



**PROGRAM STUDI**

**S1 SISTEM KOMPUTER**

**UNIVERSITAS DIPONEGORO**

# Dasar Algoritma

**Oky Dwi Nurhayati, ST, MT**  
**email: [okydnd@undip.ac.id](mailto:okydnd@undip.ac.id)**



# Materi

pseudocode

Masalah komputasi

Masalah analisis algoritma

# Konsep dasar algoritma

Beberapa metode untuk merancang algoritma dalam program komputer :

1. Diagram Alir (Flow Chart)
2. Kode Semu (Pseudo Code)
3. Algoritma Fundamental

Knuth (1973) menyatakan 5 komponen utama dalam algoritma yaitu finiteness, definiteness, input, output dan effectiveness.

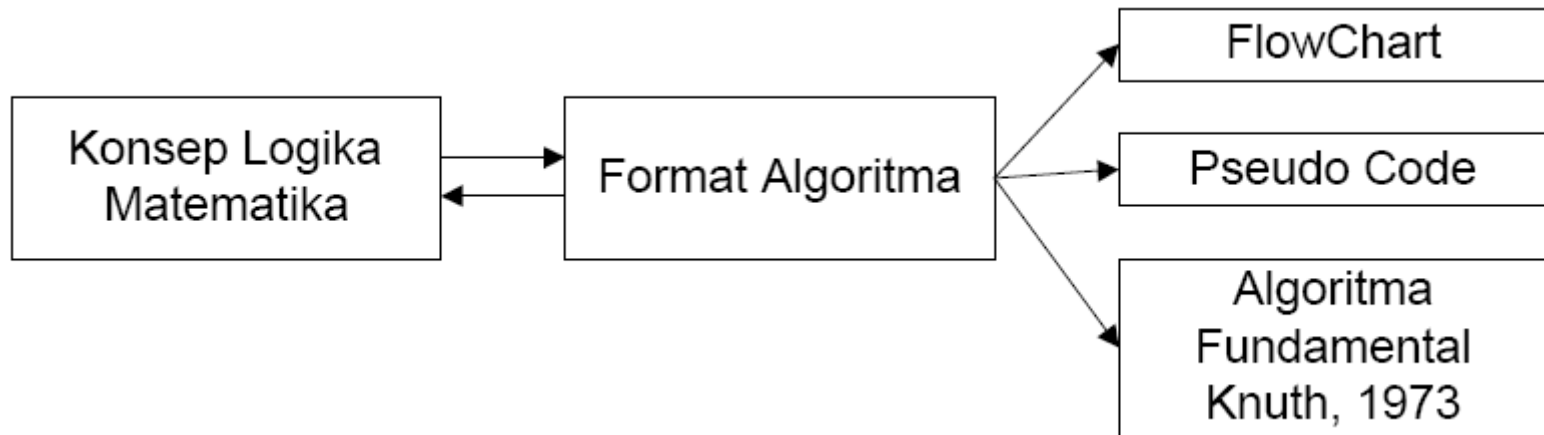
Komponen yang harus ada dalam merancang algoritma:

1. Komponen masukan : terdiri dari pemilihan variable, jenis variable, tipe variable, konstanta dan parameter (dalam fungsi).
2. Komponen keluaran: merupakan tujuan dari perancangan algoritma dan program.

Permasalahan yang diselesaikan dalam algoritma dan program harus ditampilkan dalam komponen keluaran. Karakteristik keluaran yang baik adalah menjawab permasalahan dan tampilan yang ramah

3. Komponen proses : merupakan bagian utama dan terpenting dalam merancang sebuah algoritma. Dalam bagian ini terdapat logika masalah, logika algoritma (sintaksis dan semantik), rumusan, metode (rekursi, perbandingan, penggabungan, pengurangan dll).

