



# PEMROGRAMAN MODULAR (PROSEDUR DAN FUNGSI)

Algoritma dan Struktur Data

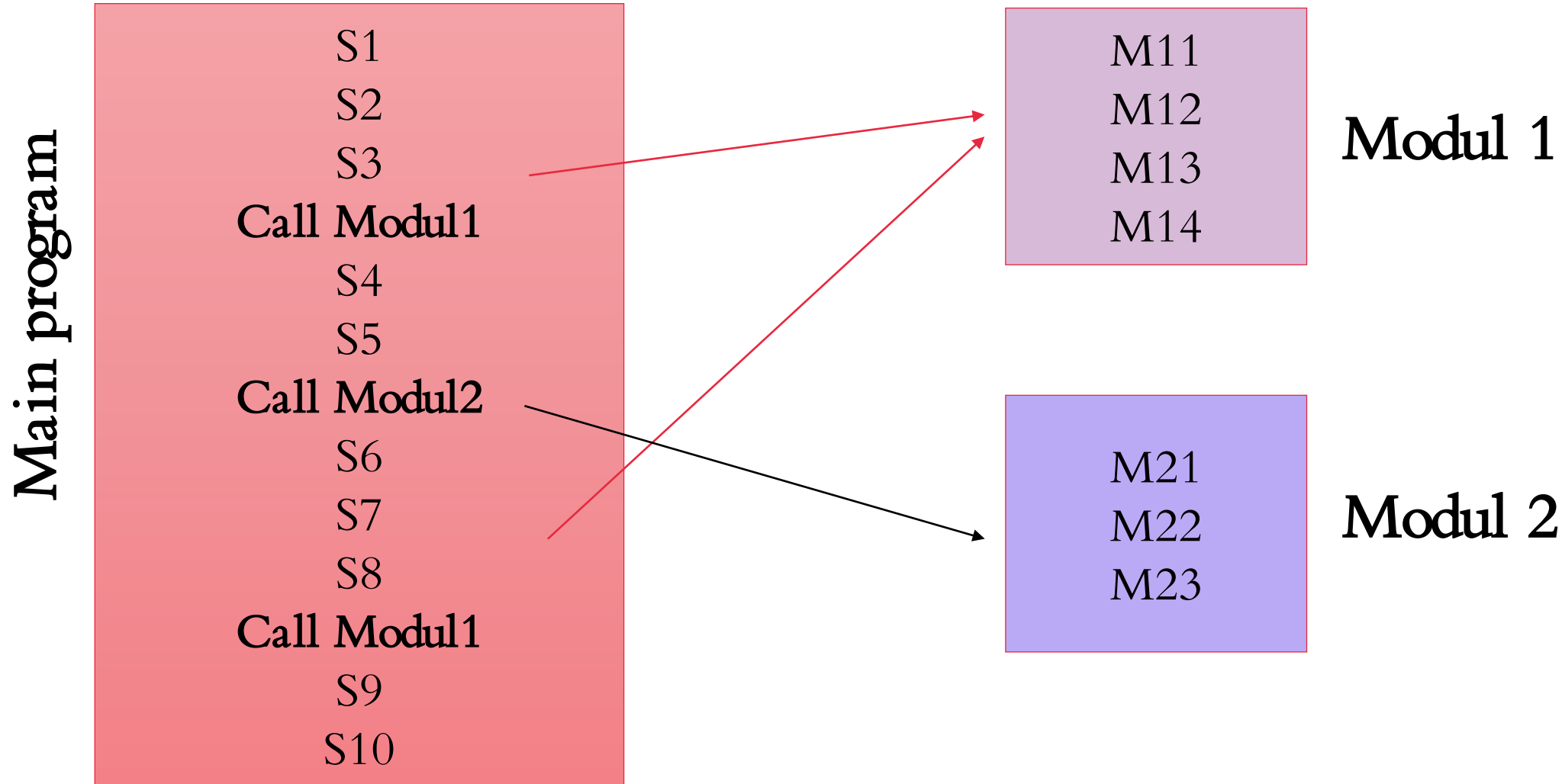
## PEMROGRAMAN MODULAR

- Salah satu kemudahan yang diberikan dalam memprogram untuk menghindari penulisan sintaks yang sama secara berulang-ulang.
- “Untuk mencapai suatu tujuan besar, maka tujuan tersebut harus dibagi-bagi menjadi tujuan kecil sampai tujuan kecil itu merupakan tujuan yang dapat dicapai berdasarkan kondisi dan potensi yang dimiliki saat itu “
- (Al-Khuwarizmi)

