
```

```
void Baca (int *A, int *B)
/* Membaca nilai A dan B */
{
     /*ALGORITMA:*/
     /*Baca nilai A dan B */
     printf("A = ?"); scanf("%d", &A);
     printf("B = ?"); scanf("%d", &B);
}

void Tukar(int *A, int *B)
/*Mempertukarkan nilai A dan B*/
{
     /*DEKLARASI*/
     int temp;    {peubah bantu}
     /*ALGORITMA:*/
     temp = *A;
     *A = *B;
     *B = temp;
}

void Tulis(int A, int B)
/* Mencetak nilai A dan B */
{
     /*ALGORITMA:*/
     printf ("A = %d \n", A);
     printf ("B = %d \n", B);
}
```

Contoh 2

```
/* PROGRAM xyz */
#include <stdio.h>

main() /* Program Utama */
{
/* DEKLARASI */
   int A, B, C, D, temp;

/* ALGORITMA */
   …
/* Pertukarkan nilai A dan B */
   temp = A;
   A = B;
   B = temp;
   …

   If (C > D)
{
/* Pertukarkan nilai C dan D */
     temp = C;
     C = D;
     D = temp;
}
…
}
```

Memiliki aktivitas yang sama :

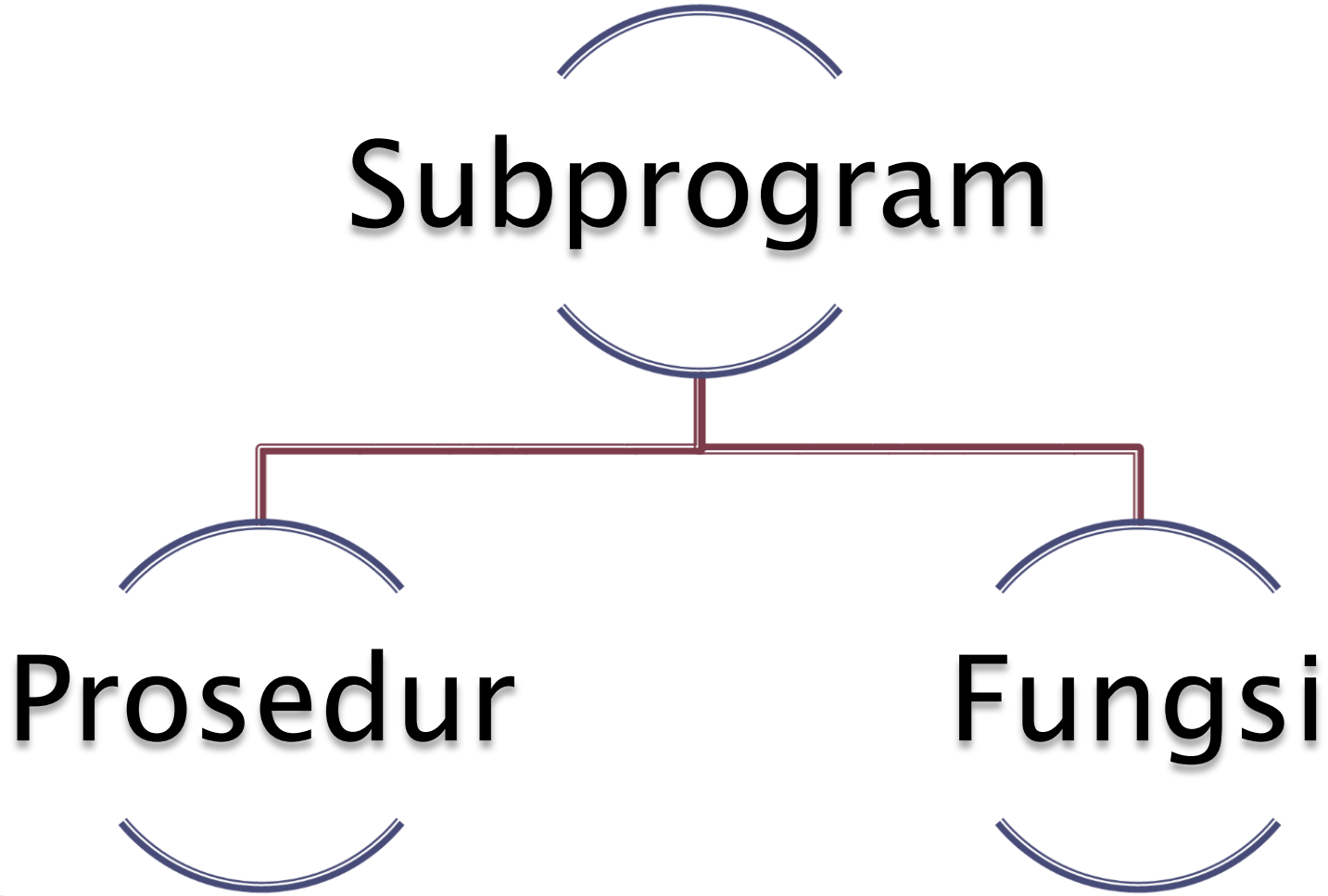
Pertukaran dua buah peubah

```
/* PROGRAM xyz */
/* Mempertukarkan nilai A dan B. Nilai A dan B dibaca terlebih dahulu. */

#include <stdio.h>

main() /* Program Utama */
{
/* DEKLARASI */
   int A, B, C, D, temp;

/* ALGORITMA */
   …
/* Pertukarkan nilai A dan B */
   temp = A;
   A = B;
   B = temp;
   …
}
   If (C > D)
{
/* Pertukarkan nilai C dan D */
     temp = C;
     C = D;
     D = temp;
}
…
}
```

```
/* Pertukarkan nilai A dan B */
   Tukar(&A, &B);
   …
}
   If (C > D)
{
/* Pertukarkan nilai C dan D */
    Tukar(&C, &D);
}
   …
   …
}

void Tukar(int *A, int *B)
/*Mempertukarkan nilai A dan B*/
{
    /*DEKLARASI*/
    int temp;  {peubah bantu}
    /*ALGORITMA:*/
    temp = *A;
    *A = *B;
    *B = temp;
}
```

Keuntungan Pemrograman Modular

- √ Menghindari penulisan teks program yang sama berulang kali.
- √ Kemudahan menulis dan menemukan kesalahan (debug) program.


Subprogram
Prosedur
Fungsi