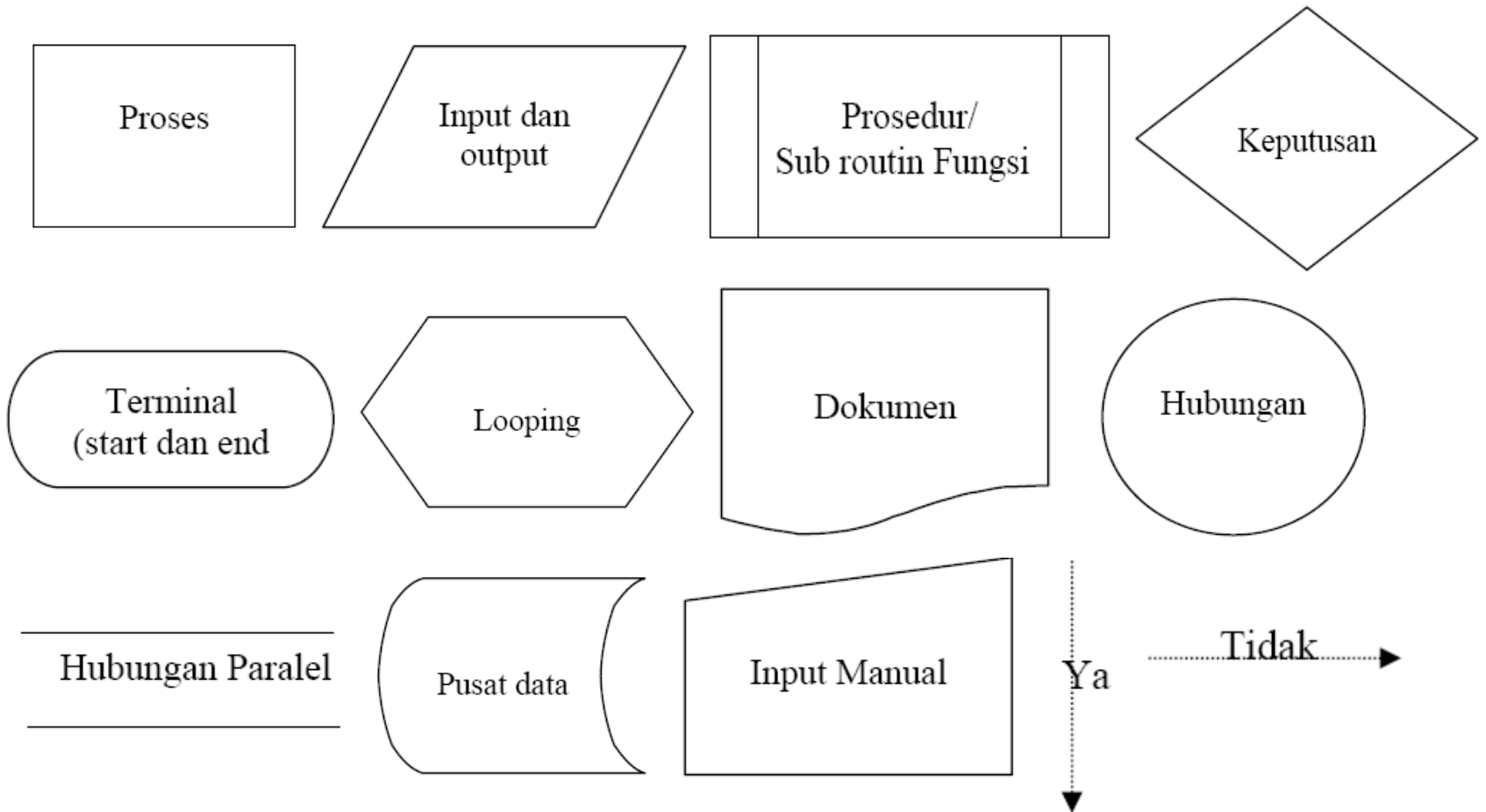
# Struktur Hubungan dan Jenis Algoritma



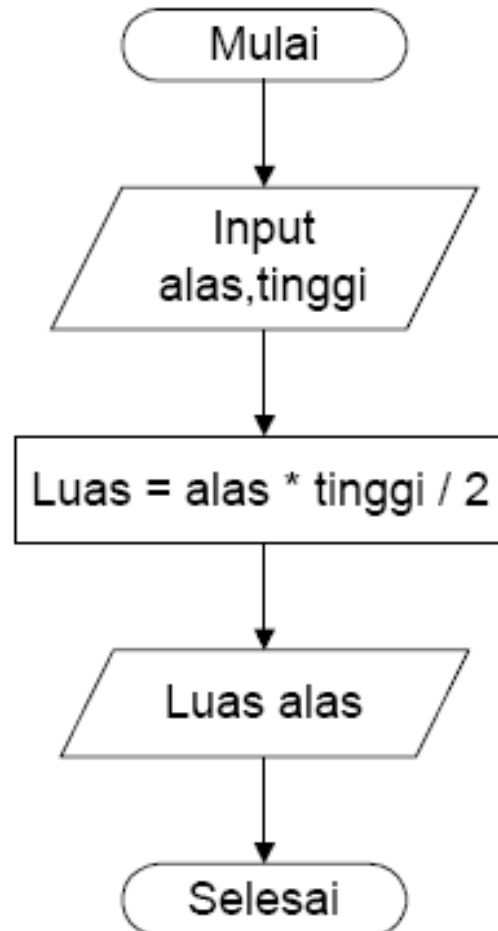
Flowchart : Algoritma ini menggunakan sejumlah simbol untuk menyatakan kegiatan-kegiatan secara keseluruhan.

Knuth (1973) menyajikan format algoritma yang dapat digunakan secara bebas untuk berbagai bahasa pemrograman, artinya dapat dengan mudah diimplementasikan menggunakan Pascal, C, Fortran, PL atau BASIC.

# Diagram alir



# Contoh penggunaan program flowchart



# Pseudocode

Kode atau tanda yang menyerupai (pseudo) atau merupakan penjelasan cara menyelesaikan suatu masalah. Pseudo-code sering digunakan oleh manusia untuk menuliskan algoritma.