## TUJUAN DARI PROSES MODULARISASI

- Modularisasi menghindari penulisan teks program (syntax) yang sama secara berulang kali (efisiensi algoritma).
- Modularisasi memudahkan memprogram dan menemukan kesalahan (debug) program.
- Pemrograman modular akan memudahkan dari sisi pemahaman dan pembacaan syntx program

# RUNTUNAN PROSES DALAM EKSEKUSI MODULAR



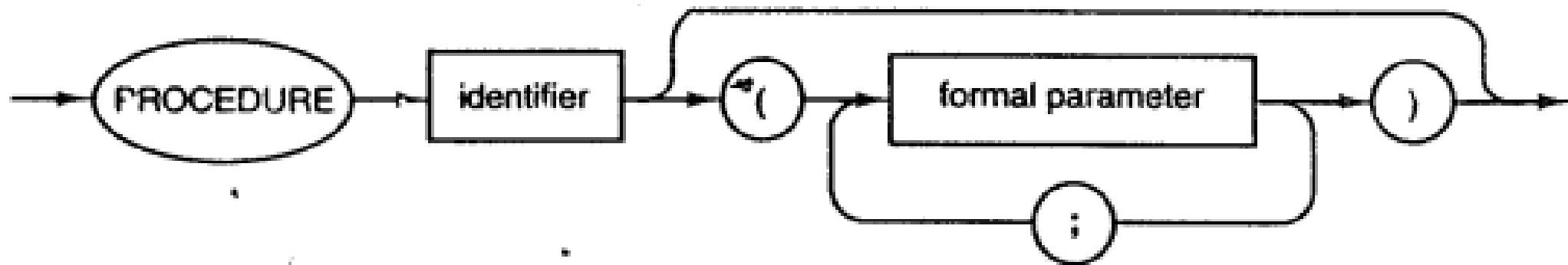
# JENIS PEMROGRAMAN MODULAR

- PROCEDURE (Prosedur)
- FUNCTION (Fungsi)

# PROSEDUR

- Tipe Subrutin yang digunakan dalam algoritma yang bertujuan untuk menerapkan konsep **pemrograman modular** dan **mempersingkan pemrograman**

procedure heading



## PROSEDUR DALAM BAHASA C

- Pendefinisian prosedur ditulis di luar blok program utama (main)
- Umumnya diletakkan setelah blok main()
- Sebenarnya, dalam Bahasa, tidak dikenal istilah prosedur, Semua modul program adalah fungsi. Prosedur adalah fungsi yang tidak mengembalikan nilai apapun. Karena itu, return valuenya adalah “void”
- Dalam Bahasa C bila prosedur tidak mempunyai parameter, maka tanda kurung tetap ditulis setelah nama prosedur tersebut.

VOID	NON VOID
fungsi yang <i>tidak mengembalikan nilai</i>	fungsi yang <i>mengembalikan nilai</i>
sering disebut = <i>prosedur</i>	sering disebut = <i>function</i>
tidak ada keyword <i>return</i>	ada keyword <i>return</i>
tidak ada <i>tipe data</i> dalam deklarasi fungsi	ada <i>tipe data</i> dalam deklarasi fungsi
menggunakan keyword <i>void</i>	tidak ada keyword <i>void</i>
secara langsung tidak dapat ditampilkan hasilnya	secara langsung dapat ditampilkan hasilnya



# CONTOH SEDERHANA PROSEDUR

- Prosedur Urus Paspor (dikantor imigrasi)

1. Isi formulir permintaan paspor dengan lampiran dokumen identitas
2. Serahkan formulir yang sdh diisi beserta biaya pembuatan paspor
3. Wawancara dengan petugas imigrasi
4. Terima paspor

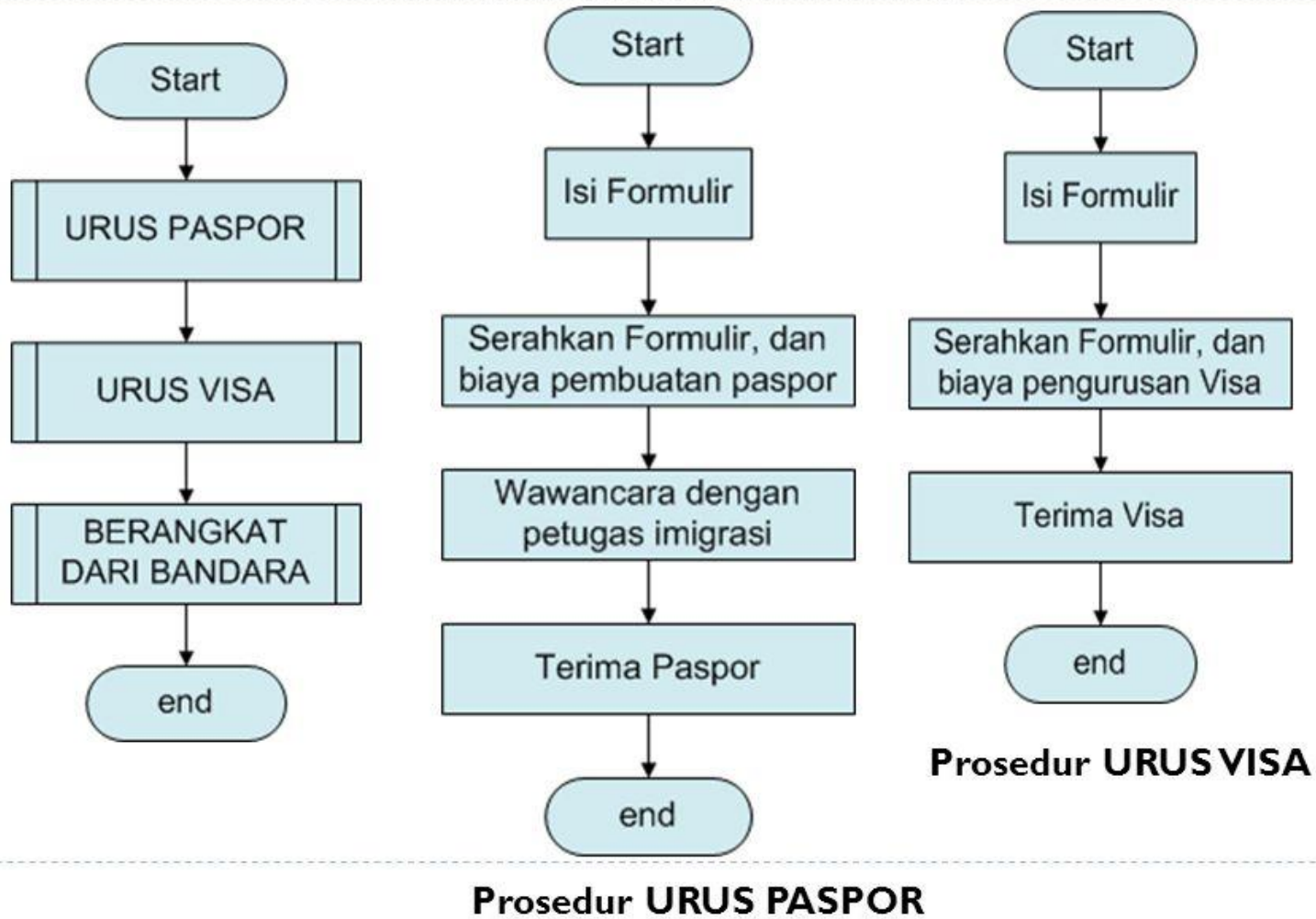
- Prosedur Urus Visa (di kantor kedutaan besar)

1. Isi formulir permohonan cvisa dengan lampiran dokumen identitas diri termasuk passport
2. Serahkan formulir yang diisi beserta biaya pengurusan visa
3. Terima visa

- Prosedur Berangkat dari Bandara

1. Datang ke bandara satu jam sebelum keberangkatan
2. Jika sudah saatnya jadwal untuk naik ke pesawat tunjukkan tiket, paspor dan visa