Problem: mencari bilangan terbesar dari dua bilangan yang diinputkan.

*Contoh Pseudo-code:*

1. Masukkan bilangan pertama
2. Masukkan bilangan kedua
3. Jika bilangan pertama  $>$  bilangan kedua maka kerjakan langkah 4, jika tidak, kerjakan langkah 5.
4. Tampilkan bilangan pertama
5. Tampilkan bilangan kedua



# Aturan Algoritma Fundamental

1. Nama/judul algoritma harus ditulis dengan huruf kapital
2. Berikan komentar dan penjelasan pendahuluan.
3. Pernyataan dan struktur Kontrol
4. Nama-nama variabel harus ditulis dengan huruf besar
5. Input dan output
6. Prosedur
7. Fungsi

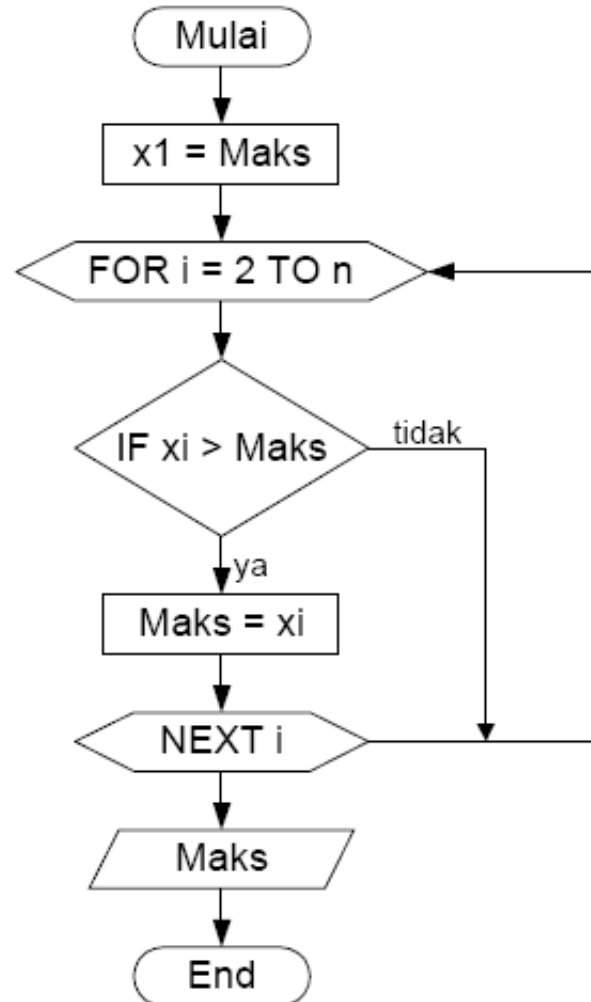
Masalah : Mencari elemen terbesar dari data dengan n bilangan.