# Flowchart Prosedur



# NOTASI ALGORITMIK UNTUK PROSEDUR

C

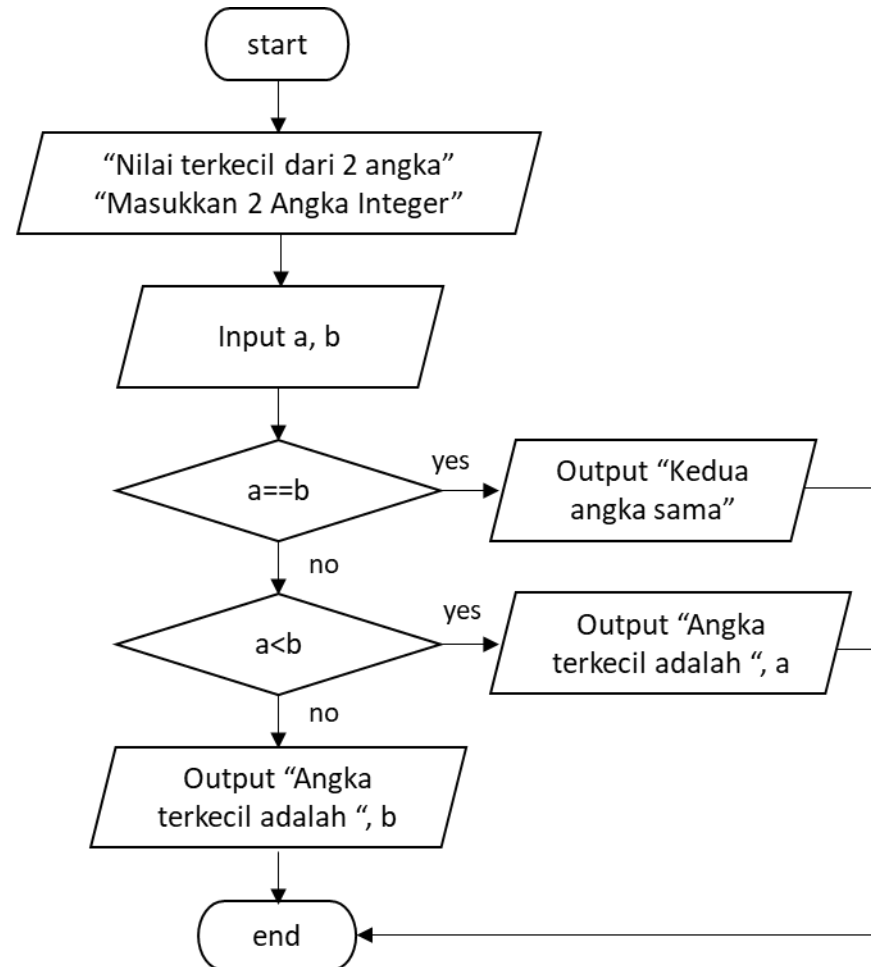
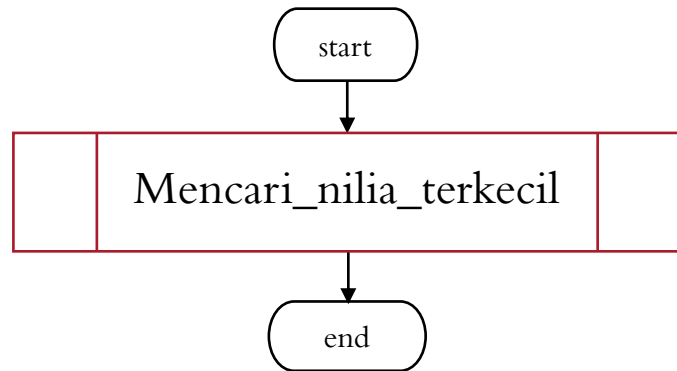
```
void NAMA_PROSEDUR()  
/* Spesifikasi prosedur, berisi penjelasan tentang apa yang dilakukan  
   oleh prosedur ini. */  
/* K.Awal : keadaan sebelum prosedur dilaksanakan. */  
/* K.Akhir: keadaan setelah prosedur dilaksanakan. */  
  
{  
  /* DEKLARASI */  
  /* semua nama yang dipakai dalam prosedur dan hanya berlaku lokal di  
     dalam prosedur didefinisikan di sini */  
  
  /* DESKRIPSI: */  
  
  /* badan prosedur, berisi kumpulan instruksi */  
}
```

Cara memanggil prosedur:

```
NAMA_PROSEDUR();
```

- Sumber: Rinaldi Munir (2000)