Buatlah algoritma dari masalah ini menggunakan

Kode Semu : Algoritma Maksimum

1. Mula-mula masukkan bilangan dalam register  $x_i$  ke dalam register yang dinamakan maks.
2. Untuk  $i = 2, 3, \dots, n$ , lakukan : Bandingkan bilangan dalam register  $x_i$  dengan bilangan dalam register maks. Jika bilangan dalam register  $x_i$  lebih besar daripada bilangan dalam register maks, pindahkan bilangan dalam register  $x_i$  ke register maks; jika tidak jangan lakukan apa-apa.
3. Terakhir, bilangan dalam register maks adalah elemen terbesar di antara n bilangan.

# FlowChart (Diagram Alir)



# Algoritma Fundamental

Mencari elemen terbesar di dalam data dengan n bilangan.

1. [Inisialisasi]

Maks  $\leftarrow$  x<sub>1</sub>

2. [Mulai Loop]

I  $\leftarrow$  1

3. [Naikkan Pencacah]

I  $\leftarrow$  I + 1

4. [Bandingkan]

IF Maks < x<sub>i</sub>

THEN ~~Maks~~  $\leftarrow$  x<sub>i</sub> ELSE GOTO 3

5. [Ulangi Loop]

GOTO 3

6. [Selesai]

Exit

# Bahasa pemrograman pascal

Pascal adalah bahasa tingkat tinggi yang berorientasi pada segala tujuan.

struktur program pascal dapat terdiri dari :

1. Judul program
2. Bagian deklarasi
  - Deklarasi label
  - Deklarasi konstanta
  - Deklarasi tipe
  - Deklarasi variabel
  - Deklarasi prosedur
  - Deklarasi fungsi
3. Bagian pernyataan

# Tipe Data

Penentuan tipe data berguna untuk memberi pengenal pada isi data yang akan diakses, oleh variabel.

## a. Tipe Data Integer

Tipe data ini digunakan untuk menyatakan bilangan yang tidak mempunyai angka desimal. Tipe Integer terdiri dari beberapa tipe lagi, yang sebagian berbeda rentang nilai dan ukuran memorinya

| Tipe     | Rentang Nilai            | Byte |
|----------|--------------------------|------|
| Byte     | 0-225                    | 1    |
| Word     | 0-65535                  | 2    |
| ShortInt | -128 - 127               | 1    |
| SmallInt | -32768 - 32767           | 2    |
| Integer  | -2147483648 - 2147483647 | 4    |
| Cardinal | 0 – 2147483647           | 4    |
| LongInt  | -2147483648 - 2147483647 | 4    |

## b. Tipe Real

Tipe real digunakan untuk menyatakan bilangan yang mempunyai angka desimal. Tipe data real ini terdiri dari beberapa tipe lagi, yang sebagian rentang nilai dan ukuran penggunaan memorinya :

| Type     | Rentang Nilai                                       | Byte |
|----------|---|------|
| Real48   | $2.9 \times 10^{-39}$ s.d. $1.7 \times 10^{30}$     | 6    |
| Single   | $1.5 \times 10^{-45}$ s.d. $3.4 \times 10^{38}$     | 4    |
| Double   | $5.0 \times 10^{-325}$ s.d. $1.7 \times 10^{308}$   | 8    |
| Extended | $3.6 \times 10^{-4951}$ s.d. $1.1 \times 10^{4932}$ | 10   |
| Comp     | $-2^{63} + 1$ s.d. $2^{63} - 1$                     | 8    |
| Currency | -922337203685477.5808 s.d.<br>922337203685477.5807  | 8    |

### c. Tipe Boolean

Tipe data boolean untuk menyatakan data logika, yaitu True ( benar) dan False (salah). Borland Delphi 6.0 mempunyai 4 tipe boolean yaitu seperti berikut ini.

| Tipe     | Byte |
|----------|------|
| Boolean  | 1    |
| ByteBool | 1    |
| WordBool | 2    |
| LongBool | 4    |

Disarankan penggunaan tipe logika menggunakan tipe Boolean. Untuk ByteBool, WordBool, dan LongBool sebaiknya hanya digunakan untuk menjaga kompatibilitas, yaitu jika program akan dihubungkan dengan program bahasa lain.



#### d. Tipe Character

Tipe data character digunakan untuk menyatakan karakter satu huruf.

| Tipe     | Byte | Isi                |
|----------|------|--------------------|
| Char     | 1    | 1 karakter ANSI    |
| AnsiChar | 1    | 1 karakter ANSI    |
| WideChar | 2    | 1 karakter Unicode |

#### e. Tipe String

Tipe data string digunakan untuk menyatakan sederetan karakter yang membentuk satu kesatuan, misalnya nama, alamat dan sebagainya.

| Tipe        | Byte       | Isi               |
|-------------|------------|-------------------|
| ShortString | 2 s.d. 256 | 256 karakter      |
| AnsiString  | 4 s.d. 2GB | $2^{31}$ karakter |
| WideString  | 4 s.d. 2GB | $2^{30}$ karakter |

Tipe ShortString disediakan hanya untuk menjaga kompatibilitas dengan versi sebelumnya. AnsiString untuk menyimpan karakter ANSI dan WideString dapat menyimpan karakter Unicode.

f. g. **Tipe Record**

Tipe record digunakan untuk menyimpan sekumpulan data yang mungkin tipenya berbeda, tetapi saling berhubungan.

## h. Tipe Terenumerasi dan Subrange

Tipe data terenumerasi dan tipe data subrange dipakai untuk menyatakan data berurutan yang bertipe sama.

### iii. Konstanta

Konstanta adalah nilai yang bersifat tetap, misal angka 1, huruf A, nama dan lainlain.

Contoh :

**Const**

Diskon = 0.25;

Univ = 'Diponegoro';

# Variabel

Variabel adalah suatu pengenal yang menampung data, yang terdapat pada memori. Setiap variabel pasti mempunyai nama, yang sering disebut sebagai *identifier*.

Pendeklarasian variabel pada delphi:

```
var nama_variabel: tipe_variabel;
```

Contoh :

```
var
```

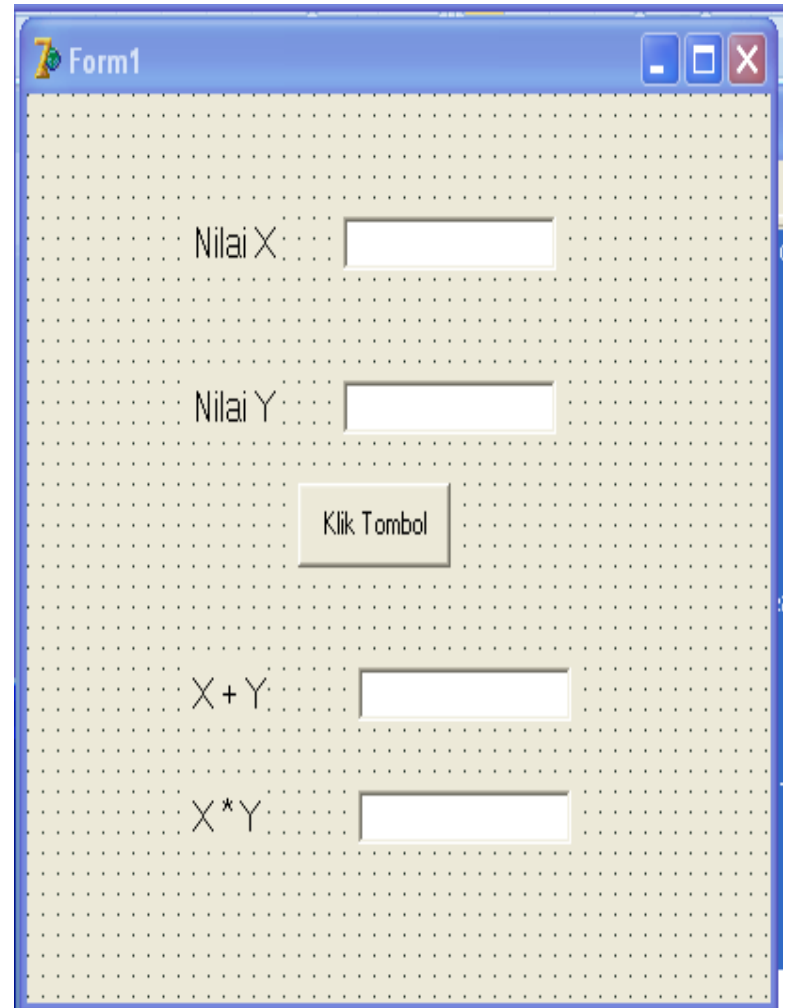
```
    i, j, k : integer;
```

```
    a, b, c : char;
```