# CONTOH FLOWCHART PROSEDUR



## IDENTIFIER GLOBAL & LOKAL

- Dengan adanya subrutin maka identifier akan dibagi menjadi dua tipe yaitu:
  - Identifier global: identifier yang dikenal oleh semua bagian program (semua prosedur dan program utama)
  - Identifier local adalah identifier yang dikenal oleh satu atau sebagai program

1. PROGRAM p;  
VAR x : real;

PROCEDURE pl;  
VAR y : integer;  
begin  
.....;  
.....;  
end;

begin  
.....;  
.....;  
end.

daerah berlakunya y

daerah  
berlakunya x

2. Program P ;  
 Var x,y : real; \_\_\_\_\_ y (real) berlaku disemua blok  
 ..... program P, kecuali di P1  
 ..... x (real) berlaku disemua  
 ..... blok P, kecuali di P2

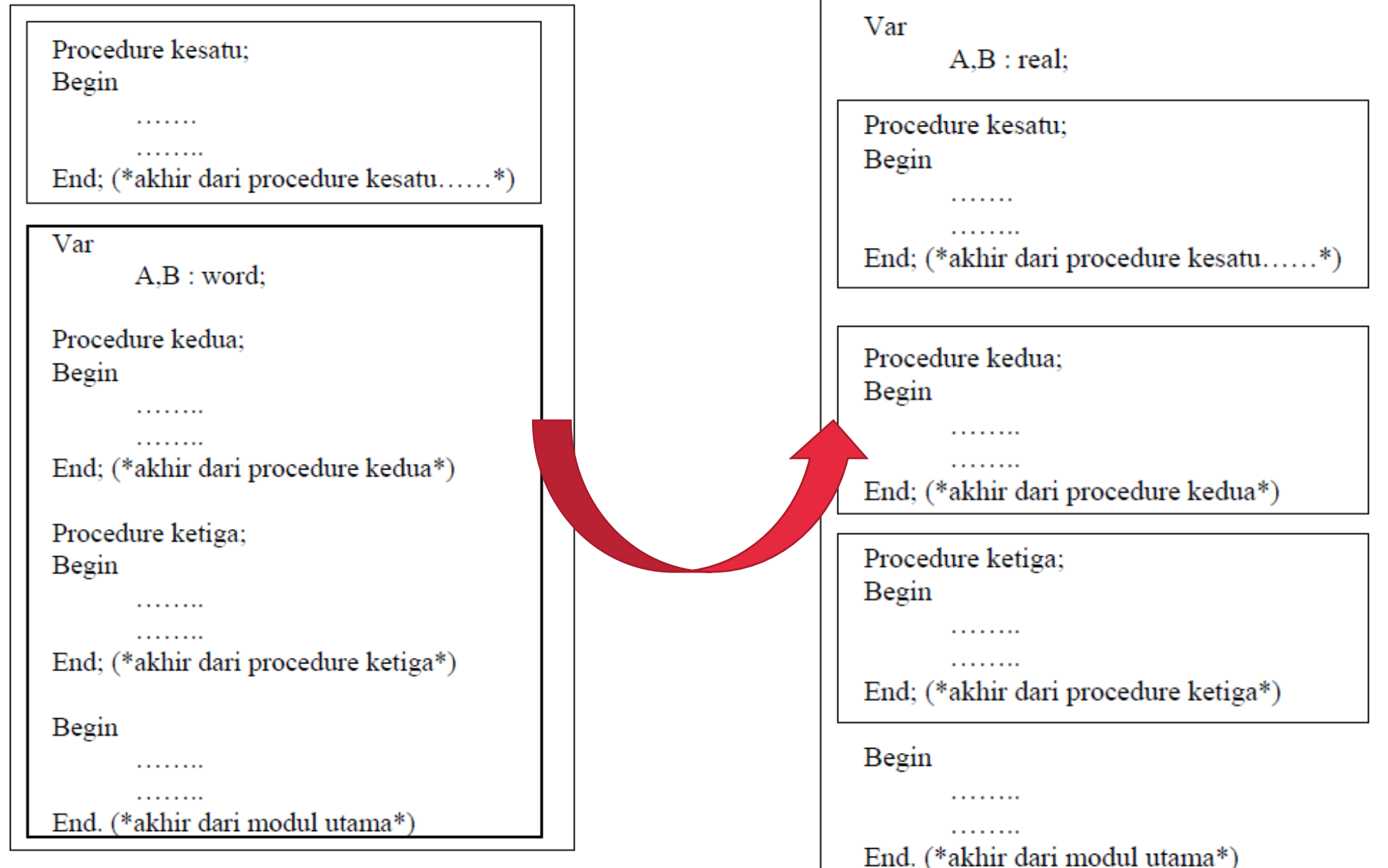
Procedure P1 ;  
 Var y : integer; \_\_\_\_\_ hanya berlaku di P1 saja (lokal)  
 .....  
 Begin  
 .....  
 .....  
 End;