# Menggunakan tipe bilangan bulat

```
procedure
 TForm1.Button1Click(Sender: TObject);
var
 X,Y,Z : byte;
Begin

 X := StrToInt(Edit1.Text);
 Y := StrToInt(Edit2.Text);
 // menjumlahkan X dan Y pada Edit3
 Z := X + Y;
 Edit3.Text := IntToStr(Z);
 // mengalikan X dan Y pada Edit4
 Z := X * Y;
 Edit4.Text := IntToStr(Z);
end;
```



# Menggunakan tipe bilangan real

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
X,Y,Z : real48;
```

```
i,j,k : extended;
```

```
Begin
```

```
X := StrToFloat(Edit1.Text);
```

```
Y := StrToFloat(Edit2.Text);
```

```
Z := X * Y;
```

```
Edit3.Text := FloatToStr(Z);
```

```
i:= StrToFloat(Edit1.Text);
```

```
j:= StrToFloat(Edit2.Text);
```

```
k:= i * j;
```

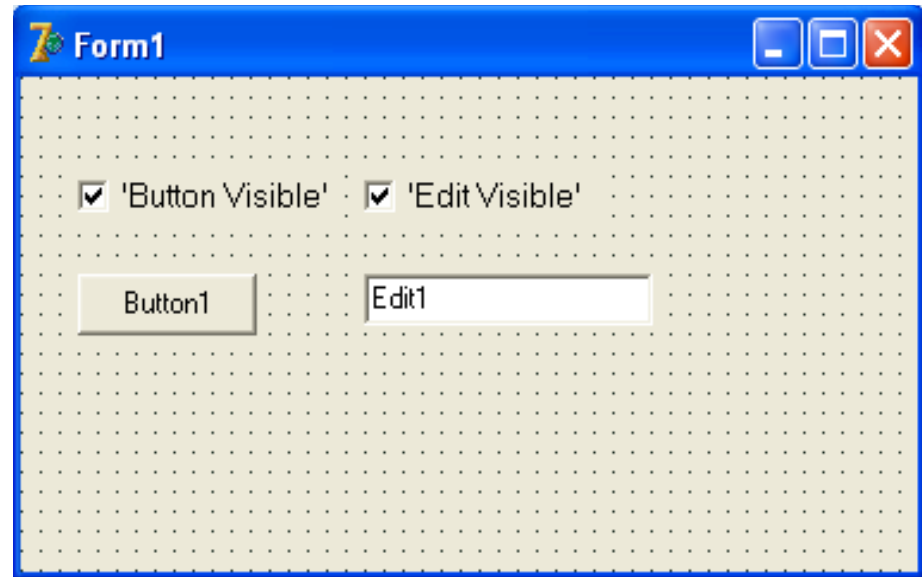
```
Edit4.Text := FloatToStr(k);
```

```
end;
```

# Menggunakan tipe bilangan boolean

```
procedure  
TForm1.CheckBox1Click(Sender:  
TObject);  
begin  
  Button1.Visible:=  
  CheckBox1.Checked;  
end;
```

```
procedure  
TForm1.CheckBox2Click(Sender:  
TObject);  
begin  
  Edit1.Visible:=CheckBox2.Checked;  
end;
```



Dengan mengubah properti checked dari kedua kotak cek diatas menjadi true maka kotak cek akan memiliki tanda cek

# Tipe array

Merupakan kumpulan variabel yang bertipe sama

## **Deklarasi dengan array**