Procedure P2;  
 Var x : char; \_\_\_\_\_ hanya berlaku di P2 saja (lokal)  
 .....  
 Begin  
 .....  
 .....  
 End;

Begin  
 .....  
 x := .....  
 y := .....  
 .....  
 End.

\_\_\_\_\_ x dan y yang dimaksud adalah x dan y real  
 (variabel global)

# MENG-GLOBAL-KAN VARIABEL





# PROSEDUR / FUNGSI DENGAN PARAMETER

- Parameter :yaitu Variabel yang digunakan untuk saling bertukar informasi antar bagian program:
  - antar prosedur
  - antar fungsi
  - program utama
- Parameter terbagi 2:
  - Parameter actual : parameter yang disertakan pada waktu pemanggilan
  - parameter formal: parameter yang dideklarasikan bersamaan dengan judul prosedur

# PARAMETER BY VALUE

- Parameter by Value yaitu parameter formal diprosedur akan berisi nilai yang akan dikirimkan yang kemudian bersifat local diprosedur,
  - Jika nilai parameter formal ini berubah tidak akan mempengaruhi parameter aktualnya/parameter searah
- Parameter by Reference terjadi jika yang dikirimkan ke fungsi berupa referensi (alamat) dan diterima oleh fungsi sehingga perubahan terhadap variable penerima ini akan mempengaruhi variable argumen yang memanggilnya

# FUNGSI

tipe data nilai kembalian

tipe data parameter

```
int nama_fungsi(int parameter)
{
    //tubuh fungsi berisi
    //kode program dari fungsi
}
```

# F U N G S I

- Fungsi identik dengan pengembalian nilai dari hasil prosesnya.
- Makanya perlu untuk menentukan tipe data untuk nilai yang dikembalikan

<b>FUNGSI</b>	<b>PROSEDUR</b>	<b>CONTOH</b>
Pendefinisian: tanpa void, tetapi dengan tipe pengembalian	Pendefinisian: dengan void	<i>Fungsi</i> : int tambah(int a, int b) <i>Proc</i> : void penambahan (int*hasil, int a, int b)
Pendefinisian: Alur logika dalam body fungsi HARUS berakhir pada SATU return	Tidak ada return. Paling tidak, tidak ada return yang dimaksud untuk mengembalikan value ke luar.	<i>Fungsi</i> : Int tambah(int a, int b) (                    return a+b; ) <i>Proc</i> : Void penambahan(int *hasil, int a, int b) (                    (*hasil)=a+b; )
Pemanggilan: harus menempel pada statemen	Pemanggilan: harus berdiri sendiri	<i>Fungsi</i> : z=tambah(a,b) ; <i>Proc</i> : Penambahan(&hasil, a,b) ;
by design:parameter input saja	by design: parameter input dan output	
Diperlukan sebagai value	Diperlakukan sebagai statemen.	<i>Fungsi</i> : Printf ("%d\n", tambah(a,b)) ; <i>Proc</i> : Penambahan (&hasil, a,b) ;

# KESIMPULAN

- Fungsi adalah suatu bagian dari program yang dimaksudkan untuk mengerjakan suatu tugas tertentu dan letaknya dipisahkan dari bagian program yang dijalankan.
- Prosedur adalah suatu bagian yang digunakan untuk mengenali subprogram di dalam suatu program.

## Perbedaan fungsi dan prosedur :

- Fungsi dapat mengembalikan suatu nilai yang dapat digunakan dalam ekspresi. Sedangkan, prosedur tidak memiliki nilai yang berhubungan dengan namanya.
- Fungsi digunakan untuk membuat operasi-operasi yang tidak ada dalam fungsi utama. Sedangkan, prosedur digunakan untuk menstrukturkan suatu program dan untuk memperbaiki kejelasan dan keumumannya.