```
var a: array [1..8] of longint;
```

## **Deklarasi tanpa array**

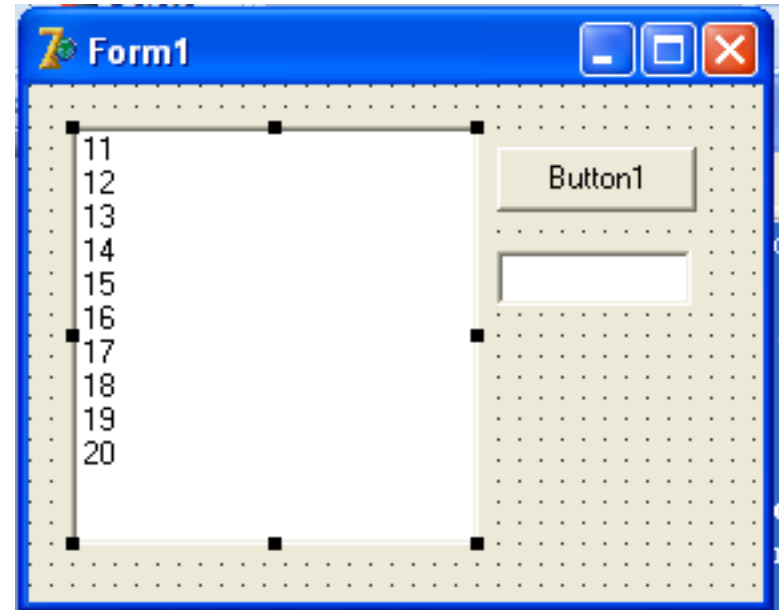
```
var a1, a2, a3, a4, a5, a6, a7, a8 : longint;
```



# array

```
procedure
 TForm1.Button1Click(Sender:
 TObject);
var
 A: array [0..20] of integer;
 i: integer;
 j: integer;
begin
   for i:=0 to ListBox1.Items.Count - 1
 do
   A[i] := StrToInt(ListBox1.Items[i]);

   j:=0;
   for i:= 0 to ListBox1.Items.Count - 1
 do
   j:= j + A[i];
   Edit1.Text:= IntToStr(j);
end;
```



# Tipe Record

Bila dengan array, variabel yang digabung memiliki tipe Yang sama, sedangkan record berfungsi menggabungkan beberapa variabel yang tipenya tidak harus sama.

**Sintaks :**

**type**

**nama\_record = record**

**//anggota record diletakkan disini**

**end;**

**Contoh :**

**type**

**mahasiswa = record**

**Nama : string;**

**Alamat : string;**

**end;**

# Operator

## a. Operator Aritmetika

| Operator | Operasi         | Tipe diproses   | Tipe Hasil    | Contoh             |
|----------|-----------------|-----------------|---------------|--------------------|
| +        | penjumlahan     | integer, real   | integer, real | $X + Y$            |
| -        | pengurangan     | integer, real   | integer, real | hasil - 1          |
| *        | perkalian       | integer, real   | integer, real | $P * InterestRate$ |
| /        | pembagian       | integer, real   | real          | $X / 2$            |
| Div      | pembagian bulat | integer integer | integer       | Total div UnitSize |
| Mod      | sisa bagi       | integer integer | integer       | $Y \text{ mod } 6$ |

## us. Operator Relasi

| Operator | Keterangan                   |
|----------|------------------------------|
| =        | Sama dengan                  |
| >        | Lebih besar                  |
| <        | Lebih kecil                  |
| >=       | Lebih besar atau sama dengan |
| <=       | Lebih kecil atau sama dengan |
| <>       | Tidak sama dengan            |

# Kontrol program

## a. Pengulangan While ... Do

Bentuk umum :

While *<ekspresi>* Do

Begin

*<pernyataan >*

End;

## b. Pengulangan Repeat ... Until

## c. Pencabangan bersyarat If .. Then ... Else

## d. Pencabangan bersyarat Case

# VISUAL COMPONENT LIBRARY

- Kumpulan dari sejumlah obyek tersedia pada *Visual Component Library (VCL)*. Sedangkan untuk memilih obyek terdapat pada *Componen palette*.
- Komponen-komponen dikelompokkan seseuai dengan fungsinya. Pengelompokan ini (*pages*) dinyatakan *tabs*. *Pages default* adalah :
- **Standart, Additional, Win32, System, internet, Data Access, Data Control, Midas, Decision Cube, Qreport, Dialogs, Win31, Samples, EctiveX dan lain-lain.**

- Component pallet menyediakan komponen-komponen yang diperlukan untuk menyusun program aplikasi.
- Object Inspector merupakan penghubung antara tampilan aplikasi dengan kode program, dan melakukan seting terhadap object-object yang terdapat pada form.
  - a. Properti Page, untuk menentukan properti dan komponen terpilih.
  - b. Event Page, untuk menentukan kejadian atau proses yang akan dilakukan, yang berisi perintah-perintah yang dimiliki masing-masing komponen.

# Pemrograman Visual

Pemrograman berorientasi Objek

Pemrograman yang dikendalikan oleh kejadian (*even drivent*)

Langkah membuat aplikasi :

- a) Mambuat *user interface*
- b) Menulis kode

Form tersusun sejumlah kontrol atau objek (Button, Label, Edit, RadioButton, CheckBox, dll).

Kontrol mempunyai tiga komponen :

- a) Properti : misalnya menentukan warna dan mengatur Font.
- b) Kejadian (*event*)
- c) Metode : suatu fungsi untuk melakukan operasi terhadap kontrol.

# Masalah analisis algoritma

Terdapat dua tipe analisis algoritma, yaitu :

## 1. Memeriksa kebenaran algoritma

Dapat dilakukan dengan cara perurutan, memeriksa bentuk logika, implementasi algoritma, pengujian dengan data dan menggunakan cara matematika untuk membuktikan kebenaran.

## 2. Penyederhanaan Algoritma

Membagi algoritma menjadi bentuk yang sederhana.

Dalam analisis sebuah algoritma biasanya yang dijadikan ukuran adalah operasi aljabar seperti penjumlahan, pengurangan, perkalian dan pembagian, proses pengulangan (looping/Iterasi), proses pengurutan (sorting) dan proses pencarian (searching).



# Analisis Algoritma

- Sebuah algoritma tidak hanya harus benar, tetapi juga harus mangkus (*efficient*)
- Ukuran kemangkusan algoritma: waktu dan ruang memori (*space*)
- Algoritma yang mangkus: algoritma yang meminimumkan kebutuhan waktu dan ruang

## Alat ukur kemangkusan algoritma:

1. Kompleksitas waktu,  $T(n)$
2. Kompleksitas ruang,  $S(n)$

$n =$  ukuran masukan yang diproses oleh algoritma

$T(n)$  : jumlah operasi yang dilakukan untuk menjalankan sebuah algoritma sebagai fungsi dari ukuran masukan  $n$ .

$S(n)$ : ruang memori yang dibutuhkan algoritma sebagai fungsi dari ukuran masukan  $n$

Operasi yang dihitung hanyalah operasi dasar

Operasi dasar: operasi khas yang mendasari suatu algoritma

Misalnya:

- operasi perbandingan elemen pada algoritma pengurutan/pencarian
- operasi penjumlahan dan perkalian pada algoritma perkalian matriks

## Kompleksitas waktu asimptotik:

- perkiraan kasar kebutuhan waktu algoritma dengan meningkatnya nilai  $n$
- *menyatakan laju pertumbuhan waktu, bukan menyatakan jumlah operasi dasar sesungguhnya.*

## Tiga cara menyatakan waktu asimptotik:

1.  $O(f(n))$ : *untuk batas atas laju kebutuhan waktu*
2.  $W(g(n))$ : *untuk batas bawah laju kebutuhan waktu*
3.  $Q(h(n))$  : *jika  $f(n) = g(n)$*

# Notasi O (Big O)

Misalkan 4 program yang mensorting  $n$  bilangan dengan fungsi yang menyatakan sejumlah langkah yang dijumlahkan masing-masing program untuk sorting  $n$  bilangan :

$$f1(n) = n, f2(n) = n^2, f3(n) = 2^n, f4(n) = n!$$

Bila  $n = 4$  maka  $f1(n) = 4$ ,  $f2(n) = f3(n) = 16$  dan  $f4(n) = 2^4$  sedangkan apabila  $n = 100$ ,

program keempat akan memerlukan  $2^{100}$  langkah.